



Microsoft Azure AI Fundamentals

SECOND EDITION

Exam Ref

AI-900

Julian Sharp
Francesco Esposito
Stefano Costanzo
Martina D'Antoni

Exam Ref AI-900 Microsoft Azure AI Fundamentals

Second Edition

**Julian Sharp
Francesco Esposito
Stefano Costanzo
Martina D'Antoni**



Exam Ref AI-900 Microsoft Azure AI Fundamentals Second Edition

Published with the authorization of Microsoft Corporation by:
Pearson Education, Inc.

Copyright © 2026 by Pearson Education, Inc.
Hoboken, New Jersey

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms, and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearson.com/global-permission-granting.html.

No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-13-541713-3

ISBN-10: 0-13-541713-9

Library of Congress Control Number: 2025946664

`$PrintCode`

TRADEMARKS

Microsoft and the trademarks listed at www.microsoft.com on the “Trademarks” webpage are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

WARNING AND DISCLAIMER

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The author, the publisher, and Microsoft Corporation shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the programs accompanying it.

EDITOR-IN-CHIEF

Julie Phifer

EXECUTIVE EDITOR

Loretta Yates

ACQUISITIONS EDITOR

Shourav Bose

DEVELOPMENT EDITOR

Songlin Qiu

MANAGING EDITOR

Sandra Schroeder

SENIOR PROJECT EDITOR

Tracey Croom

COPY EDITOR

Linda Laflamme

INDEXER

Rachel Kuhn

PROOFREADER

Donna E. Mulder

TECHNICAL EDITOR

Thomas Palathra

COVER DESIGNER

Twist Creative, Seattle

COMPOSITOR

codeMantra

GRAPHICS
codeMantra

Contents at a glance

Introduction

CHAPTER 1 **Describe artificial intelligence workloads and considerations**

CHAPTER 2 **Describe fundamental principles of machine learning on Azure**

CHAPTER 3 **Describe features of computer vision workloads on Azure**

CHAPTER 4 **Describe features of natural language processing (NLP) workloads on Azure**

CHAPTER 5 **Describe features of generative AI workloads on Azure**

CHAPTER 6 **AI-900 Microsoft Azure AI Fundamentals exam updates**

Index

Contents

Introduction

- Organization of this book*
- Preparing for the exam*
- Microsoft certifications*
- Access the Exam Updates chapter and online references*
- Errata, updates, & book support*

Chapter 1 Describe artificial intelligence workloads and considerations

- Skill 1.1: Identify features of common AI workloads
 - Identify computer vision workloads
 - Identify natural language processing workloads
 - Identify document processing workloads
 - Identify generative AI workloads
- Skill 1.2: Identify guiding principles for Responsible AI
 - Describe the Fairness principle
 - Describe the Reliability and Safety principle
 - Describe the Privacy and Security principle
 - Describe the Inclusiveness principle
 - Describe the Transparency principle
 - Describe the Accountability principle
 - Understand the Microsoft AI for Good program

Chapter summary

Thought experiment

Thought experiment answers

Chapter 2 Describe fundamental principles of machine learning on Azure

- Skill 2.1: Identify common machine learning types
 - Understand machine learning model types
 - Describe regression models
 - Describe classification models
 - Describe clustering models
 - Describe deep learning techniques
 - Describe features of the Transformer architecture
- Skill 2.2: Describe core machine learning concepts
 - Understand the machine learning workflow
 - Identify the features and labels in a dataset for machine learning
 - Describe how training and validation datasets are used in machine learning
 - Describe how machine learning algorithms are used for model training
 - Select and interpret model evaluation metrics
- Skill 2.3: Identify core tasks in creating a machine learning solution
 - Understand machine learning on Azure
 - Understand Azure Machine Learning Studio
 - Describe data ingestion and preparation
 - Describe feature selection and engineering
 - Describe model training and evaluation
 - Describe model deployment and management
- Skill 2.4: Describe capabilities of no-code machine learning with Azure Machine Learning
 - Describe Azure Automated Machine Learning
 - Describe Azure Machine Learning Designer

Chapter summary

Thought experiment

Thought experiment answers

Chapter 3 Describe features of computer vision workloads on Azure

Skill 3.1: Identify common types of computer vision solutions

- Describe AI Services

- Understand computer vision

- Describe image classification

- Describe object detection

- Describe optical character recognition

- Describe facial detection, recognition, and analysis

Skill 3.2: Identify Azure tools and services for computer vision tasks

- Understand the capabilities of the Vision service

- Understand the Custom Vision service

- Understand the Face service

Chapter summary

Thought experiment

Thought experiment answers

Chapter 4 Describe features of natural language processing (NLP) workloads on Azure

Skill 4.1: Identify features of common NLP workload scenarios

- Describe natural language processing

- Describe language modeling

- Describe key phrase extraction

- Describe named entity recognition

- Describe sentiment analysis

- Describe speech recognition and synthesis

- Describe translation

Skill 4.2: Identify Azure tools and services for NLP workloads

- Identify the capabilities of the Azure AI Language service

- Identify the capabilities of the Azure AI Speech service

- Identify the capabilities of the Azure AI Translator service

Chapter summary

Thought experiment

Thought experiment answers

Chapter 5 Describe features of generative AI workloads on Azure

Skill 5.1: Identify features of generative AI solutions

- Identify features of generative AI models

- Identify common scenarios for generative AI

- Identify Responsible AI considerations for generative AI

Skill 5.2: Identify Azure services and capabilities for generative AI

- Describe features and capabilities of Azure AI Foundry

- Describe features and capabilities of the Azure AI Foundry model catalog

- Describe features and capabilities of Azure OpenAI Service

Chapter summary

Thought experiment

Thought experiment answers

Chapter 6 AI-900 Microsoft Azure AI Fundamentals exam updates

The purpose of this chapter

- About possible exam updates

- Impact on you and your study plan

Exam objective updates

Updated technical content

Objective mapping

Index

Acknowledgments

I owe a huge thank you to my incredible team, the ones who keep everything running smoothly while I sneak away to write books. Without you, the world around us might very well catch fire (and I mean that only slightly metaphorically). Your dedication and reliability make this work, as well as many others, possible.

A special thank you to Loretta and Shourav for bringing us on board for the review, and to Julian, whose original work laid the foundation for everything that followed.

And of course, sincere thanks to Songlin and Thomas, our editors, for their attention and feedback throughout the process.

Francesco Esposito

Special thanks to Micucci and Massimo for their unwavering support and the many conversations over coffee that sustained me throughout the writing process. To Cosmin and Christina for their encouragement and valuable insights.

Stefano Costanzo

For this journey, I would like to express my deepest gratitude to Francesco, whose constant support—both as a friend and throughout our professional path—has made all of this, and so much more, possible.

A special thanks goes to Stefano, the ideal review partner in what was the first editing experience for both of us. Your insight, commitment, and spirit of collaboration made the process truly rewarding.

I am also sincerely grateful to Shourav and Loretta for inviting us into this project and believing in our contribution. Finally, thank you to the editors, Songlin and Thomas, who reviewed and refined our work with care and precision.

Some contributions, however, are not tied to a single project but to a lifetime of encouragement. And so, thank you to everyone who has silently supported me along the way.

Martina D'Antoni

About the authors

JULIAN SHARP is a solutions architect, trainer, and Microsoft Business Applications MVP with over 30 years of experience in IT. He completed his MA in Mathematics at the University of Cambridge. Julian has spoken at Microsoft Ignite and many other community events. For the past 15 years, he has been a Microsoft Certified Trainer delivering certification training around Dynamics 365, Azure, and the Power Platform. He has taught thousands of students with a high pass rate. Julian has a passion for artificial intelligence to enhance user experience and customer data in the solutions that he designs.

FRANCESCO ESPOSITO, armed with a strong mathematical background, is the author of three Microsoft Press books on machine learning. He also founded Youbiquitous, an AI factory mostly working on LLMs and computer vision, and KBMS Data Force, a startup focused on digital therapy and intelligent healthcare. He is also Head of Engineering & Data at Crionet, a company that provides innovative solutions for the sport and media industries.

STEFANO COSTANZO is a computer engineer with a deep passion for artificial intelligence. He completed his Master's degree at Tor Vergata University, where his thesis explored how large language models and information-retrieval techniques can be combined to detect fake news. Stefano approaches AI development with a solution-oriented mindset. He sees artificial intelligence as an innovative tool that can be integrated into various sectors in surprising ways, unlocking new possibilities for creative solutions.

MARTINA D'ANTONI is a software engineer with a background in clinical and biomedical engineering. Since 2021, Martina has worked in the healthcare sector, applying a blend of traditional software development and artificial intelligence to support innovation in medical technologies. Driven by curiosity and a keen interest in both data and language, she approaches problems with precision and analytical rigor. Her passion for continuous learning fuels her commitment to exploring how emerging technologies, particularly AI, can enhance patient care and shape the future of healthcare, as well as how they can more broadly support and integrate with human development.

Introduction

Artificial intelligence (AI) impacts almost everything we do today with devices and computers. The purpose of the AI-900 exam is to test your understanding of the fundamental concepts of AI and the AI services Microsoft provides in Azure. The exam includes high-level concepts for AI and machine learning (ML), as well as the capabilities of particular Azure AI services.

Having a high-level appreciation of AI is important to everyone involved in building and using solutions that make use of AI. There are no prerequisite skills or experience required.

Like the exam, this book takes a high-level approach and is geared toward giving you a broad understanding of the use cases for AI, as well as the common AI services in Azure. Both the exam and the book are at such a high level that there is no coding involved.

This book covers every major topic area found on the exam, but it does not cover every exam question. Only the Microsoft exam team has access to the exam questions, and Microsoft regularly adds new questions to the exam, making it impossible to cover specific questions. You should consider this book to be a supplement to your relevant real-world experience and other study materials. If you encounter a topic in this book that you do not feel completely comfortable with, use the “Need more review?” links you’ll find in the text to find more information and take the time to research and study the topic. Great information is available on Microsoft Learn, and in blogs and forums.

Organization of this book

This book is organized by the “Skills measured” list published for the exam. The “Skills measured” list is available for each exam on the Microsoft Learn website: microsoft.com/learn. Each chapter in this book corresponds to a major topic area in the list, and the technical tasks in each topic area determine a chapter’s organization. If an exam covers six major topic areas, for example, the book will contain six chapters.

Preparing for the exam

Microsoft certification exams are a great way to build your resume and let the world know about your level of expertise. Certification exams validate your on-the-job experience and product knowledge. Although there is no substitute for on-the-job experience, preparation through study and hands-on practice can help you prepare for the exam. This book is *not* designed to teach you new skills.

We recommend that you augment your exam preparation plan by using a combination of available study materials and courses. For example, you might use the Exam Ref and another study guide for your “at home” preparation and take a Microsoft Official Curriculum course for the classroom experience. Choose the combination that you think works best for you. You can learn more about available classroom training and find free online courses and live events at microsoft.com/learn.

Note that this Exam Ref is based on publicly available information about the exam and the authors’ experience. To safeguard the integrity of the exam, authors do not have access to the live exam.

Microsoft certifications

Microsoft certifications distinguish you by proving your command of a broad set of skills and experience with current Microsoft products and technologies. The exams and corresponding certifications are developed to validate your mastery of critical competencies as you design and develop, or implement and support, solutions with Microsoft products and technologies both on-premises and in the cloud. Certification brings a variety of benefits to the individual and to employers and organizations.

Need More Review? All Microsoft Certifications

For information about Microsoft certifications, including a full list of available certifications, go to microsoft.com/learn.

Check back often to see what is new!

Access the Exam Updates chapter and online references

The final chapter of this book, “AI-900 Azure AI Fundamentals exam updates,” will be used to provide information about new content per new exam topics, content that has been removed from the exam objectives, and revised mapping of exam objectives to chapter content. The chapter will be made available from the link below as exam updates are released.

Throughout this book are addresses to webpages that the authors have recommended you visit for more information. Some of these links can be very long and painstaking to type, so we’ve shortened them for you to make them easier to visit. We’ve also compiled them into a single list that readers of the print edition can refer to while they read.

The URLs are organized by chapter and heading. Every time you come across a URL in the book, find the hyperlink in the list to go directly to the webpage.

You can download the Exam Updates chapter and the URL list at MicrosoftPressStore.com/ERA19002e/downloads.

Errata, updates, & book support

We've made every effort to ensure the accuracy of this book and its companion content. You can access updates to this book—in the form of a list of submitted errata and their related corrections—at MicrosoftPressStore.com/ERA19002e/errata.

If you discover an error that is not already listed, please submit it to us at the same page.

For additional book support and information, please visit MicrosoftPressStore.com/Support.

Please note that product support for Microsoft software and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to support.microsoft.com.

Describe artificial intelligence workloads and considerations

Artificial intelligence (AI) is computers analyzing and responding in a way that simulates a human. AI is a technology that takes information from its environment and responds based on what it learns. The goal of AI is to create a machine that can mimic human behavior.

AI is more than learning—it is knowledge representation, reasoning, and abstract thinking. *Machine learning (ML)* is the subset of AI that takes the approach of teaching computers to learn for themselves, rather than teaching computers all that they need to know. ML is the foundation for modern AI. ML focuses on identifying and making sense of the patterns and structures in data.

ML is about machines' reasoning and decision-making using software that learns from past experiences. ML allows computers to consistently perform repetitive and well-defined tasks that are difficult for humans to accomplish. Over the past few years, machine learning algorithms have proved that computers can learn tasks that are tremendously complicated for machines, demonstrating that ML can be employed in a wide range of scenarios and industries. These advancements have occurred in a very short time frame, and an ever broader range of industries and organizations are trying to implement emerging AI technologies to change and improve how they operate.

AI is now part of many of the software applications that you use every day, sometimes without your even knowing it. For instance, Microsoft PowerPoint offers a Design Ideas feature that provides recommendations on slides, themes, and layouts. Likewise, Microsoft Word will provide suggestions on how to rewrite sentences to enhance clarity. AI-related features can be found in an endless number of applications, in contexts like email filtering, voice-activated assistants, recommendation systems, and automated customer service.

In this opening chapter, you will explore the main types of AI workloads and how they map to real-world applications. Each workload discussed corresponds to services available within the Microsoft Azure platform, which will be examined in more detail in the chapters that follow.

Skills covered in this chapter:

- [Skill 1.1: Identify features of common AI workloads](#)
- [Skill 1.2: Identify guiding principles for Responsible AI](#)

Skill 1.1: Identify features of common AI workloads

Artificial intelligence is software that mimics human behaviors and capabilities. Today, software can use AI to automatically detect and predict actions that machines, and humans, should take.

AI workloads on Azure can generally be classified into four categories. Each category is supported by multiple purpose-built Azure AI services that you should be familiar with for the AI-900 exam.

This skill covers how to:

- [Identify computer vision workloads](#)
- [Identify natural language processing workloads](#)
- [Identify document processing workloads](#)
- [Identify generative AI workloads](#)

Identify computer vision workloads

Computer vision affords machines the capacity to process and interpret visual information from images and video streams. Much like humans quickly recognize objects, read text, and identify faces, computer vision systems can analyze visual input and make actions and provide interpretations that become useful to various applications.

Core computer vision capabilities

Computer vision systems can:

- Generate descriptive captions for images
- Classify and categorize images for searchability and organization
- Perform object detection, locating objects using spatial coordinates
- Execute face detection, identifying faces and analyzing attributes
- Recognize specific brands, products, landmarks, or celebrities
- Use optical character recognition (OCR) to extract printed or handwritten text from images and convert it into digital, searchable content

Computer vision systems use deep neural networks, trained on millions of example images to recognize patterns, shapes, textures, and relationships of objects that humans recognize.

Azure AI Vision, which [Chapter 3](#), “[Describe features of computer vision workloads on Azure](#),” discusses more deeply, includes the Image Analysis

feature, which provides unified tagging, captioning, object detection, and OCR through a single endpoint. Each capability can be reached via REST APIs (representational state transfer application programming interfaces), client SDKs (software development kits), or the browser interface in Azure AI Foundry.

Common use cases

Computer vision systems benefit a wide variety of industries, such as:

- Retail businesses use automated barcode and text reading technology, which can speed inventory and order management.
- In manufacturing, computer vision is applied to automated quality assurance activities.
- Pharmaceutical companies use automation to inspect medication bottles from all sides and to check caps, closures, and labels to make sure everything adheres to packaging requirements.
- In healthcare, computer vision systems analyze medical scans to identify potential abnormalities, classify which cases require a specialized medical professional's attention, and enable diagnostic examiners to focus on only the most complicated and critical cases.

Identify natural language processing workloads

Similar to the way that computer vision helps machines understand what they are looking at, *natural language processing* (NLP) allows machines to know what is written and spoken. Think about how easily you can read a text message, know what it means, identify the mood of the person who sent it, and reply to them appropriately. Likewise, NLP allows computers to exhibit this linguistic intelligence and capabilities of meaningfully interacting by analyzing language and understanding its meaning.

In general, NLP takes the complexity of human language and translates it into structured information that machines can manipulate and act upon. Each time you write an email, speak with a voice assistant, or chat with a customer service bot, the NLP systems are busy working behind the curtains to not only determine the words that you input, but also the context, intent, and emotional tone of the entire conversation.

Core NLP capabilities

Natural language processing systems can be used for:

- **Language detection** Recognize the language being used
- **Key phrase extraction** Identify the main topics of the conversation
- **Named entity recognition (NER)** Extract names of people, organizations, locations, and dates
- **Sentiment analysis** Determine whether the tone is positive, negative, or neutral
- **Question answering and summarization** Understand questions and generate concise answers or content summaries

Azure AI Language, which [Chapter 4, “Describe features of natural language processing \(NLP\) workloads on Azure,”](#) discusses in detail, offers capabilities such as entity recognition, sentiment analysis, key phrase extraction, question answering, and summarization, all accessible through a unified endpoint. These features are available via REST APIs, client SDKs, or through Azure AI Foundry.

Azure provides access to these capabilities with its two main services that complement each other well: Azure AI Language and Azure AI Speech.

- Azure AI Language covers text analytics functionality.
- Azure AI Speech covers the process of transcription, both speech-to-text and text-to-speech.

Together, they enable developers to create applications that understand written and spoken human communication.

Common use cases

When an organization is managing information that will change quickly and comes from different outlets, NLP can discern meaningful insights quickly, allowing for faster responses than if someone had to do it manually.

Conversational AI is dependent on NLP, as well. Modern chatbots and intelligent agents use NLP to understand customers' questions, no matter how they are phrased. This type of flexibility allows users to speak naturally, rather than requiring them to remember a specific series of commands or keywords, and it provides an easier and more intuitive experience.

Email management is another example of an everyday application of NLP. Spam detection systems are simply another form of NLP that analyze not only email subject lines but also the contents of messages to automatically categorize messages as potential junk mail or legitimate correspondence.

Voice-enabled applications are excellent examples of how NLP connects human speech to digital interaction. These applications can transform spoken words into text in real time, as well as process text and its meaning into simulated speech that sounds natural. Interacting with such applications, therefore, feels more conversational and less robotic.

Identify document processing workloads

Every day, businesses encounter innumerable documents—invoices, agreements, receipts, forms, reports—that contain knowledge that is locked in unstructured formats. *Document processing workloads* convert these documents from their original form—PDFs, scanned images, Microsoft Office documents—into structured, business-ready data that your applications can use.

Core document processing capabilities

A system that uses document processing can perform:

- **Layout analysis** The system can recognize and preserve the visual layout of complex documents (such as contracts or reports), maintaining the logical reading order, indentation, and sections.

- **Named field extraction** The system can be trained to extract specific fields (such as Invoice Date, Total Amount, or Policy Number) from custom document formats using custom models, enabling high accuracy even with nonstandard templates.
- **Form recognition** In structured forms (application forms or surveys), the system identifies labels and their corresponding values, even when they are spread across multiple pages or vary in placement.
- **Table extraction** The system can detect and extract tables, preserving rows, columns, and nested structures. This is especially useful for processing expense reports, financial statements, and product catalogs.

Azure AI Document Intelligence (formerly Form Recognizer), which [Chapter 3](#) discusses more thoroughly, acts as your digital document assistant. Its capabilities are accessible via REST APIs, SDKs, or directly in Azure AI Foundry.

By working with Azure AI Document Intelligence, you can take advantage of powerful capabilities to help you process documents at scale. Some of the service's key capabilities include:

- Batch document processing, which allows hundreds of documents to be handled at once and is ideal for such high-volume scenarios as invoice processing or document archiving
- Creation of machine-readable PDFs with searchable text even from scanned images
- Support for printed and handwritten text in more than 250 languages and dialects, making it a globally usable solution
- Extraction of searchable content in over 100 languages

Common use cases

Workloads in document processing provide immediate value in many business situations, such as:

- Accounts payable departments can automatically extract vendor details, amounts, and line items from invoices.
- Human resources teams can process resumes and job applications, extracting directly the pertinent qualifications and experience without relying on human readers.
- For many industries, compliance and audit functions provide the ability to extract critical information from myriad documents faster, with an assurance they meet regulatory requirements while limiting the time reviewing manually.
- Insurance companies can process claims faster, extracting policy numbers, dates, and claims via automated processing from submitted documents.

Identify generative AI workloads

Generative AI differs from traditional AI in that it produces new content, rather than just analyzing existing data. Traditional AI analyzes an input like an image and classifies it, predicts the outcome of some event, and so on. Generative AI applies the patterns it learned during training to create a new image, text, code, video, or audio.

Generative AI relies on foundation models that train on massive amounts of data. Traditional machine learning approaches are quite different, because you build a model for each task. In generative AI, there are no custom models per se; rather, there are pretrained models that you can reuse and adapt in specific situations through the deft art of prompt engineering and fine-tuning.

That said, you can understand why generative AI requires a new operational model. GenAIOps expands the existing MLOps activities used with machine learning to also include the operationalization processes required by generative models.

Specifically, engineers identified a number of activities required to operationalize the prompt-managed, iterative development processes required by generative language models. These include prompt management, model selection, iterative output management, and refinement through continuous output feedback and Retrieval-Augmented Generation (RAG) activities to help ground and provide data from your organization for the AI responses.

Core generative AI capabilities

Generative AI systems can be used for:

- **Natural language generation** Generate text, summaries, answers, or rewrites from brief prompts
- **Image generation** Create images from text descriptions, which is ideal for design, content, or marketing
- **Code generation** Write or complete code, explain existing code, or translate between programming languages
- **Audio generation** Produce lifelike speech for support tools, accessibility, or media content

Beyond content creation, generative models can be augmented with your enterprise data through Retrieval-Augmented Generation (RAG). This enables grounded responses that combine the creative power of generative models with the factual accuracy and relevance of your internal knowledge sources.

Thanks to multilingual support, generative AI can generate or translate content in many of the world's languages, making it suitable for global applications. Combined with prompt engineering and fine-tuning, these capabilities can be tailored to meet your organization's specific tone, compliance needs, or domain knowledge.


Azure OpenAI Service, which [Chapter 5](#), “Describe features of generative AI workloads on Azure,” discusses, enables access to powerful foundation models like GPT and DALL·E, supporting text generation, summarization, code creation, and image generation. These generative capabilities can be accessed through REST APIs, language-specific SDKs, or explored interactively in Azure AI Foundry.

Common use cases

Generative AI does well in a variety of scenarios. For example, it can:

- Draft emails, articles, social media posts, or other content from bullet points or guidelines
- Create boilerplate code, help debug code based on a description in natural language, improve code quality through suggestions, and help onboard new developers faster by explaining code in natural language

- Create images based on text descriptions, create audio content, or even create captions for visual content
- Provide accurate, grounded answers to complex questions when combined with your relevant documents via a RAG-powered question-answering system
- Power virtual customer service agents that provide accurate, conversational responses, resolve issues more efficiently, and personalize interactions using customer data
- Help marketing teams brainstorm campaign ideas, generate ad copy variations, and localize content for different markets in seconds

 **Exam Tip**

You will be asked in the exam to pick the correct AI workload from an example given or to identify an example for a specified AI workload. Make sure you understand the use cases for each workload.

Skill 1.2: Identify guiding principles for Responsible AI

Responsible AI is the provision of AI-based solutions to difficult problems without any unintended negative consequences.

This section covers the six principles of Responsible AI. By following these principles, you can ensure that your AI-enhanced solutions will put people first.

 **Exam Tip**

Make sure you can describe each principle of Responsible AI in a single sentence.

This skill covers how to:

- [Describe the Fairness principle](#)
- [Describe the Reliability and Safety principle](#)
- [Describe the Privacy and Security principle](#)
- [Describe the Inclusiveness principle](#)
- [Describe the Transparency principle](#)
- [Describe the Accountability principle](#)
- [Understand the Microsoft AI for Good program](#)

Note Principles for Responsible AI

These principles should be taken into account when creating solutions that use AI. Not every principle will apply to every requirement, but you should check your solution against each principle to see if it applies.

Describe the Fairness principle

The Fairness principle of Responsible AI is concerned with treating all people fairly and reducing unfairness.

A Responsible AI-based solution must operate without giving any unfair advantage to, withholding opportunities from, or allocating resources to a specific group of people. There should be no bias regarding a person's gender or any other characteristic.

AI systems can reinforce existing stereotypes and underrepresentation. If correctly addressed, AI systems can reduce unfairness.

An AI model should be interpreted to quantify the extent of how data influences the model's prediction to help eliminate bias.

The Fairness principle means:

- Eliminating bias for gender, age, or ethnicity
- Removing unfair advantages
- Preventing unfair allocation of resources
- Preventing the withholding of information
- Mitigating bias at each stage of developing and deploying AI systems

Some examples of where and when the Fairness principle can have a significant impact are:

- Bank loans and credit decisions
- Hiring staff

- Criminal justice system

Describe the Reliability and Safety principle

Reliability and Safety requires the rigorous testing of an AI-based system's functionality and deployment to ensure that it works as expected and eliminates potential risk to human life.

Features of Reliability and Safety are:

- Rigorous testing
- Works as expected
- Eliminates threat of harm to human life

Areas where Reliability and Safety must be applied include:

- Autonomous vehicles
- Healthcare diagnosis

Describe the Privacy and Security principle

Privacy and Security requires that an AI-based system should be secure and respect privacy. AI-based systems typically operate on high volumes of data, including personal data that should not be disclosed.

The reliance on data used in training the model and new data used for predictions is subject to privacy rules.

AI systems that run on a user's device should not leak the user's data. One way to achieve this is to run the AI processing on the device and not transfer the personal data to a cloud service.

To be responsible, you should:

- Respect privacy
- Be secure
- Avoid disclosing personal data

Concerns covered by Privacy and Security include:

- **Data origin** Where has the data come from? Is it user or public data?
- **Data use** Validate that the data you are using has not been corrupted or interfered with.

Describe the Inclusiveness principle

Inclusiveness requires AI-based solutions to empower everyone and supply benefits to all parts of society, regardless of gender, physical ability, ethnicity, sexual orientation, or any other factors.

Inclusiveness means:

- Empowering everyone
- Engaging all communities in the world
- Intentionally designing for the Inclusiveness principle

Describe the Transparency principle

Transparency is the principle that AI-based solutions should be understandable. Users should be aware of the purpose of an AI-based system, how it operates, its scope, and its limitations.

Transparency is essentially about gaining the trust of users, and it means:

- Defining why your solution uses AI
- Defining the scope of AI in your solution
- Stating the limitations of AI in your solution

AI-based solutions should be understandable. You must be open with users and stakeholders about how and why you are using AI.

Transparency also means that people can understand the behavior of an AI system and interpret the outputs of the AI algorithm. For example, in some industries, you must be able to explain to regulators how the AI algorithm has generated its results.

Describe the Accountability principle

Accountability requires the people involved in designing and developing AI-based solutions to follow clearly defined ethical policies and legal standards and to operate within a clear governance framework due to the impact AI can have on the world.

Accountability requires that you have:

- Governance framework

- Ethical policies
- Legal standards

Civil liberties are an area where accountability is a crucial factor when determining if and how to use AI—for example, the use of facial recognition.

Need More Review? Responsible AI

For more information on Responsible AI, see www.microsoft.com/ai/responsible-ai.

Understand the Microsoft AI for Good program

AI for Good is a Microsoft program that puts AI technology, cloud software, and other resources in the hands of those working to create a more sustainable and accessible world.

AI for Good is enabling advances in healthcare, environmental protection, humanitarian action, cultural heritage, and other areas to make a better world for everyone.

More Info AI for Good

For more information on AI for Good, see www.microsoft.com/ai/ai-for-good.

Chapter summary

In this chapter, you learned some of the general concepts related to artificial intelligence. You learned about the features of common AI workloads, as well as the principles of Responsible AI. Here are the key concepts from this chapter:

- AI is a technology that mimics the human brain and uses machines to complete complex tasks that humans find time-consuming to do.
- Machine learning uses algorithms to discover patterns and structures in existing data, building a model that can be used to take actions (like classify and predict) on unseen data.
- Computer vision is the analysis of images and video to extract information that can be used by computers. Computer vision can interpret the contents of the image to classify the image, detect objects in the image, and analyze and describe the image.
- Document intelligence is the analysis of structured and unstructured documents to extract and organize data in a format that can be used by computers. Document intelligence can interpret the layout and content of documents such as forms, invoices, and contracts. It can identify key-value pairs, tables, handwritten and printed text, and recognize important fields.
- Natural language processing is the analysis of speech and text to extract the meaning and intent of words in a way that can be used by computers. Natural language processing can interpret text, analyze the text to determine the language used, determine the sentiment expressed, extract key phrases, and identify key entities and actions.
- Conversational AI is used to create applications where AI agents engage humans in conversations (dialogues). Conversational AI commonly appears as chatbots on websites.
- Generative AI is a type of AI that goes beyond analyzing data to create new content. These systems use foundation models trained on large datasets and are capable of analyzing and producing text, images, video, audio, and code.
- Responsible AI is the use of AI in solutions without having unintended negative impacts.
- Fairness is the principle that AI-based systems should treat all people fairly and reduce bias.
- Reliability and Safety is the principle that requires rigorous testing of AI-based systems to eliminate harm to human life.
- Privacy and Security is the principle that AI-based systems should be secure and respect privacy of personal data.
- Inclusiveness is the principle that AI-based solutions should empower everyone and supply benefits to all parts of society, regardless of any characteristics or factors.
- Transparency is the principle that AI-based solutions should be understandable, or interpretable. Transparency requires that AI-based systems have a defined purpose and scope and are clear on the limitations of the AI in the solution.
- Accountability is the principle that everyone at every stage in the development of AI-based systems is accountable for the impact that the systems may have.

Thought experiment

In this thought experiment, demonstrate your skills and knowledge of the topics covered in this chapter. You can find the answers in the next section.

You work for Contoso Medical Group (CMG), and your management is interested in using AI in your applications and operations. CMG manages and monitors drug trials, evaluating the efficacy of the treatments.

The CMG IT department is resource-constrained, and they do not have data scientists or skilled AI developers available.

Having timely and accurate responses from patients improves the accuracy of the analysis performed. CMG has created an app to capture and track a patient's daily symptoms. CMG has recently added the capability to take pictures to capture skin conditions. CMG is unable to analyze the images due to the volume of images being captured. CMG is concerned about the amount of data storage for these images, as well as controlling access to the images.

CMG receives a lot of patient history and prescription records that are keyed into CMG's computer systems. These paper records are important information used to track a patient's response to drugs and treatments.

The support department is unable to handle the many inquiries CMG receives. Customers are receiving inconsistent responses depending on whom they speak to and how they are accessing customer support, whether by phone, web, or mobile app.

Your manager has come to you asking for solutions that address these issues. Whatever solution you offer must consider that the medical data in this application is covered under HIPAA, and your manager wants CMG to retain all control of the data. Your manager also wants to carefully control costs.

You have decided that CMG can use AI, but there are several issues that you need to resolve before proceeding.

Answer the following questions:

1. Which AI workload should you use for the customer support department?
2. Which principle of Responsible AI should you employ to gain user trust in your bot?
3. Which AI workload should you use to analyze the images for skin conditions?
4. Which AI workload should be used to digitize and extract information from handwritten medical forms and prescriptions?
5. Which principle of Responsible AI protects a patient's personal information?
6. Which Responsible AI principle ensures that CMG's AI solution does not unfairly disadvantage any group of patients?
7. Which principle of Responsible AI requires rigorous testing of your AI-based app?

Thought experiment answers

This section contains the solutions to the thought experiment. Each answer explains why the answer choice is correct.

1. Conversational AI will allow simple inquiries to be handled by an automated bot. You can create a chatbot for your website, create an assistant for customer service agents, and even enable a bot in a mobile app.
2. The Transparency principle states that AI-based solutions should be understandable. Users should be aware of the purpose of the AI-based system, how it operates, and its scope and limitations. A chatbot should clearly tell the user that it is a bot and what it can and cannot do.
3. Computer vision allows you to analyze images. With existing images, you can train computer vision to classify images and determine the type of skin complaint.
4. Document intelligence is designed to process structured and unstructured documents. It can extract key data such as patient details, prescribed medications, and dosage instructions from scanned images or PDFs of medical records, enabling faster digitization and reducing manual data entry errors.
5. The Privacy and Security principle is concerned with keeping AI-based solutions secure and preventing personal data from being disclosed.
6. Fairness is the principle that ensures AI systems treat all individuals equitably and avoid bias. In healthcare, this means ensuring that models perform equally well across diverse patient populations, regardless of demographic or socioeconomic factors.
7. Reliability and Safety requires the rigorous testing of an AI-based system's functionality and deployment to ensure that it works as expected and to eliminate potential risk to human life.

Describe fundamental principles of machine learning on Azure

Machine learning (ML) is the current focus of AI in computer science. Machine learning focuses on identifying and making sense of the patterns and structures in data and using those patterns in software for reasoning and decision making. ML uses past experiences to make future predictions.

ML allows computers to consistently perform repetitive and well-defined tasks that are difficult to accomplish for humans. Over the past few years, machine learning algorithms have proved that computers can learn tasks that are tremendously complicated for machines and have demonstrated that ML can be employed in a wide range of scenarios and industries.

This chapter explains several machine learning algorithms, including clustering, classification, regression, deep learning techniques, and Transformer architecture. The chapter then explains how machine learning works in terms of organizing datasets and applying algorithms to train a machine learning model. The chapter then looks at the process of building a machine learning model and the tools available in Azure.

Skills covered in this chapter:

- [Skill 2.1: Identify common machine learning types](#)
- [Skill 2.2: Describe core machine learning concepts](#)
- [Skill 2.3: Identify core tasks in creating a machine learning solution](#)
- [Skill 2.4: Describe capabilities of no-code machine learning with Azure Machine Learning](#)

Skill 2.1: Identify common machine learning types

Machine learning requires lots of data to build and train models to make predictions and inferences based on the relationships in data. You can use machine learning to predict a new value based on historical values and trends, to categorize a new piece of data based on data the model has already seen, and to find similarities by discovering the patterns in the data.

Humans can often see patterns in small datasets with a few parameters. For example, consider [Figure 2-1](#): a small set of data regarding students studying for this exam.

Student	Background	Hours Studied	Completed labs	Score	Pass
Student A	Computer Science	5	No	500	No
Student B	Computer Science	6	Yes	600	No
Student C	Mathematics	8	Yes	650	No
Student D	History	10	No	623	No
Student E	Computer Science	12	Yes	733	Yes
Student F	Mathematics	15	No	697	No
Student G	Mathematics	18	No	727	Yes
Student H	Computer Science	20	Yes	850	Yes
Student I	History	20	Yes	715	Yes
Student J	English	25	No	780	Yes
Student K	English	30	No	767	Yes

FIGURE 2-1 Sample data

You can probably see a pattern that shows studying more hours leads to a higher exam score and passing the exam. However, can you see a pattern between the students' academic backgrounds and whether they pass or fail? Can you answer the question of how much completing the labs affects their score? What if you were to have more information about the student, and what if there were many more records of data? This is where machine learning can help.

This skill covers how to:

- [Understand machine learning model types](#)
- [Describe regression models](#)
- [Describe classification models](#)
- [Describe clustering models](#)
- [Describe deep learning techniques](#)
- [Describe features of the Transformer architecture](#)

Understand machine learning model types

The amount of data created by businesses, people, their devices, and applications in ordinary daily life has grown exponentially and will grow even more as sensors are embedded into machinery in factories, personal devices, and homes. This volume of data is such that we each can leverage it to improve the

way we make decisions and how we operate.

When you decide that you want to use machine learning, one of the first things is to decide which type of learning you will use in your model:

- Supervised
- Unsupervised
- Reinforcement

The type of learning determines how your model will use data to determine its outcome.

Supervised learning

In *supervised learning*, the existing data contains the desired outcome, called a *label* in machine learning. The labeled value is the output you want your model to determine for new data. A label can either be a value or a distinct category.

The other data that is supplied and that is used as inputs to the model are called *features*. A supervised learning model uses the features and label to train the model to fit the label to the features. After the model is trained, supplying the model with the features for new data will predict the value, or category, for the label.

Use supervised learning when you already have existing data that contains both the features and the label.

Unsupervised learning

In *unsupervised learning*, you do not have the outcome or label in the data. Instead, you use machine learning to determine the structure of the data and to look for commonalities or similarities in the data. Unsupervised learning separates the data based on the features.

Use unsupervised learning when you are trying to discover something about your data that you do not already know.

Reinforcement learning

Reinforcement learning uses feedback to improve the outcomes from the machine learning model. Reinforcement learning does not have labeled data. Reinforcement learning uses a computer program, an *agent*, to determine if the outcome is optimal or not and feeds that back into the model so it can learn from itself.

Use reinforcement learning when trying to model a task, such as building a model to play chess. This type of learning is commonly used in robotics.

Describe regression models

You will have probably used regression in school to draw a best fit line through a series of data points on a graph. Using the data from [Figure 2-1](#), the hours studied are plotted against the exam scores in [Figure 2-2](#).

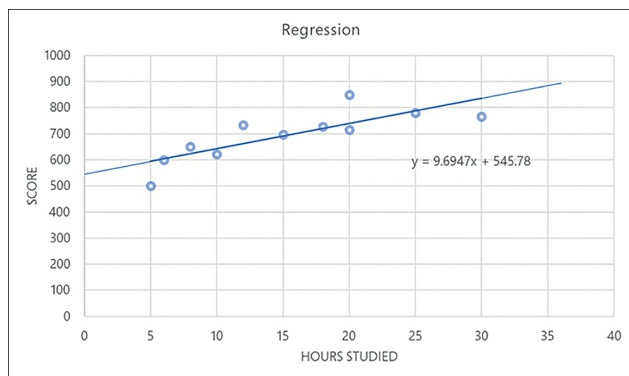


FIGURE 2-2 Regression graph

Regression is an example of supervised machine learning where the features (in this case, the hours studied) and the label (the exam score) are known and are both used to make the model fit the features to the label.

This graph is a very simple example of linear regression with just one feature. Regression models in machine learning can have many features. There are regression algorithms other than a simple linear regression that can be used.

Exam Tip

For this exam, you do not need to know about the various algorithms, but you must be able to differentiate between the three learning model types, as well as the regression, classification, and clustering models.

Regression is used to predict a numeric value using a formula that is derived from historic data. Regression predicts continuous values, not distinct categories. For [Figure 2-2](#), the model generated the formula $y = 9.6947x + 545.78$, which implies that every hour of studying increases the exam score by almost 10 points. Suppose you asked the model how many hours a student should study to pass the exam. It would predict 16 hours of studying would achieve a passing score of 700. (Microsoft exams require a score of at least 700 to pass.)

However, this is when you need to start considering how the data can affect the model. If a student who studied for 30 hours and scored 650 was added

to the data, the regression formula would change to $y = 6.7243x + 579.49$, as shown in Figure 2-3.

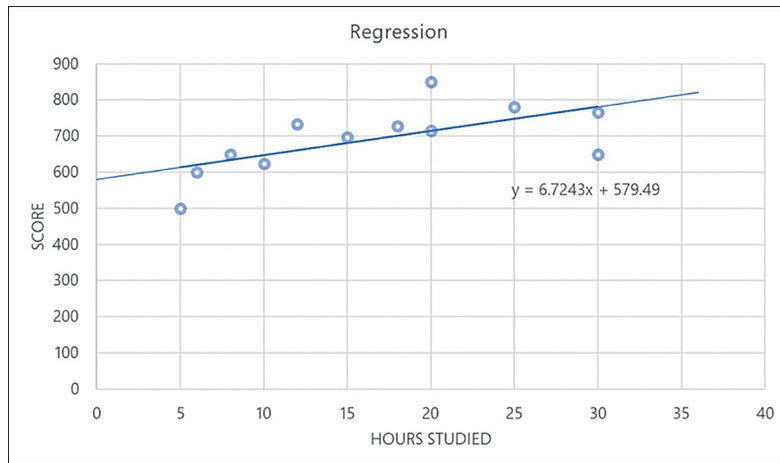


FIGURE 2-3 Regression graph with additional data

With this change to the model, the result changes to 18 hours of studying required to pass the exam. In machine learning, one of the major concerns is how data can bias the model (for more detail see the “Bias” section later in this chapter).

Describe classification models

Classification machine learning models are used to predict mutually exclusive categories, or classes. *Classification* involves learning that uses labels to classify data and is an example of supervised machine learning.

Classification is used to make predictions when you do not require continuous values but need distinct categories, such as Pass or Fail.

Using the data from Figure 2-3, you could build and train a classification model to use the hours studied to predict whether a student passes the exam or not. Using this data, a simple two-class model would likely predict that someone studying for less than 18 hours will fail and 18 hours or more will pass.

In a classification model, you can compare the actual labels with the prediction from the model, as shown in the table in Figure 2-4.

Student	Hours Studied	Pass/Fail		Result
		Actual	Model	
Student A	5	No	No	True Negative
Student B	6	No	No	True Negative
Student C	8	No	No	True Negative
Student D	10	No	No	True Negative
Student E	12	Yes	No	False Negative
Student F	15	No	No	True Negative
Student G	18	Yes	Yes	True Positive
Student H	20	Yes	Yes	True Positive
Student I	20	Yes	Yes	True Positive
Student J	25	Yes	Yes	True Positive
Student K	30	Yes	Yes	True Positive
Student L	30	No	Yes	False Positive

FIGURE 2-4 Classification model

You can see that the classification model correctly predicts all but two of the results. If the model predicts a pass and the actual is a pass, this is a true positive. If the model predicts a fail and the actual is a fail, this is a true negative.

To refine a classification model, you need to understand where the model gets it wrong. For example, the model predicts a pass for Student L, but the actual result was a fail—this is a false positive. Student E actually passed, but the model predicted that the student will fail—this is a false negative.

Describe clustering models

Clustering machine models learn by discovering similarities, patterns, and relationships in the data without the data being labeled. *Clustering* is an example of unsupervised learning where the model attempts to discover structure from the data or tell you something about the data that you didn’t know.

Clustering analyzes unlabeled data to find similarities in data points and groups them together into clusters. A clustering algorithm could be used, for example, to segment customers into multiple groups based on similarities in the customer’s details and history.

A clustering model predicts mutually exclusive categories, or *classes*. *K-means clustering* is a common clustering model where K is the number of distinct clusters you want the model to group the data by. The way clustering works is to calculate the distance between the data point and the center of the cluster, and then to minimize the distance of each data point to the center of its cluster.

Consider the sample data but assume no one has taken the exam yet, so you do not have the scores or pass/fail. Instead, you have unlabeled data. To determine if a relationship exists between the background of the students and the hours studied, you could plot these as shown in Figure 2-5.

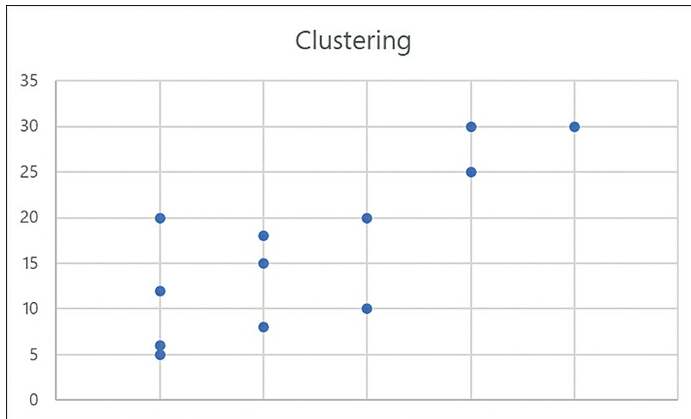


FIGURE 2-5 Clustering data

If you were to create a clustering model with $K = 3$, then it might group the data into the three clusters—A, B, and C—as shown in Figure 2-6.

A common example of a clustering model is the recommender model. This is the model that is shown in Figure 2-6 for a company that wants to provide recommendations to its users by generating personalized targeted recommendations. A *recommender model* looks for similarities between customers. For example, a video streaming service knows which movies customers watch and can group customers by the types of movies they watch. A customer can then be shown other movies watched by other customers with similar viewing histories.

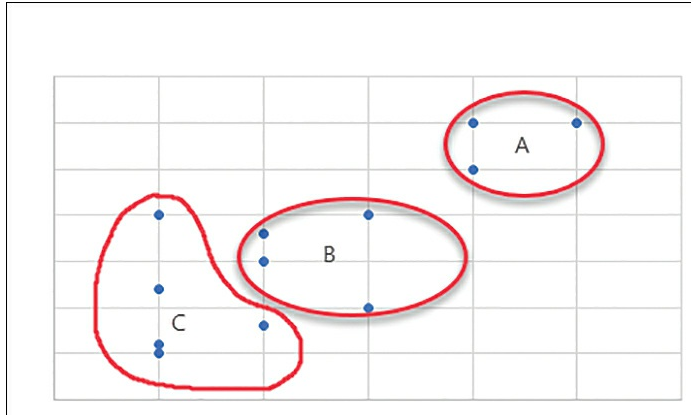


FIGURE 2-6 Clustering model

Describe deep learning techniques

Deep learning is a machine learning technique inspired by the way the human brain learns. It is based on the use of artificial neural networks, which are designed to emulate biological neurons' behavior. Both biological and neural neurons are connected with other neurons in multilayered structures, and these connections ensure signal propagation. In contrast to biological neurons, which activate and send signals to connected neurons when stimulated, a *neuron* in neural networks applies a mathematical function to an input value, combined with a weight, and passes the result through an activation function to decide whether to propagate it to the next layer.

Complex tasks such as natural language processing and image recognition are made possible by deep learning techniques, although they can also be used to solve traditional machine learning problems like classification and regression.

Just like supervised learning models, deep learning models learn from labeled data by finding a function that given input features (x) can predict the corresponding output label (y). What makes deep learning different, often outperforming traditional approaches when large amounts of data and computational resources are available, is the complexity of this learning function resulting from the combination of nested functions of the layers of the network.

During training, the model takes input data and passes it through the network to produce a prediction (\hat{y}), compares this with the known true label (y), and calculates an error (known as the *loss function*) based on their difference. Then, through a process called *backpropagation*, the model updates the weights associated with each neuron to minimize this loss. Figure 2-7 illustrates this training mechanism.

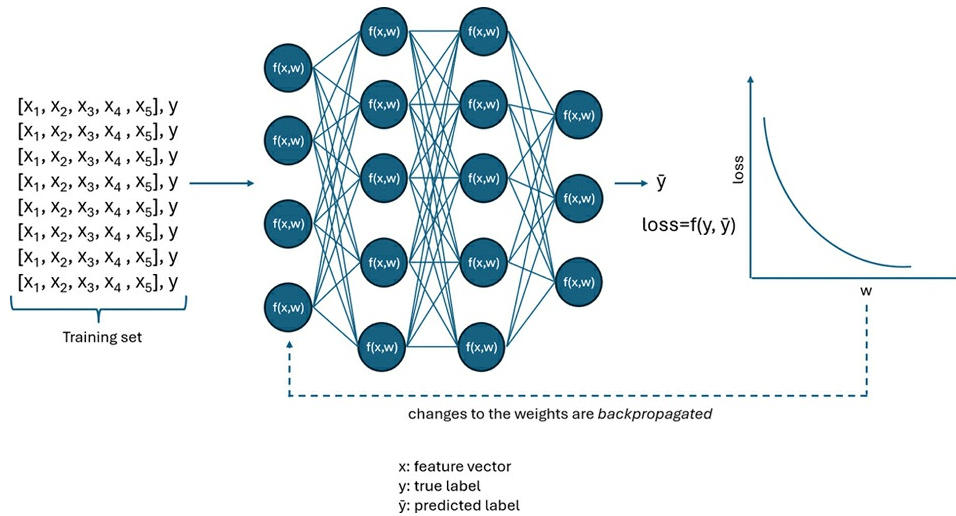


FIGURE 2-7 Deep learning models training

To better understand how deep learning works, consider a deep learning classification model trained to classify the species of a bird based on such quantifiable features as color, wing length, and body weight. Each observation of a bird is represented as a feature vector with a numerical value for each characteristic considered and is passed into the input layer of the neural network.

Each neuron of the input layer computes a weighted sum of the inputs using weights learned during training (w) and passes the result to the activation function, which determines whether it meets the threshold to be passed to the next layer. Repeating this process, data transformation at each layer is propagated through the network to the output layer.

To determine the bird class, when data reaches the output layer of the neural network, a function like SoftMax is applied to convert its results into a set of probabilities for each of the possible categories. The class with the highest probability is selected as the prediction.

Note Neural Network Training Performance

Although it's easier to imagine training data being processed one item at a time, in practice the data is grouped into batches and processed as matrices using linear algebra. Because of this, training deep learning models is significantly faster on machines equipped with GPUs (graphic processing units), which are specifically designed to handle large-scale matrix and vector operations efficiently.

Describe features of the Transformer architecture

Another breakthrough in the evolution of machine learning, especially in natural language processing, has been the development of Transformer models. These models form the underlying structure of today's large language models (LLMs). What makes Transformers so revolutionary is their ability to model semantic relationships between words and their consequent ability to generate text that is hard to distinguish from human writing.

Transformer architecture is based on two main multilayered networks: an encoder and a decoder.

The *encoder* receives text as input, breaks it into tokens (which may be words or parts of words), and generates vector representations of the input text known as *embeddings*.

Embeddings capture the meaning of every token in relation to its use. Considering a high-dimensional semantic space, words with similar meanings are represented by vectors that have similar directions in this space. Cosine distance is usually used to determine, in this semantic space, how similar two words are.

The *decoder* uses these embeddings to generate new sequences of text, predicting one token at a time to form coherent responses. Figure 2-8 provides a simplified schematic illustration of the process.

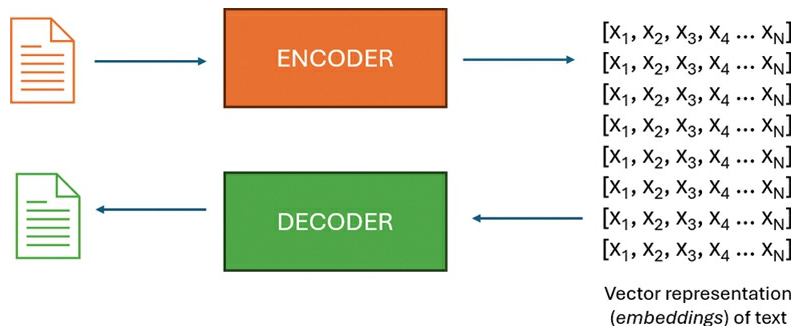


FIGURE 2-8 Basic schema of encoder and decoder

The reason why Transformer architecture outperforms earlier machine learning models for natural language processing, such as RNNs (recurrent neural networks) and LSTM (long short term memory), is the attention mechanism. The *attention mechanism* captures the relationships and dependencies between

tokens within a sentence, regardless of their distance from each other. This enables Transformers to learn semantic rules more deeply than previous neural networks did. In modern architectures, the so-called *self-attention* mechanism is used, which computes how much each token in the input sequence should attend to (or influence) every other token (including itself) to build a contextualized representation of meaning.

Note Attention Versus Self-Attention

Attention refers to a mechanism that lets a model focus on relevant parts of another sequence, while self-attention is a special case in which the model attends to different parts of the same sequence to understand the context of each element.

Both the encoder and the decoder use attention layers.

In the encoder block, the attention mechanism is used in the embeddings' generation process. It allows the creation of vectors with regard to all the relationships and dependencies within the input text, regardless of the distance between words. This context-sensitive approach allows the same word to have different embeddings depending on the context in which it is used. To further improve embedding generation, each token embedding is enriched with positional encoding, representing the position of the token in the sequence.

In the decoder block, the attention mechanism is used in the process of predicting the next token in a sentence. It evaluates the previous tokens' embeddings to determine the most influential tokens for the prediction of the next token. The attention layer assigns weights to each token (precisely: to each token's embedding), generating an attention score that influences the prediction of the next token. While training, the model attempts to predict the next token based on past tokens.

In practice, a technique called *multi-head attention* applies multiple attention mechanisms in parallel on different learned projections of each token's embedding, allowing the model to capture diverse contextual relationships.

The output of each step is passed through a SoftMax layer to produce a probability distribution over the vocabulary, from which the most likely next token is sampled or selected. This process repeats iteratively, using the previously generated tokens as input to generate the sequence one token at a time. This iterative process continues until the sequence is complete, generating an EOS (end of sequence) token.

As in any neural network, the prediction of the output is compared to the actual value, and the loss is calculated. The weights are then incrementally adjusted to reduce the loss and improve the model. Through this training, the model learns not only individual word meanings but also how meanings shift depending on context.

Transformers' power lies in the combination of massive training data, the use of embeddings to capture semantic relationships, and the attention mechanism to model contextual meaning. This is what enables LLMs like GPT-4 to produce human-like responses.

Note Different Implementations of Transformers

There are different implementations of the Transformer architecture depending on the intended task. For instance, BERT (Bidirectional Encoder Representations from Transformers) uses only the encoder component and is optimized for tasks such as classification and information retrieval. In contrast, GPT (Generative Pretrained Transformer) uses only the decoder and is designed to generate natural language, predicting the most likely next token given a preceding sequence.

Skill 2.2: Describe core machine learning concepts

Machine learning has several common concepts that apply when building machine learning models. These concepts apply no matter which tools, languages, or frameworks you use. This section explains the fundamental concepts and processes involved in building machine learning models.

This skill covers how to:

- Understand the machine learning workflow
- Identify the features and labels in a dataset for machine learning
- Describe how training and validation datasets are used in machine learning
- Describe how machine learning algorithms are used for model training
- Select and interpret model evaluation metrics

Understand the machine learning workflow

Building a machine learning model follows the process outlined in [Figure 2-9](#). It is important to note that building a model is an iterative process in which the model is evaluated and refined.

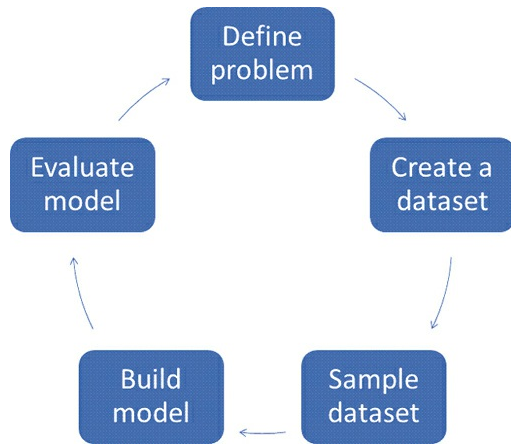


FIGURE 2-9 Machine learning workflow

First, you define the problem. This means translating the business problem into a machine learning problem statement. For example, if you were asked to understand how groups of customers behave, you would transform that to “create a clustering model using customer data.”

Think back to the example: How much activity does a student need to undertake to pass the exam with at least a 700 score? So far, you have just considered the hours studied, but suppose you have been asked to look at other factors, such as whether students have completed the labs and their choice of degree subject. You could transform these into the following problem statements:

- Create a regression model to predict a student’s score for the exam using their degree subject and their exam preparation activities.
- Create a classification model to predict if a student will pass the exam using their degree subject and their exam preparation activities.

Identify the features and labels in a dataset for machine learning

The next step in the machine learning workflow is to create a dataset. Data is the most important asset you have, because you use data to train your model. If your data is incomplete or inaccurate, then it will have a major impact on how well your model performs.

You must first collect your data. This can mean extracting data from multiple systems, transforming, cleansing, and importing the data.

Exam Tip

How to extract and transform data is outside the scope of this exam.

Figure 2-10 adds some additional exam results to the dataset you used earlier in this chapter.

Student	Background	Hours Studied	Completed Labs	Score	Pass
Student A	Computer Science	5	No	500	No
Student B	Computer Science	6	Yes	600	No
Student C	Mathematics	8	Yes	650	No
Student D	History	10	No	623	No
Student E	Computer Science	12	Yes	733	Yes
Student F	Mathematics	15	No	697	No
Student G	Mathematics	18	No	727	Yes
Student H	Computer Science	20	Yes	850	Yes
Student I	History	20	Yes	715	Yes
Student J	English	25	No	780	Yes
Student K	English	30	No	767	Yes
Student L	Psychology	30	No	650	No
Student M	Computer Science	20	No	727	Yes
Student N	Mathematics	17	Yes	600	No

FIGURE 2-10 Dataset

Identify labels

If you are using supervised training—for example, a regression or a classification model—then you need to select the label(s) from your dataset. *Labels* are the columns in the dataset that the model predicts.

For a regression model, you would choose the Score column as the label, because this is a numeric value. Regression models are used to predict a range of values.

For a classification model, you would choose the Pass column for the label, because this column has distinct values. Classification models are used to predict from a list of distinct categories.

Feature selection

A *feature* is a column in your dataset. You use features to train the model to fit the label. After training the model, you can supply new data containing the same features, and the model will predict the value for the column you have selected as the label.

The possible features in the example dataset are the columns:

- Background
- Hours Studied
- Completed Labs

In the real world, you will have other possible features to choose from.

Feature selection is the process of selecting a subset of relevant features to use when building and training the model. Feature selection restricts the data to the most valuable inputs, reducing noise and improving training performance.

Feature engineering

Feature engineering is the process of creating new features from raw data to increase the predictive power of the machine learning model. Engineered features capture additional information that is not available in the original feature set.

Examples of feature engineering are:

- Aggregating data
- Calculating a moving average
- Calculating the difference over time
- Converting text into a numeric value
- Grouping data

Models train better with numeric data rather than text strings. In some circumstances, data that visually appears to be numeric may be held as text strings, and you need to parse and convert the data type into a numeric value. For example, the degree subject names in the example dataset's Background column may not perform well when you evaluate the model. One option might be to classify the degree subjects into humanities and sciences and then to convert to a Boolean value, such as `IsScienceSubject`, with values of 1 for true and 0 for false.

Bias

Bias in machine learning is the impact of erroneous assumptions that a model makes about its data. Machine learning models depend on the quality, objectivity, and quantity of data used to train it. Faulty, incomplete, or prejudicial data can result in a poorly performing model.

Chapter 1, “Describe artificial intelligence workloads and considerations,” introduced the Fairness principle and how an AI model should be concerned with how data influences the model's prediction to help eliminate bias. To adhere to this principle, you should be conscious of the provenance of the data you are using in your model. You should evaluate the bias that might be introduced by the data you have selected.

A common issue is that the algorithm is unable to learn the true signal from a dataset, and instead, noise in the data overly influences the model. Consider an example from computer vision: An army attempted to build a model that was able to find enemy tanks in photographs of landscapes. The model was built with many photographs, some with and without tanks in them. The model performed well in testing and evaluation, but when deployed, the model was unable to find tanks. Eventually, it was realized that all pictures of tanks were taken on cloudy days, and all pictures without tanks were taken on sunny days. They inadvertently had built a model that identified whether a photograph was of a sunny or a cloudy day; the noise of the sunlight biased the model. The problem of bias was resolved by adding additional photographs into the dataset with varying degrees of cloud cover.

It can be tempting to select all columns as features for your model. You may then find when you evaluate the model that one column significantly biases the model, with the model effectively ignoring the other columns. You should consider removing that column as a feature if it is irrelevant.

Normalization

A common cause of bias in a model is caused by data in numeric features having different ranges of values. Machine learning algorithms tend to be influenced by the size of values, so if one feature ranges in values between 1 and 10 and another feature between 1 and 100, the latter column will bias the model toward that feature. You can mitigate possible bias by normalizing the numeric features, so they are on the same numeric scale.

After feature selection, feature engineering, and normalization, the example dataset might appear as in [Figure 2-11](#).

ID	Feature	Feature	Feature	Label	Label
Student	IScienceSubject	Hours Studied	Completed Labs	Score	Pass
Student A	1	0.1667	0	500	0
Student B	1	0.2000	1	600	0
Student C	1	0.2667	1	650	0
Student D	0	0.3333	0	623	0
Student E	1	0.4000	1	733	1
Student F	1	0.5000	0	697	0
Student G	1	0.6000	0	727	1
Student H	1	0.6667	1	850	1
Student I	0	0.6667	1	715	1
Student J	0	0.8333	0	780	1
Student K	0	1.0000	0	767	1
Student L	0	1.0000	0	650	0
Student M	1	0.6667	0	727	1
Student N	1	0.5667	1	600	0

FIGURE 2-11 Normalized dataset

Describe how training and validation datasets are used in machine learning

After you have created your dataset, you need to create sample datasets for use in training and evaluating your model.

Typically, you split your dataset into two datasets when building a machine learning model:

- **Training** The training dataset is the sample of data used to train the model. It is the largest sample of data used when creating a machine learning model.
- **Testing** The testing, or validation, dataset is a second sample of data used to provide a validation of the model to see if the model can correctly predict, or classify, using data not seen before.

A common ratio between training and validation data volumes is 70:30, but you may vary this ratio depending on your needs and size of your data.

You need to be careful when splitting the data. If you take simply the first set of rows, then you may bias your data by date created or however the data is sorted. You should randomize the selected data so that both training and testing datasets are representative of the entire dataset.

For example, you might split the example dataset as shown in Figure 2-12.

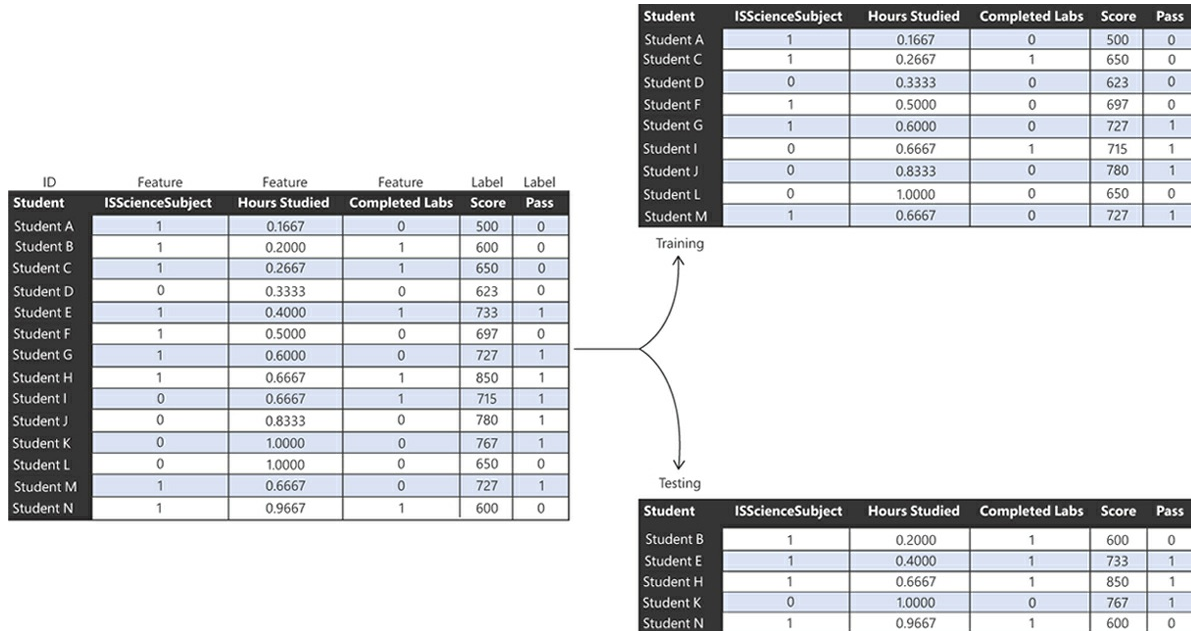


FIGURE 2-12 Training and testing datasets

Describe how machine learning algorithms are used for model training

A machine learning model learns the relationships between the features and the label in the training dataset.

It is at this point that you select the algorithm to train the model with.

Note Algorithms

For this exam, you do not need to know about algorithms or which algorithm should be used for which machine learning problem. If you want to find out more about algorithms, see aka.ms/mlcheatsheet.

The algorithm finds patterns and relationships in the training data that map the input data features to the label that you want to predict. The algorithm outputs a machine learning model that captures these patterns.

Training a model can take a significant amount of time and processing power. Data scientists use the scalability of cloud storage to build models more quickly and with more data than can be achieved with on-premises hardware.

After training, you use the model to predict the label based on its features. You provide the model with new input containing the features (Hours Studied, Completed Labs) and the model will return the predicted label (Score or Pass) for that student.

Select and interpret model evaluation metrics

After a model has been trained, you need to evaluate how well the model has performed. To do this, you need to score the model with data that it hasn't seen and that wasn't used to build it.

To evaluate the model, use the data that you earlier split from the testing dataset and compare the prediction values for the label with the known actual values to obtain a measure of the amount of error. Next, create metrics to help gauge the performance of the model and explore the results.

The model type you used—regression, classification, or clustering—will determine the best method to measure and evaluate your model.

Evaluate regression models

When evaluating a regression model, you estimate the amount of error in the predicted values.

To determine the amount of error in a regression model, measure the difference between the actual values you have for the label and the predicted values for the label. These are known as the *residual values*. A way to represent the amount of error is to draw a line from each data point perpendicular to the best fit line, as shown in [Figure 2-13](#).



FIGURE 2-13 Regression errors

The length of the lines indicates the size of residual values in the model. A model is considered to fit the data well if the difference between actual and predicted values is small.

The following metrics can be used when evaluating regression models:

- **Mean absolute error (MAE)** Measures how close the predictions are to the actual values; lower is better
- **Root mean squared error (RMSE)** The square root of the average squared distance between the actual and the predicted values; lower is better
- **Relative absolute error (RAE)** Relative absolute difference between expected and actual values; lower is better
- **Relative squared error (RSE)** The total squared error of the predictions divided by the total variance of the actual values (with respect to their mean); lower is better
- **Mean zero one error (MZOE)** If the prediction was correct or not with values 0 or 1
- **Coefficient of determination (R2 or R-squared)** A measure of the variance from the mean in its predictions; the closer to 1, the better

Later you will learn how Azure Machine Learning calculates these metrics for you.

Evaluate classification models

In a classification model with distinct categories predicted, you are interested when the model gets it right or gets it wrong.

A common way to represent how a classification model is right or wrong is to create a confusion matrix. In a *confusion matrix*, the numbers of true positives, true negatives, false positives, and false negatives are shown:

- **True positive** The model predicted true, and the actual is true.

- **True negative** The model predicted false, and the actual is false.
- **False positive** The model predicted true, and the actual is false.
- **False negative** The model predicted negative, and the actual is true.

The total number of true positives is shown in the top-left corner, and the total number of true negatives is shown in the bottom-right corner. The total number of false positives is shown in the top-right corner, and the total number of false negatives is shown in the bottom-left corner, as you can see in [Figure 2-14](#).

		Actual	
		1	0
Predicted	1	6	1
	0	1	6

FIGURE 2-14 Confusion matrix

Note Confusion Matrix

Confusion matrices are not always shown in the same way. For instance, some authors or tools may swap the actual and predicted headings, and the 1 and 0 could be reversed.

From the values in the confusion matrix, you can calculate metrics to measure the performance of the model:

- **Accuracy** The number of true positives and true negatives; the total of correct predictions, divided by the total number of predictions
- **Precision** The number of true positives divided by the sum of the number of true positives and false positives
- **Recall** The number of true positives divided by the sum of the number of true positives and false negatives
- **F-score** Combines precision and recall as a weighted mean value
- **Area under curve (AUC)** A measure of true positive rate over true negative rate

All these metrics are scored between 0 and 1, with closer to 1 being better.

As you will see later, Azure Machine Learning generates the confusion matrix and calculates these metrics for you.

Evaluate clustering models

Clustering models are created by minimizing the distance of a data point to the center point of its cluster.

When evaluating cluster models, you use the average distance to cluster Center metric. It is a measure of how focused the clusters are and the lower the value, the better.

Skill 2.3: Identify core tasks in creating a machine learning solution

Azure provides several tools and services to build and manage machine learning models. While the tools vary, many of the tasks involved are very similar. This section describes how you use Azure Machine Learning to build and deploy machine learning models.

This skill covers how to:

- [Understand machine learning on Azure](#)
- [Understand Azure Machine Learning Studio](#)
- [Describe data ingestion and preparation](#)
- [Describe feature selection and engineering](#)
- [Describe model training and evaluation](#)
- [Describe model deployment and management](#)

Understand machine learning on Azure

Azure provides many ways to create and use artificial intelligence (AI) models. You can use the prebuilt models in Azure AI Services, or you can build and deploy your own models with Azure Machine Learning services.

Machine learning on Azure

Microsoft provides a number of services created by Microsoft for machine learning and supports a wider set of open source and third-party services for data science and AI that you can use within Azure for your own AI solutions.

The Azure Marketplace contains services and solutions for machine learning from both Microsoft and its partners, as shown in [Figure 2-15](#).

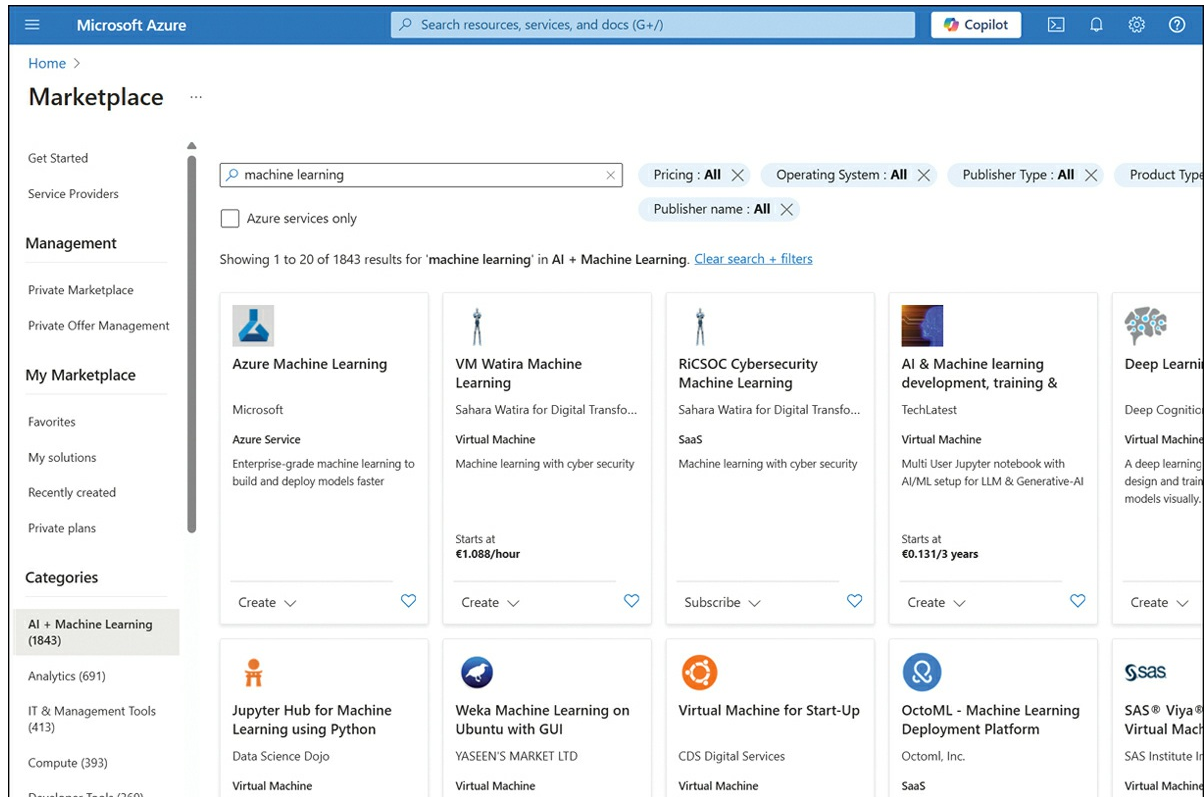


FIGURE 2-15 Azure Marketplace

If you are used to tools such as PyCharm or Jupyter notebooks, you can use these within Azure and leverage other Azure services such as compute and storage.

If you are used to frameworks such as PyTorch, scikit-learn, TensorFlow, or ONNX (Open Neural Network Exchange), you can use these frameworks within Azure.

If you are used to using Apache Spark, you can use Azure Databricks, the Microsoft implementation of Apache Spark that integrates tightly with other Azure services.

If you are used to Microsoft SQL Server machine learning, you can use an Azure Data Science virtual machine (DSVM), which comes with ML tools and R and Python installed.

You can also configure and use your own virtual machine that includes the tools and services that you prefer.

Many options are available, but this chapter will focus on the native Azure services provided by Microsoft.

Azure Machine Learning workspace

To use Azure Machine Learning services, you first need access to an Azure subscription. An Azure subscription defines how you are billed for using Azure resources.

You then need to create an Azure Machine Learning workspace. An Azure ML workspace is a container for all of your resources used in building and deploying models, data, experiments, compute, and web services.

The purpose of a Machine Learning workspace is to make the building, training, and deployment of machine learning models easier, reducing the steps required and integrating storage and compute all within a secure environment.

[Figure 2-16](#) shows the service description for Azure Machine Learning workspace.

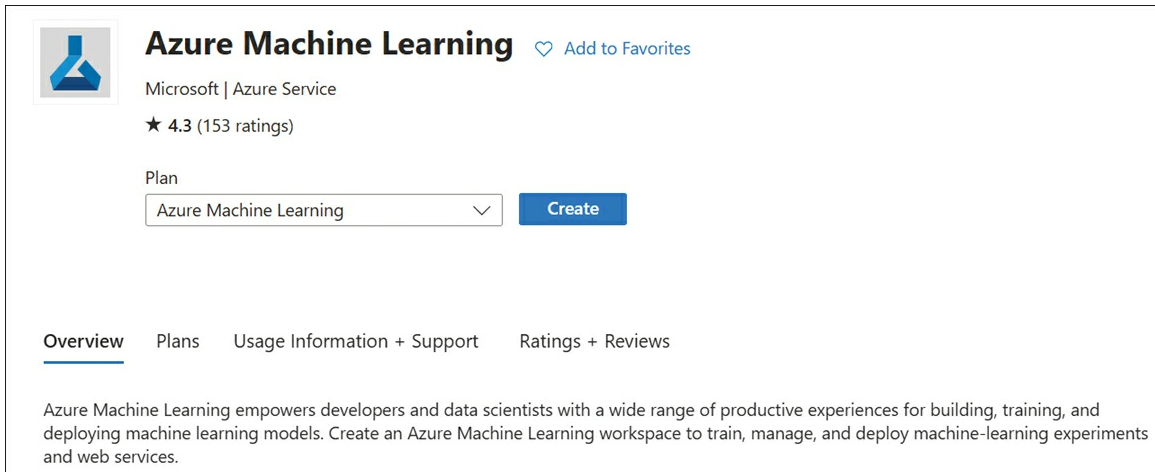


FIGURE 2-16 Machine Learning workspace service description

Click the Create button, to open the Create Machine Learning workspace, as shown in Figure 2-17.

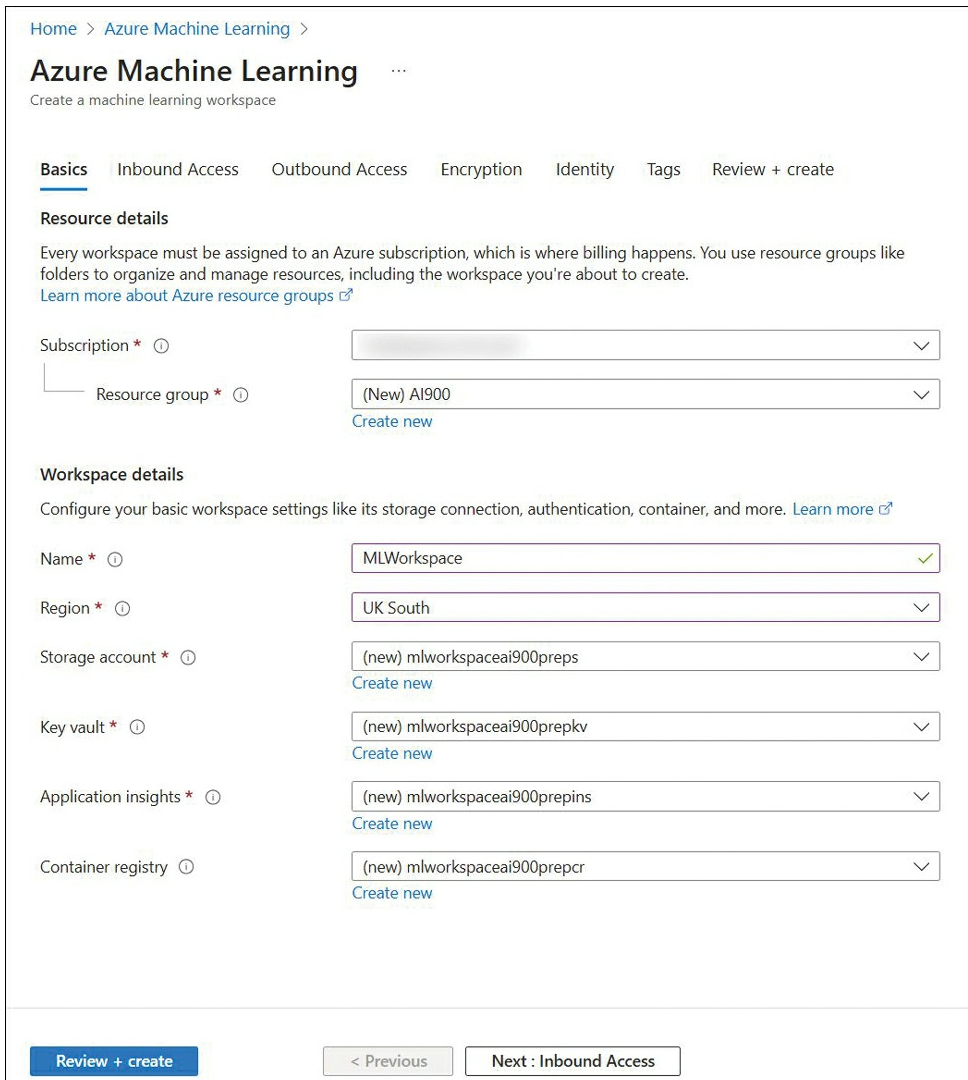


FIGURE 2-17 Creating a Machine Learning workspace resource

Here you can select the subscription, resource group, and region where you want to deploy the resource. Next, create a unique name for the workspace.

There are four settings related to Azure services:

- **Storage Account** The default datastore for the workspace for data used to train models as well as files used and generated by the workspace
- **Key Vault** Securely stores secrets such as authentication keys and credentials that are used by the workspace
- **Application Insights** Stores monitoring information for your deployed models
- **Container Registry** Stores Docker images used in training and deployments

You can either select existing resources or create new resources for these related services.

Click Review + Create to validate the options, then click Create to create the resource. The resource will be deployed after a few minutes.

Once your resource has been created, you can view the resources associated with the workspace, as shown in [Figure 2-18](#).

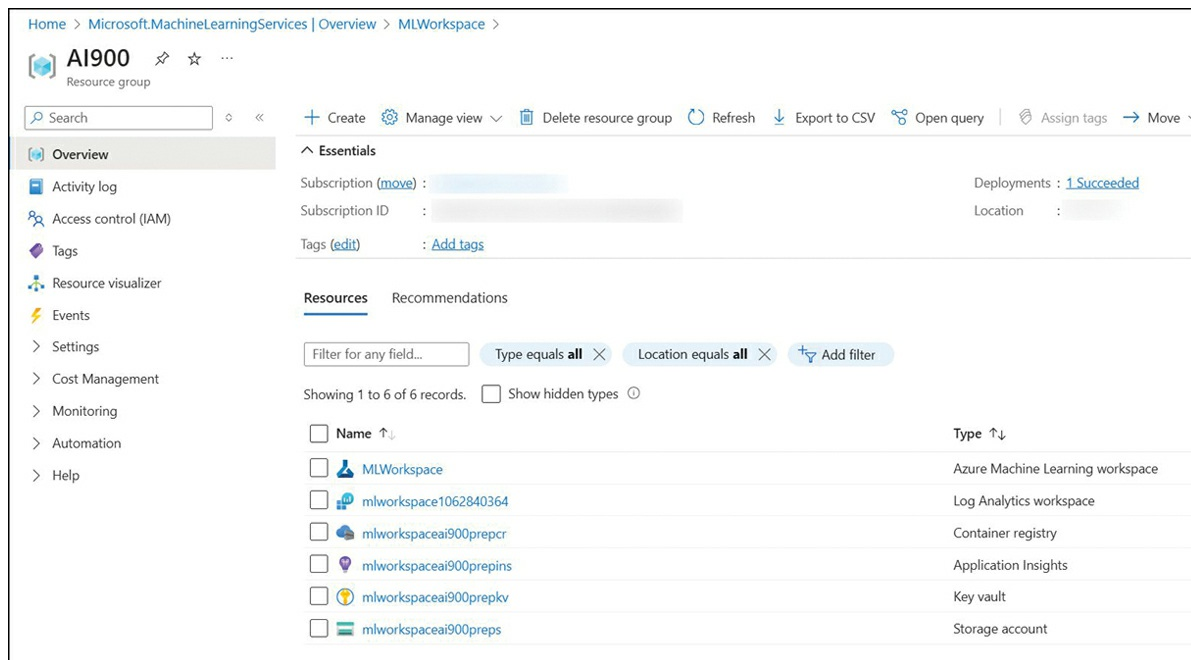


FIGURE 2-18 Azure resources for a Machine Learning workspace

You can create a Machine Learning workspace using the CLI as follows:

```
az ml workspace create --name <workspace_name> --resource-group <resource_group_name> --location <region>
```

To access the workspace, you need to open the workspace in the portal. This will display the details of the workspace, as shown in [Figure 2-19](#).

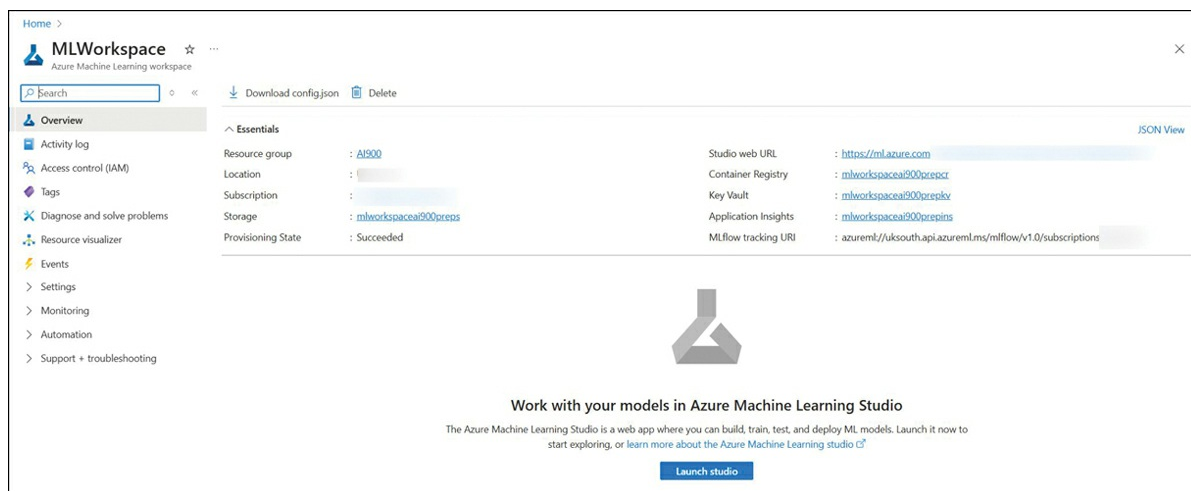


FIGURE 2-19 Machine Learning workspace resource

You can use Role-Based Access Control (RBAC) to give users permission to access the workspace and its resources, as well as create and run experiments.

An Azure Machine Learning workspace is used to manage your machine learning assets:

- **Compute** The virtual machines and Docker containers used for development, training, and deployment of models.
- **Datasets** The data used for experimentation and training models.
- **Pipelines** The connected steps for a machine learning model. A pipeline can contain the data ingestion, training, scoring, and evaluation steps.
- **Experiments** The results for each run of a pipeline, including metrics and outputs.
- **Models** The trained models.
- **Endpoints** The web services for deployed models.

Before you can start working with your workspace, you need to assign compute resources to the workspace. These are referred to as *compute targets*:

- **Compute instance** A development virtual machine used to process data and build your model.
- **Compute cluster** A cluster of scalable virtual machines for training your model and running experiments.
- **Kubernetes cluster** Azure Kubernetes Service (AKS) cluster for running a deployed model to perform predictions.
- **Attached compute** Existing Azure virtual machines or other ML services, such as Azure Databricks.
- **Serverless instance** An automatically managed compute resource used during prompt flow tasks such as authoring, batch runs, and evaluation. It requires no manual setup and scales dynamically based on workload.

Note Compute

You need compute targets to run machine learning workloads in the cloud instead of on your local computer. You can perform model development on local compute with low volumes of data using Visual Studio Code.

The high-level process for an Azure Machine Learning workspace is shown in [Figure 2-20](#).

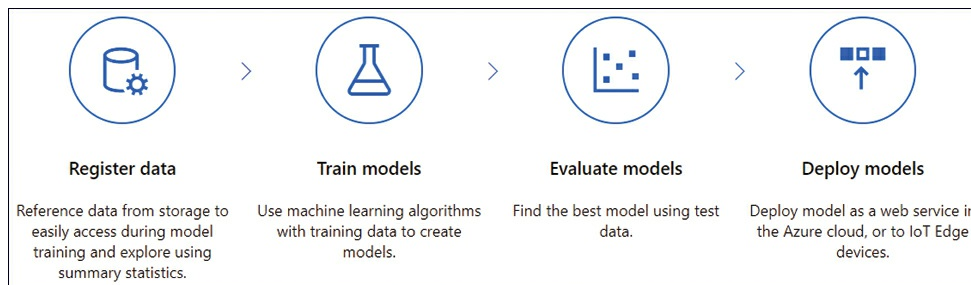
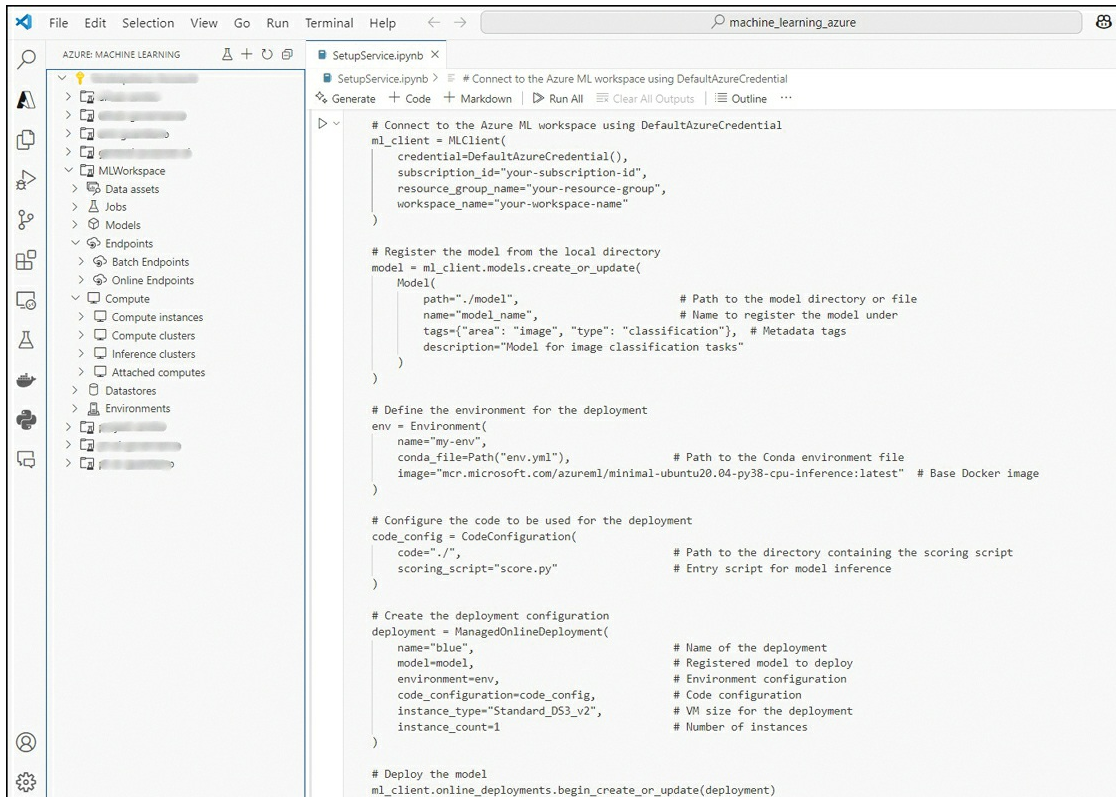


FIGURE 2-20 Machine Learning workspace process

You need to manage the data needed to build machine learning solutions. You can then build and train your model. All the outputs of each experiment are stored in the workspace to allow you to evaluate the model. When you are confident that your model performs well, you publish your model as a web service that can be accessed via an HTTP endpoint.

Azure Machine Learning workspaces are tightly integrated with the Visual Studio Code IDE. Using Visual Studio Code you can build, train, and deploy machine learning models with Azure Machine Learning. Visual Studio Code supports creating machine learning models with .NET and R languages, as well as with Python and Jupyter notebooks. [Figure 2-21](#) shows the Visual Studio Code connected to the Azure Machine Learning workspace and running Python code in a Jupyter notebook.



The screenshot shows the Visual Studio Code interface with a file explorer on the left and a code editor on the right. The file explorer displays the 'AZURE MACHINE LEARNING' workspace structure, including folders for 'Data assets', 'Jobs', 'Models', 'Endpoints', 'Batch Endpoints', 'Online Endpoints', 'Compute', 'Compute instances', 'Compute clusters', 'Inference clusters', 'Attached computes', 'Datastores', and 'Environments'. The code editor displays a Python script named 'SetupService.ipynb' with the following content:

```
SetupService.ipynb > # Connect to the Azure ML workspace using DefaultAzureCredential
SetupService.ipynb > # Connect to the Azure ML workspace using DefaultAzureCredential
Generate + Code + Markdown | Run All | Clear All Outputs | Outline ...

# Connect to the Azure ML workspace using DefaultAzureCredential
ml_client = MLClient(
    credential=DefaultAzureCredential(),
    subscription_id="your-subscription-id",
    resource_group_name="your-resource-group",
    workspace_name="your-workspace-name"
)

# Register the model from the local directory
model = ml_client.models.create_or_update(
    Model(
        path="./model", # Path to the model directory or file
        name="model_name", # Name to register the model under
        tags={"area": "image", "type": "classification"}, # Metadata tags
        description="Model for image classification tasks"
    )
)

# Define the environment for the deployment
env = Environment(
    name="my-env",
    conda_file=Path("env.yml"), # Path to the Conda environment file
    image="mcr.microsoft.com/azureml/minimal-ubuntu20.04-py38-cpu-inference:latest" # Base Docker image
)

# Configure the code to be used for the deployment
code_config = CodeConfiguration(
    code=".", # Path to the directory containing the scoring script
    scoring_script="score.py" # Entry script for model inference
)

# Create the deployment configuration
deployment = ManagedOnlineDeployment(
    name="blue", # Name of the deployment
    model=model, # Registered model to deploy
    environment=env, # Environment configuration
    code_configuration=code_config, # Code configuration
    instance_type="Standard_DS3_v2", # VM size for the deployment
    instance_count=1 # Number of instances
)

# Deploy the model
ml_client.online_deployments.begin_create_or_update(deployment)
```

FIGURE 2-21 Visual Studio Code



Exam Tip

For this exam, you do not need to know how to write code for machine learning, but you do need to know about the different languages and tools for building models.

Understand Azure Machine Learning Studio

Clicking the Launch Studio button in the Machine Learning workspace resource opens the Machine Learning Studio at ml.azure.com, shown in [Figure 2-22](#).

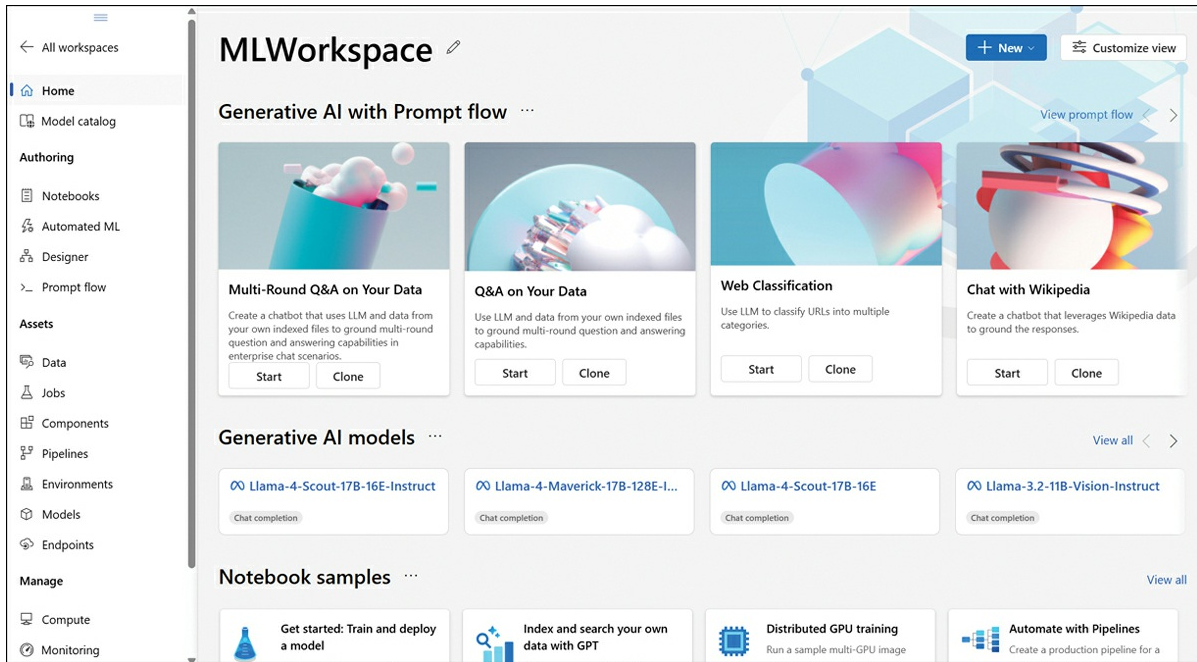


FIGURE 2-22 Azure Machine Learning Studio

Machine Learning Studio allows you to create and manage the assets in your Machine Learning workspace using a graphical user interface.

Authoring

Azure Machine Learning Studio supports both no-code and code-first experiences. You can build, train, and run machine learning models with Automated Machine Learning, notebooks, and Designer (a visual no-code tool).

Azure Machine Learning Studio supports the use of Jupyter notebooks that use the Python SDK to create and run machine learning models.

Automated Machine Learning (Automated ML) is a no-code tool that performs many of the steps required to build and train a model automatically, reducing the need for deep machine learning skills and domain knowledge. You just select the training data and the required model type, and Automated ML determines the best algorithm to use and trains the model.

The Designer is a no-code tool that allows you to build pipelines for data preparation and model creation by dragging.

The Automated ML and Designer tools are explained in [Skill 2.4](#), later in this chapter.

Prompt Flow is another tool available in Azure Machine Learning that enables the visual design, testing, and deployment of AI applications powered by large language models (LLMs), streamlining prompt engineering and collaboration.

Compute

Before you can start ingesting data or building a model, you must first assign a compute instance. A *compute instance* is a configured development virtual machine environment for machine learning. A compute instance is used as a compute target for authoring and training models for development and testing purposes.

Clicking Compute in the navigation pane of Azure Machine Learning Studio (left side) displays the Compute options, as shown in [Figure 2-23](#).

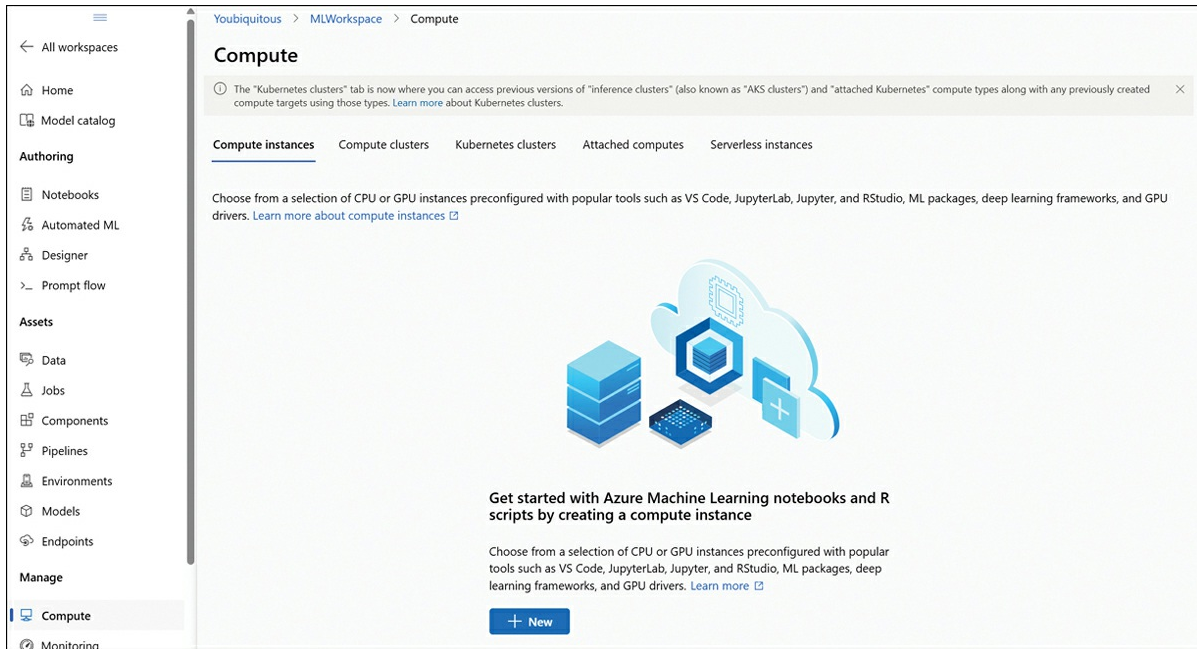


FIGURE 2-23 Machine Learning Studio compute

Clicking + New opens the Create Compute Instance pane, as shown in Figure 2-24.

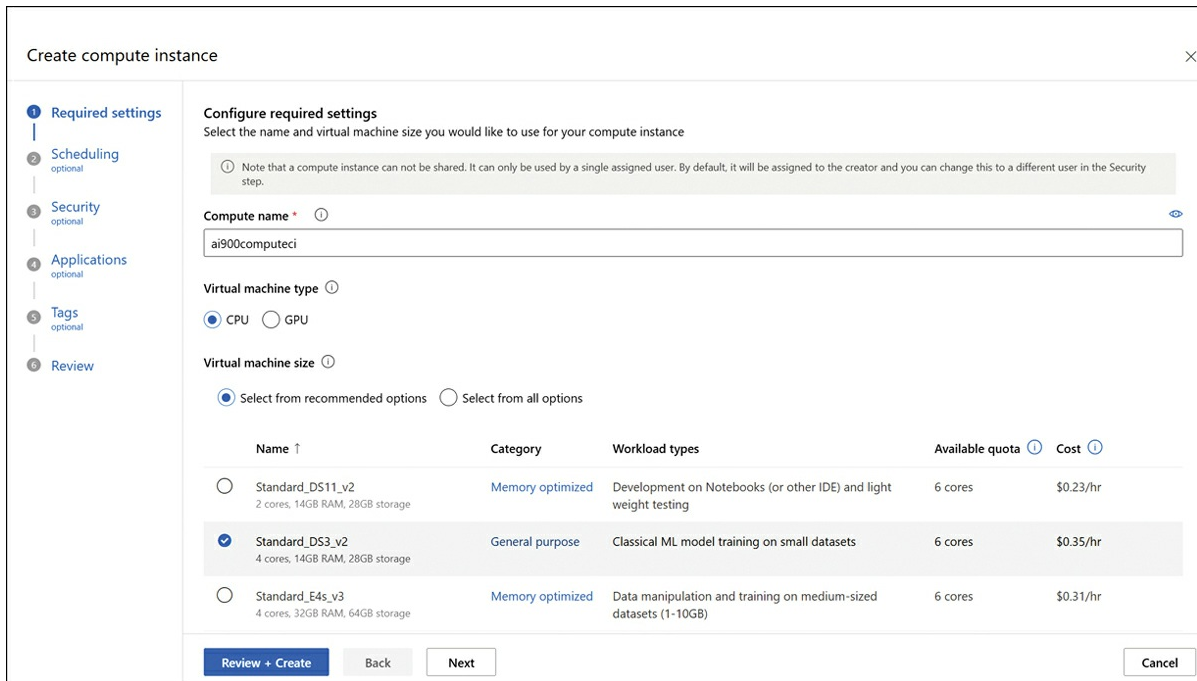


FIGURE 2-24 Create a compute instance

Provide a unique name for the instance, select the machine type and size, then click Review + Create to create the virtual machine. The virtual machine will be created and started after a few minutes. You will be able to see the state of the virtual machine, as shown in Figure 2-25.

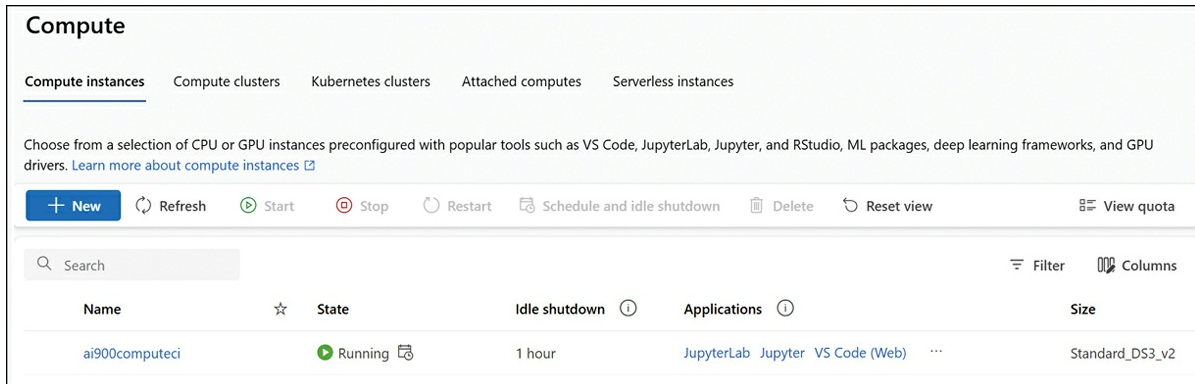


FIGURE 2-25 Compute instances

Note Stop Compute Instances When Not in Use

It is important to shut down the virtual machine used for the compute instance in order to reduce Azure spent. You can define when a compute should automatically shut down in the advanced settings. You can also define automatic start and stop schedules.

You can stop the virtual machine used for the compute instance by clicking the Stop button, as shown in [Figure 2-26](#).

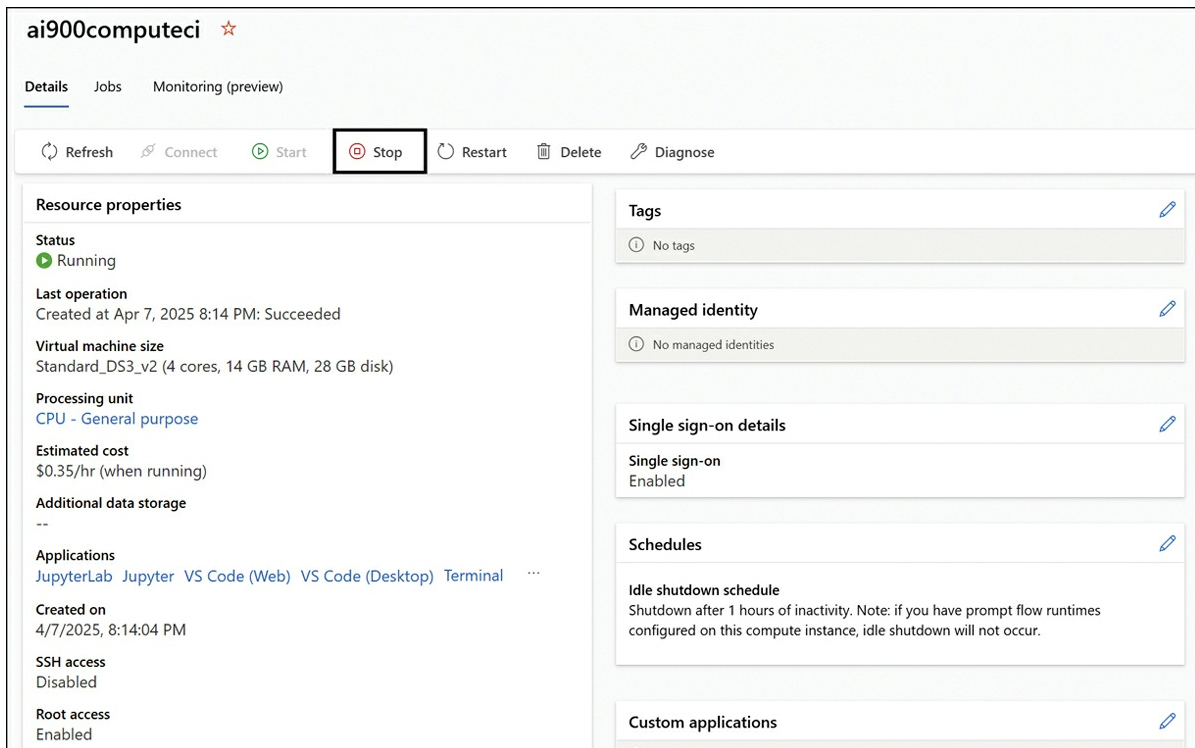


FIGURE 2-26 Compute instance

Describe data ingestion and preparation

To train a model, you need to supply data for training and testing. You are going to need to either import your own data into the Machine Learning workspace or link to existing datastores used by other applications.

Ingestion

Each Machine Learning workspace includes four preconfigured datastores, each serving a distinct purpose. The workspaceblobstore acts as the primary storage location, commonly used for uploading data, job code snapshots, and pipeline data cache. The workspaceartifactstore is designated for storing system-generated assets such as performance metrics, models, and components. The workspacefilestore offers an alternative container for data upload. Lastly, the workspaceworkingdirectory provides a storage for notebooks, compute instances, and prompt flow operations.

[Figure 2-27](#) shows the built-in datastores in the Machine Learning workspace.

Data

Data assets **Datstores** Dataset monitors PREVIEW Data import PREVIEW Data connections PREVIEW

Datstores securely connect to a storage service on Azure by storing connection information. With datstores, you no longer need to provide credential information in your scripts to access your data. [Learn more about datstores](#)

OneLake datstores used to connect to data in your Microsoft Fabric endpoint can be created via CLI and SDK only. [Learn more about OneLake datstore](#)

[+ Create](#) [Refresh](#) [Unregister](#) [Set as default datstore](#) [Reset view](#)

Search Filter Columns

Name	☆	Type	Storage name	Created on ↓
workspaceworkingdirectory		Azure file share	mlworkspaceai900preps	Apr 7, 2025 7:36 PM
workspacefilestore		Azure file share	mlworkspaceai900preps	Apr 7, 2025 7:36 PM
workspaceblobstore (Default)		Azure Blob Storage	mlworkspaceai900preps	Apr 7, 2025 7:36 PM
workspaceartifactstore		Azure Blob Storage	mlworkspaceai900preps	Apr 7, 2025 7:36 PM

FIGURE 2-27 Datstores

Click the + Create button to add in the following existing datstores:

- Azure Blob Storage
- Azure file share
- Azure Data Lake Storage Gen 1
- Azure Data Lake Storage Gen 2
- Azure SQL database
- Azure PostgreSQL database
- Azure MySQL database

If you have existing data in Azure SQL database, you need to supply the details of the Azure SQL database, as shown in [Figure 2-28](#).

FIGURE 2-28 Add Azure SQL database

To import your data into the Azure Machine Learning workspace, you need to register data assets using Machine Learning Studio. You can create a data asset from:

- Files uploaded from your local computer
- A datastore associated with the workspace
- Files accessed via an HTTP URL
- Open datasets

When you import data, you must define the dataset type as either tabular or file:

- **Tabular** Files containing data in a tabular format. You can create a tabular dataset from CSV, TSV, Parquet, JSON files, and from the output of an SQL query.
- **File** A collection of file references in datastores or public URLs. A file dataset references single or multiple files.

Note Creating Data Assets

You can create a data asset from the output of a SQL query but not from a SQL query itself. You can create a data asset from files in an Azure Blob container but not from an Azure Blob container itself. You can create a data asset from files in a folder but not from a folder itself.

The student data you used earlier in this chapter is held in a CSV file on a local computer. To create a data asset for it, first click + Create to open the Create Data Asset Wizard, shown in [Figure 2-29](#).

Create data asset

- 1 Data type
- 2 Data source

Set the name and type for your data asset

Name * 👁

Description

Data asset description

Type * ⓘ

Tabular
▼

Back
Next

FIGURE 2-29 Using the Create Data Asset Wizard, step 1

You must first provide a name for the dataset and specify its type, either Tabular or File. Upon proceeding, you are prompted to select the data source, choosing from a range of options including Azure Storage, local files, SQL databases, web-based files, or Azure Open Datasets (see Figure 2-30).

Create data asset ✕

- ✔ Data type
- 2 Data source
- 3 Destination storage type
- 4 File or folder selection
- 5 Settings
- 6 Schema
- 7 Review

Choose a source for your data asset

Choose the data source you want to create your asset from. A data source can be from a local storage location on your computer, from an attached datastore, from Azure storage, or from a publicly available web location.

From Azure storage

Create a data asset from registered data storage services including Azure Blob Storage, Azure file share, and Azure Data Lake.

From local files

Create a data asset by uploading files from your local drive.

From SQL databases

Create a dataset from Azure SQL database and Azure PostgreSQL database.

From web files

Create a data asset from a single file located at a public web URL.

From Azure Open Datasets

Create a dataset with one-click from pre-made data sets. These data sets are created by the general public and published as Azure Open Datasets

Back
Next
Cancel

FIGURE 2-30 Creating a data asset, step 2

Select From Local Files, and click Next to reach the next step in the wizard, shown in Figure 2-31.

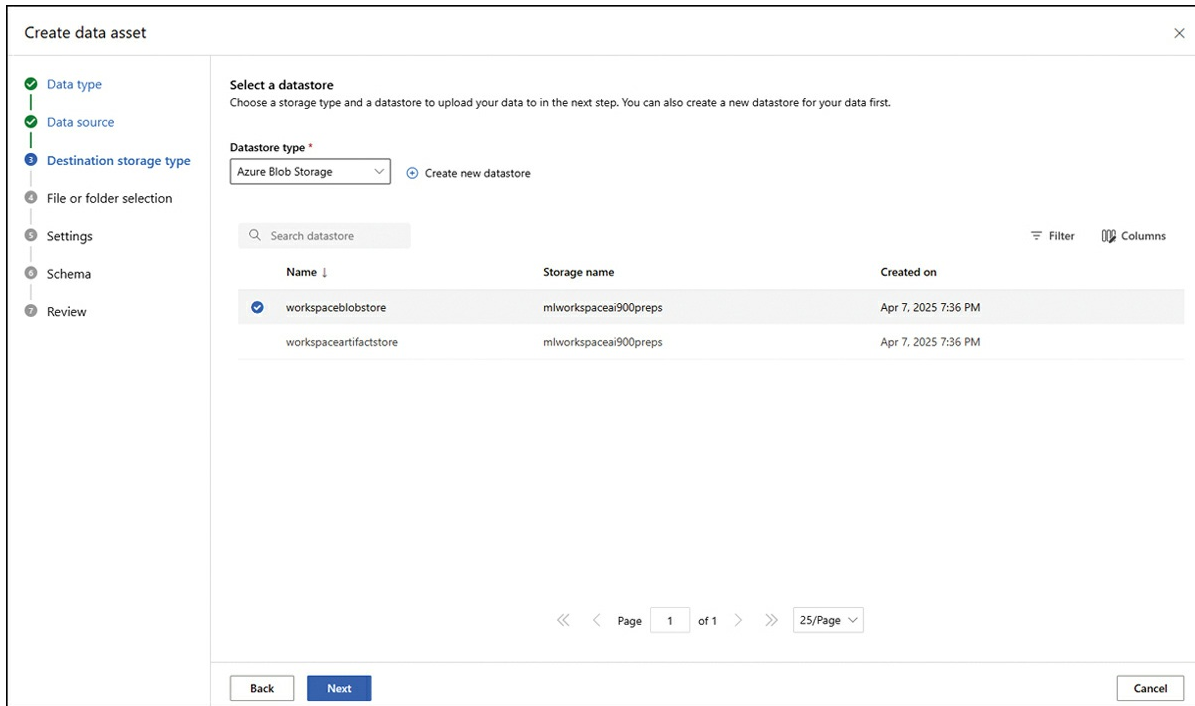


FIGURE 2-31 Creating a data asset, step 3

Here, select the datastore where the data will be imported into. This will normally be the built-in blob datastore, but you can add your own datastore if required. Click Next, then upload a single file or select an entire folder of files to upload, as shown in Figure 2-32.

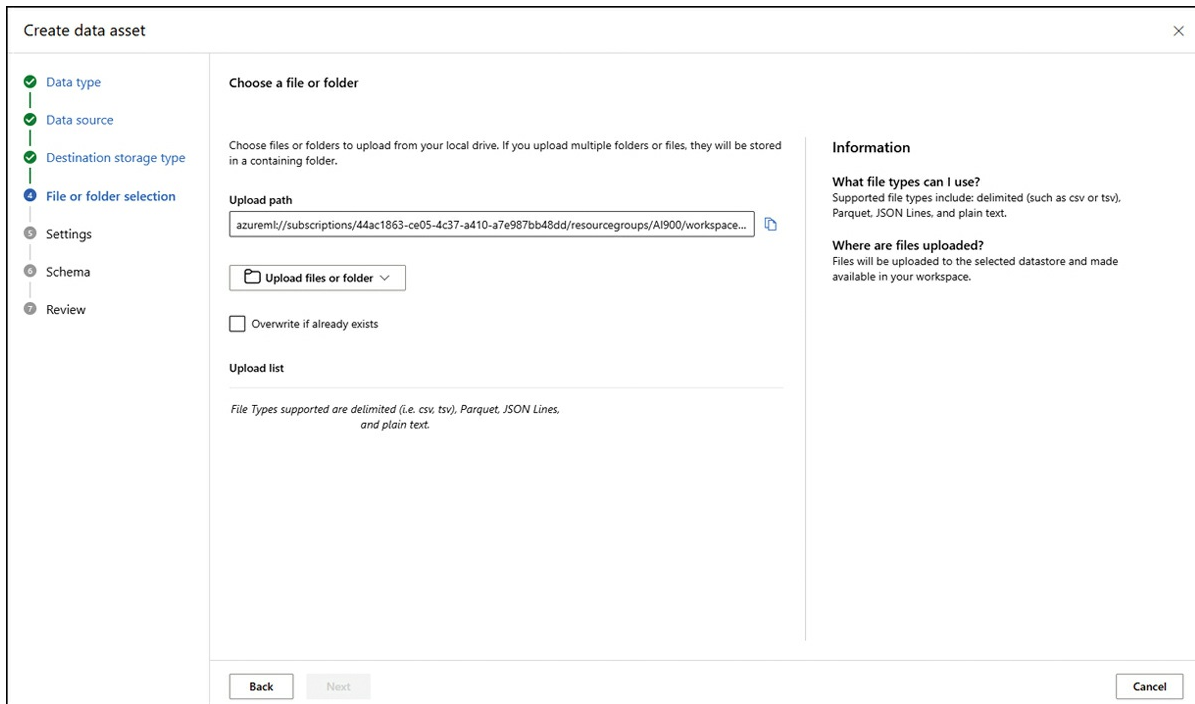


FIGURE 2-32 Creating a data asset, step 4

Click Next to parse the selected file(s) and display the next step in the wizard, shown in Figure 2-33. Here you can review the tabular data to ensure it has been parsed correctly, then click Next to continue.

Create data asset

- Data type
- Data source
- Destination storage type
- File or folder selection
- Settings**
- Schema
- Review

Settings
These settings determine how the data is parsed. The initial settings are automatically detected; you can change them as needed to reparse the data.

File format: Delimited | Delimiter: Semicolon | Example: Field1;Field2;Field3 | Encoding: UTF-8

Column headers: All files have same headers | Skip rows: None

Dataset contains multi-line data ⓘ
 ⓘ Note: Processing tabular files with multi-line data is slower because multiple CPU cores cannot be used to ingest the data in parallel. Checking this option may result in slower processing times.

Data preview

Student	Background	Hours Studied	Completed Labs	Score	Pass
Student A	Computer Science	5	false	500	false
Student B	Computer Science	6	true	600	false
Student C	Mathematics	8	true	650	false
Student D	History	10	false	623	false
Student E	Computer Science	12	true	733	true
Student F	Mathematics	15	false	697	false
Student G	Mathematics	18	false	727	true
Student H	Computer Science	20	true	850	true
Student I	History	20	true	715	true
Student J	English	25	false	780	true

Back Next Review Cancel

FIGURE 2-33 Creating a data asset, step 5

In this next step, you can exclude columns from being imported and correct any data type for each column (Figure 2-34).

Create data asset

- Data type
- Data source
- Destination storage type
- File or folder selection
- Settings
- Schema**
- Review

Schema
Column types are auto-detected based on the initial subset of the data and can be updated here. Values not aligning with the specified column type will fail conversion and would be either null-filled or replaced with error value. Any conversions preview errors are non-blocking and you can proceed.

Search column name

Include	Column name	Type	Example values	Date format ⓘ	Properties ⓘ
<input type="checkbox"/>	Path	String		Not applicable to selected type	Not applicable to ...
<input checked="" type="checkbox"/>	Student	String	Student A, Student B, Student C	Not applicable to selected type	Not applicable to ...
<input checked="" type="checkbox"/>	Background	String	Computer Science, Computer Science, ...	Not applicable to selected type	Not applicable to ...
<input checked="" type="checkbox"/>	Hours Studied	Integer	5, 6, 8	Not applicable to selected type	Not applicable to ...
<input checked="" type="checkbox"/>	Completed Labs	Boolean	false, true, true	Not applicable to selected type	Not applicable to ...
<input checked="" type="checkbox"/>	Score	Integer	500, 600, 650	Not applicable to selected type	Not applicable to ...
<input checked="" type="checkbox"/>	Pass	Boolean	false, false, false	Not applicable to selected type	Not applicable to ...

Back Next Cancel

FIGURE 2-34 Creating a data asset, step 6

Click Next to go to the final step in the wizard (Figure 2-35).

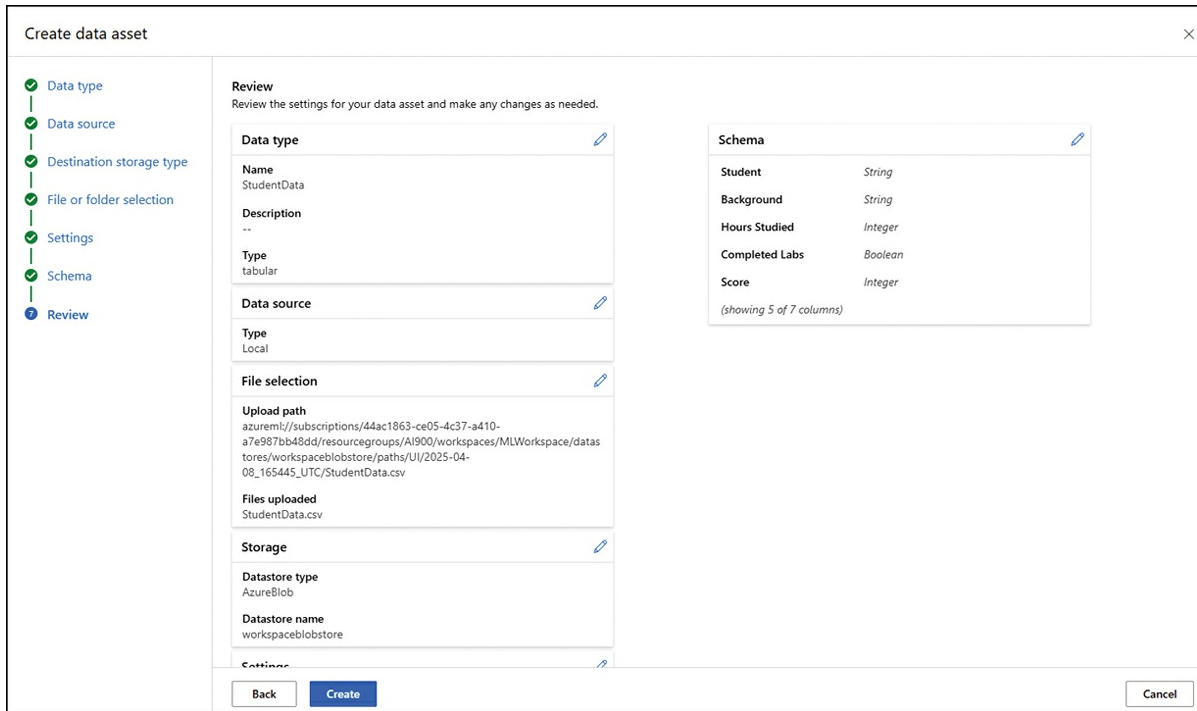


FIGURE 2-35 Creating a data asset, step 7

Now click Create to import the data and register the data asset, as shown in Figure 2-36.

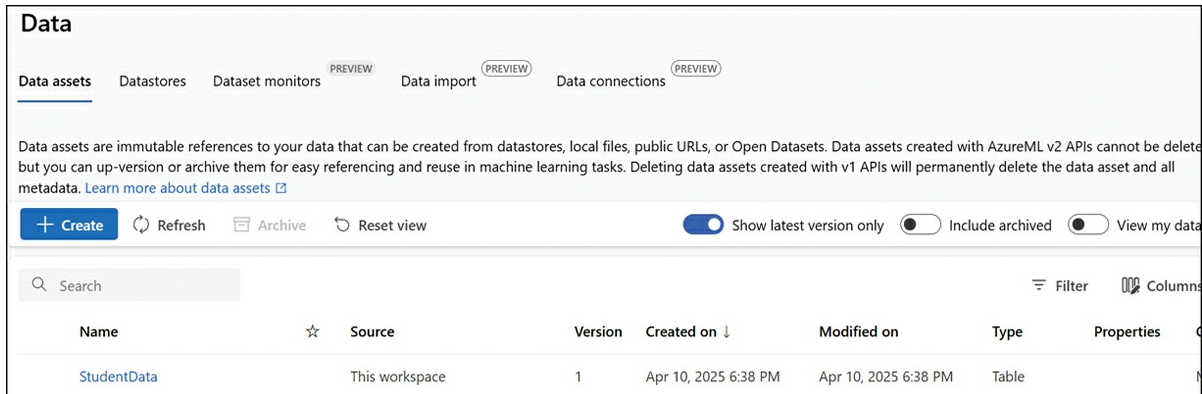


FIGURE 2-36 Registered data assets

By accessing the Details of the newly created data asset, you can also create a new version of it by clicking New Version. This allows for effective versioning and makes it possible to revert to a previous version if needed.

You can use publicly available data from URLs such as the daily bike rental data at aka.ms/bike-rentals. As in the previous steps, you begin by providing a name and selecting the dataset type (step 1), followed by choosing the data source, in this case, From Web Files (step 2). Step 3, shown in Figure 2-37, illustrates how to proceed with importing the dataset from a public URL.

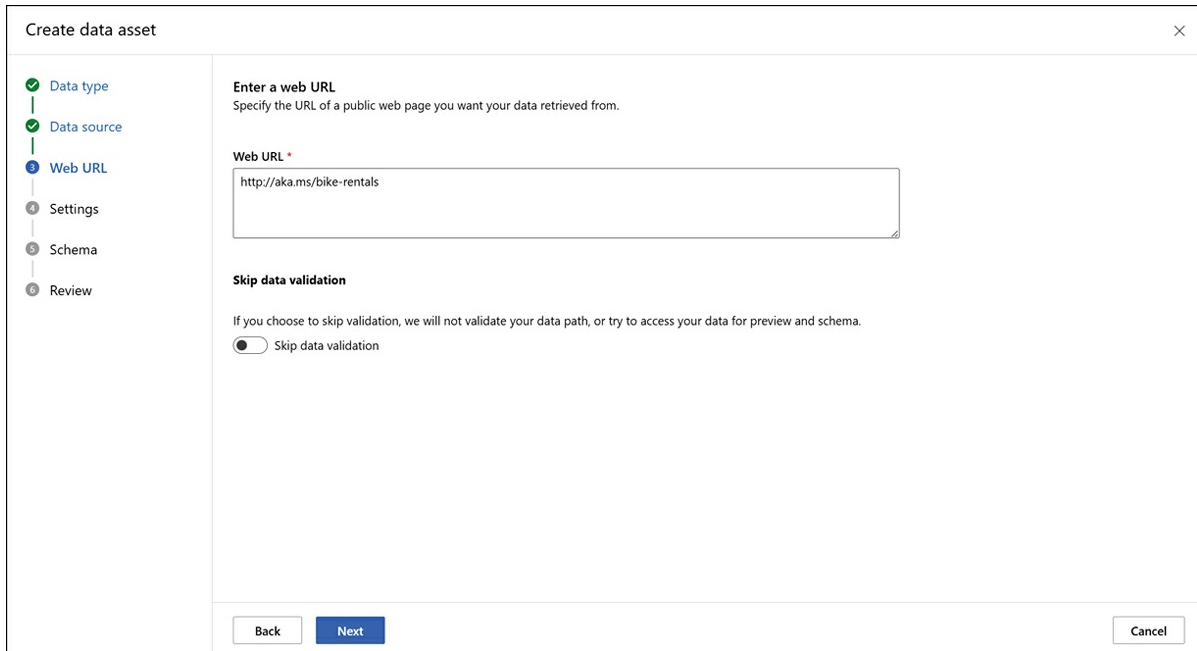


FIGURE 2-37 Creating a data asset from a web file, step 3

If, in step 2 of the dataset creation process, you select Azure Open Datasets as the data source, you will be presented with a collection of publicly available datasets hosted on Azure.

These datasets can be directly imported into your machine learning project, as shown in Figure 2-38. To find out more about these and other publicly available datasets, see docs.microsoft.com/azure/open-datasets/dataset-catalog.

Note Sample Datasets

There are a further 15 sample datasets available in the Azure Machine Learning Designer tool (docs.microsoft.com/azure/machine-learning/samples-designer#datasets).

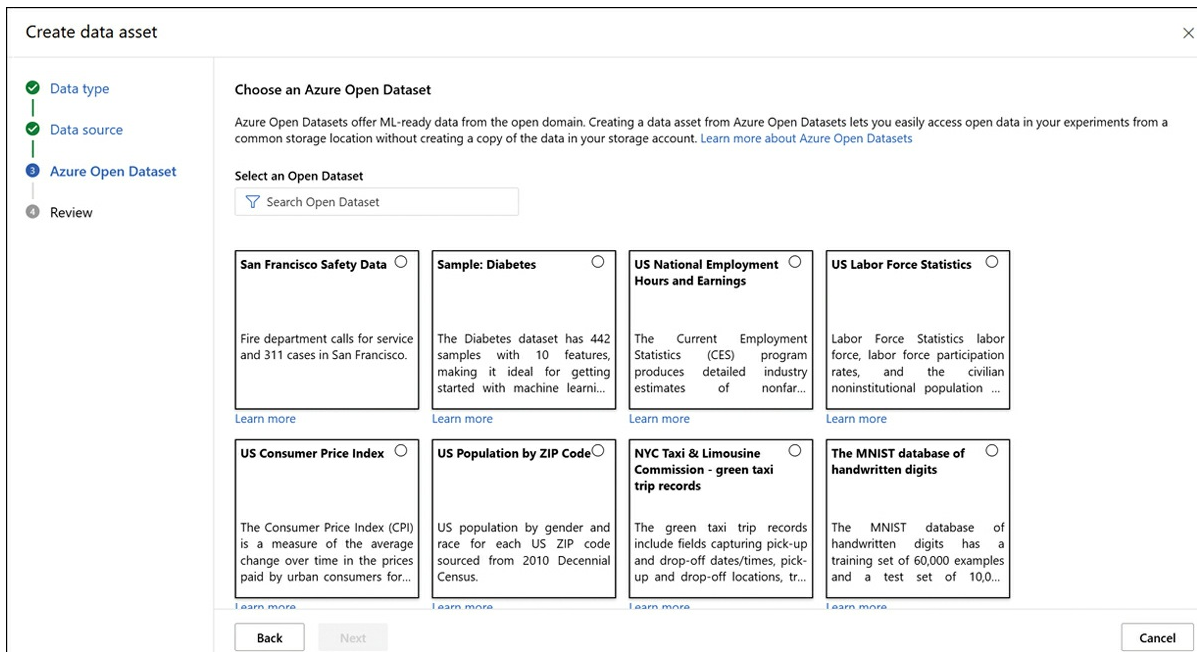


FIGURE 2-38 Open datasets

Preparation

Once your data is ingested, you need to prepare that data for training and testing models. A good place to start is to explore the profile of the data. In Machine Learning Studio, you can generate and review the profile of your data, as shown in [Figure 2-39](#).

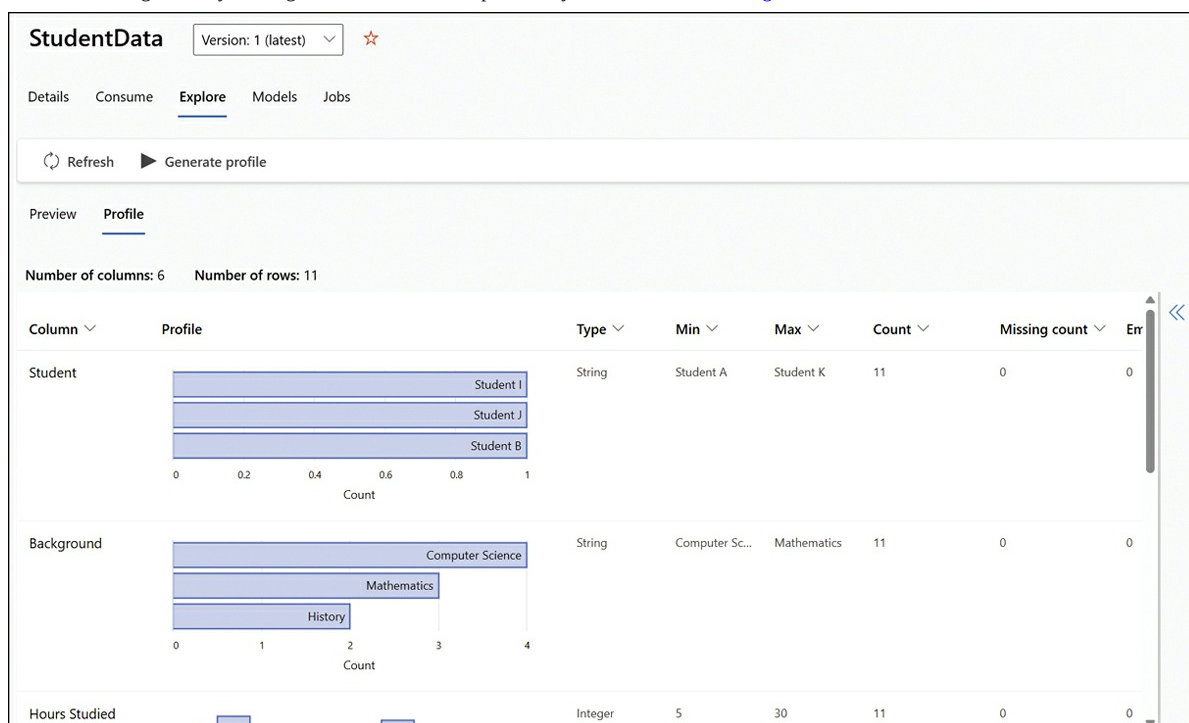


FIGURE 2-39 Data asset profile

The profile will inform you of issues with your data that require transformation and cleansing. You can perform many actions to transform your dataset:

- **Normalization** Adjusting the values in numeric columns so that all numeric values are on a common scale, normally between 0 and 1. A dataset that has features with different ranges can bias the model toward that feature.
- **Partitioning and sampling** A method for reducing the size of data while retaining the same ratio of values.

How you perform these actions will depend on the tool you choose to build and train your model.

Data that is incomplete can cause issues when training your model. You can clean missing data by methods such as the following:

- Replacing missing values with an average or another value
- Removing empty (or sparsely populated) rows

Describe feature selection and engineering

Once you've prepared the data, you need to choose the features that you will use to train the model. You can select features from the dataset, but it may be necessary to engineer new features.

Feature selection

Feature selection is the process of selecting a subset of the columns in the dataset features to exclude features that are not relevant to the machine learning problem that you are trying to resolve. Feature selection restricts the data to the most valuable inputs, reducing noise and improving training performance.

Feature selection has two main purposes:

- Increase the model's ability to classify data accurately by eliminating features that are irrelevant, redundant, or highly correlated
- Increase the efficiency of the model training process

If you can reduce the number of features without losing the variance and patterns in the data, the time taken to train the model is minimized. For instance, you can exclude features that are highly correlated with each other because such correlation just adds redundancy to the processing.

Azure Machine Learning Designer has a module for Filter-Based Feature Selection to assist in identifying features that are irrelevant. Azure Machine Learning applies statistical analysis to determine which columns are more predictive than the others and ranks the results. You can exclude the columns that have a poor predictive effect.

Feature engineering

Feature engineering is the process of creating new features from raw data to increase the predictive power of the machine learning model. Engineered features capture additional information that is not available in the original feature set. Examples of feature engineering are aggregating data, calculating a moving average, and calculating the difference over time.

It can be beneficial to aggregate data in the source data, reducing the amount of data imported and used to train the model.

In Azure Machine Learning, you can use modules such as Feature Hashing that use Azure AI Services to turn text into indices. For example, in the student data, you could apply Feature Hashing to convert the academic subject for each student into a numeric hash value.

Note Feature Hashing

The Feature Hashing module referenced in this section has been deprecated due to its reliance on the NimbusML project, which is no longer actively maintained. Consequently, this module will not receive future updates or security patches and is expected to be removed in upcoming releases of Azure Machine Learning.

Binning is an example of feature engineering in which the data is segmented into groups of the same size. Binning is used when the distribution of values in the data is skewed. Binning transforms continuous numeric features into discrete categories.

A common example of feature engineering is around dates and times. You can convert dates and times to the relative number of days, hours, and minutes, and then create the difference in minutes between two datetime columns. This might create a model that predicts more accurate outcomes.

Another similar example of feature engineering is to extract features from a single column, such as converting a date into the following data:

- Day of the week
- Day of the month
- Month
- Year
- Weekend or weekday
- Public holiday or working day

The daily bike rentals data shows how such features have been engineered.

Describe model training and evaluation

Azure Machine Learning provides several ways to train your models, from code-first solutions using the various SDKs to no-code tools. The most effective method to train your model depends on which option you choose. You can either create a configuration script—for example, in Python—and then run a training job, or you can use *pipelines*, which are workflows containing reusable modules.

Note Pipelines

Pipelines are more flexible and are not limited to training and can include other steps, such as data ingestion and preparation.

Compute cluster

Before you can train a model, you need to assign compute resources to your workspace. A compute cluster is used to train models with the Azure Machine Learning workspace. The cluster can also be used for generating predictions on large amounts of data in batch mode.

Clicking Compute in the Machine Learning Studio displays the Compute options, as shown previously in [Figure 2-23](#). Select the Compute Clusters tab to view a list of any existing clusters. To create a new one, click the + New button to access the Create Compute Cluster settings shown in [Figure 2-40](#).

Create compute cluster

Virtual Machine

Advanced Settings

Select virtual machine
Select the virtual machine size you would like to use for your compute cluster.

Location *
UK South

Virtual machine tier
 Dedicated Low priority

Virtual machine type
 CPU GPU

Virtual machine size
 Select from recommended options Select from all options

Name ↑	Category	Workload types	Available quota	Cost
<input type="radio"/> Standard_DS11_v2 2 cores, 14GB RAM, 28GB storage	Memory optimized	Development on Notebooks (or other IDE) and light weight testing	2 cores	\$0.23/hr
<input checked="" type="radio"/> Standard_D4s_v3 4 cores, 16GB RAM, 32GB storage	General purpose	Classical ML model training on small datasets	6 cores	\$0.23/hr
<input type="radio"/> Standard_E4s_v3	Memory optimized	Data manipulation and training on medium-sized	6 cores	\$0.31/hr

Back Next Cancel

FIGURE 2-40 Creating a compute cluster

You will need to select the machine type and size. Do so, then click Next to open the Configure Settings shown in Figure 2-41. Here, you need to provide a unique name for the cluster and specify the minimum and maximum number of nodes in the cluster. You then click Create to create the cluster.

Create compute cluster

Virtual Machine

Advanced Settings

Configure Settings
Configure compute cluster settings for your selected virtual machine size.

Name	Category	Cores	Available quota	RAM	Storage	Cost/Node
Standard_D4s_v3	General purpose	4	6 cores	16 GB	32 GB	\$0.23/hr

Compute name *
ai900computecl

Minimum number of nodes *
0

Maximum number of nodes *
1

Idle seconds before scale down *
120

Enable SSH access

> Advanced settings

Add tags
Name : Value Add

Back Create Download a template for automation. Cancel

FIGURE 2-41 Create Compute Cluster settings

The cluster will be created after a few minutes. You will be able to see the state of the cluster in the Compute Clusters tab, as shown in Figure 2-42.

Compute

Compute instances **Compute clusters** Kubernetes clusters Attached computes Serverless instances

Create a single or multi node compute cluster for your training, batch inferencing or reinforcement learning workloads. [Learn more about compute clusters](#) 🔗
 Alternatively, you can now run a training job without having to create and manage compute by using serverless. [Learn more here](#) 🔗

+ New ↻ Refresh 🗑 Delete ↺ Reset view ☰ View quota

🔍 Search ☰ Filter 📄 Columns

Name	☆	State	Size	Location	Created on ↓	Active runs	Idle nodes
ai900computecl		✔ Succeeded (0 nodes)	Standard_D4s_v3	uksouth	Apr 13, 2025 12:09 PM	0	0

FIGURE 2-42 Compute Clusters tab

The number of active nodes will scale up and down automatically based on the training you are performing in the workspace.

Note Minimize Compute Cluster Nodes When Not in Use

The nodes in the compute cluster will stop automatically based on the idle time, which defaults to two minutes. When not training machine learning models, you should reduce the number of nodes to reduce Azure spend. It is easy to increase the number of nodes again when you next need them.

You can change the number of nodes used in the compute cluster by navigating to the cluster’s Details page and clicking the pencil icon to edit the Resource properties, where you can modify the minimum and maximum node values, as shown in [Figure 2-43](#).

Minimum number of nodes * ⓘ

Maximum number of nodes * ⓘ

Idle seconds before scale down * ⓘ

FIGURE 2-43 Compute cluster nodes

If you are not training models, you can set both numbers to zero.

Training

Training a model requires a combination of dataset, algorithm, and compute. Azure Machine Learning allows you to easily train a model, but in the real world, you need to run multiple experiments with different features and algorithms to build the best performing model.

Before training, you should have transformed and cleansed your dataset, selected features, performed any feature engineering required, and normalized the data. By properly preparing your dataset, you can improve the accuracy of the trained model.

Azure Machine Learning logs metrics for every experiment, and you can use these metrics to evaluate the model’s performance.

Once you have deployed your model, training does not stop. You may get additional data, and you will need to train and evaluate the model. Remember, building models is an iterative and ongoing process. With Azure Machine Learning, you can create new versions of your model and then deploy a version to replace the existing deployed model. You can also revert to an older version of the model if the version you have deployed does not perform well with new data.

Many algorithms have parameters, known as *hyperparameters*, that control how model training is done, which can have a significant impact on model accuracy. Azure Machine Learning has a module where you can set and tune hyperparameters by iterating multiple times with combinations of parameters to find the best fit model.

With the K-means clustering algorithm, you can adjust the K, which is the target number of clusters you want the model to find. Increasing K increases the compute time and cost.

Scoring

Azure Machine Learning contains a module that can split data into training and testing datasets. Once the model has been trained, you use the testing dataset to score the model.

The testing dataset contains data that was not used to train the model—data that is new to the model. You use the trained model to generate predictions for each row in the testing dataset. Azure Machine Learning enables visualization of the results from scoring; for instance, [Figure 2-44](#) shows the results of a regression model using the student dataset, specifically using the hours studied as a single feature and the score as the label.

Scored_dataset		
Rows	Columns	
4	3	
Hours Studied	Score	Scored Labels
0.4	697	691.870228
0.6	727	713.161578
0	500	649.287528
0.2	623	670.578878

FIGURE 2-44 Score results for regression

In the results, you can see that the prediction (the scored label) for the first two rows is close to the actual score, but the other two rows are showing higher residual errors. The third row has a predicted score of 649, but the actual score was 500; the fourth row has a predicted score of 670, but the actual score was 623.

Figure 2-45 shows the scores for a classification model using the student dataset's Hours Studied and Completed Labs as features and Pass as the label.

Scored_dataset				
Rows	Columns			
4	5			
Hours Studied	Completed labs	Pass	Scored Labels	Scored Probabilities
0.4	false	false	false	0.5
0.6	false	true	true	1
0	false	false	false	0.375
0.2	false	false	true	0.625

FIGURE 2-45 Score results for classification

In the results, you can see that the prediction (the Scored Label) for the first three rows is correct. The first row is a true negative, with the actual and prediction both false with a probability of 50%. The second row is a true positive, with the actual and prediction both true with a probability of 100%. The third row is a true negative, with the actual and prediction both false, but with a probability of only 37.5%. The fourth row is a false positive, with the actual being a fail but the model predicting a pass.

Once you have scored your model, you use the scores to generate metrics to evaluate the model's performance and how good the predictions of the model are.

The sequence of the tasks when building a model is shown in Figure 2-46.



FIGURE 2-46 Sequence of model-building tasks

Evaluation

Evaluation is the process of measuring the accuracy of a trained model. A set of metrics is used to measure how accurate the predictions of the model are. Evaluation is a key part of the building of your model.

Azure Machine Learning generates metrics for each experiment. You can view these metrics to evaluate your model's performance (as shown previously in Figure 2-44) for a regression model using the student dataset, with the Hours Studied column as a single feature and the Score column as the label, as shown in Figure 2-47.

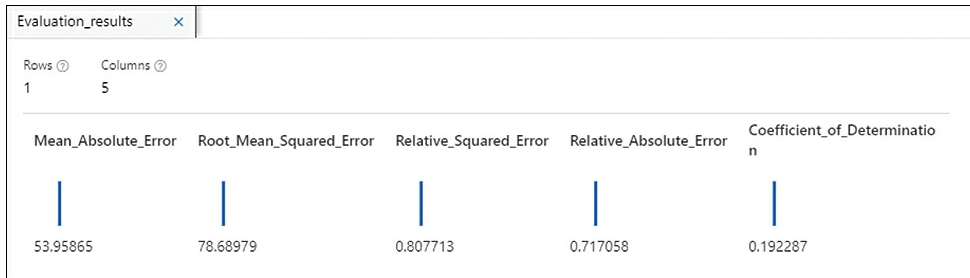


FIGURE 2-47 Metrics for a regression model

As you can see from these metrics, this model does not perform well with high error values and a low coefficient of determination. You will need to select additional features and train the model to see if you can create a better performing model.

For instance, Figure 2-48 shows the metrics for a classification model using the student dataset’s Hours Studied and Completed Labs as features and Pass as the label.

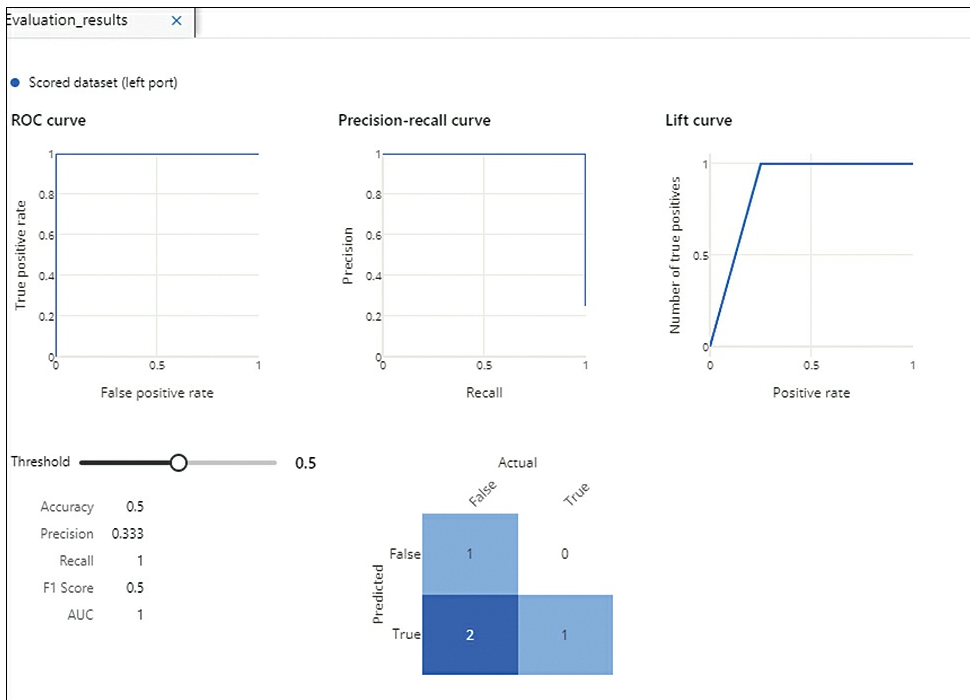


FIGURE 2-48 Metrics for a classification model

As you can see from these metrics, with a threshold set to 50%, the model is only 50% accurate with a precision of 33%. Again, you will need to add additional data, use a different algorithm, or perform feature engineering, and train the model to see if you can create a better performing model.

Be on watch for bias in the dataset too, where the algorithm is unable to separate the true signal from the noise. This is often caused by the dataset used for training. If you use a split of the same dataset for scoring and evaluating, then the model may appear to perform well but does not generalize—in other words, it does not predict well with new unseen data.

Known as overfitting, this is one of the most common problems with machine learning. *Overfitting* means that the model does not generalize well from training data to unseen data, especially data that is unlike the training data. Common causes are bias in the training data or too many features selected, meaning the model cannot distinguish between the signal and the noise.

One way to avoid overfitting is to perform cross-validation. In *cross-validation*, a dataset is repeatedly split into a training dataset and a validation dataset. Each split is used to train and test the model. Cross-validation evaluates both the dataset and the model, and it provides an idea of how representative the dataset is and how sensitive the model is to variations in input data.

Describe model deployment and management

Azure Machine Learning allows you to create and manage versions of your model and then choose which version of a model to deploy so that applications can use the model to make predictions.

In machine learning, *inferencing* refers to the use of a trained model to make predictions for new unseen data. A machine learning model can be used for two types of inferencing:

- **Real-time** For individual or small numbers of data observations
- **Batch** For large volumes of data

For batch inference processing, you create a pipeline that includes steps to load the input data, load the model, predict labels, and write the results to a datastore, normally to a file or to a database.

For real-time processing, the model is deployed as an *online endpoint*, which enables applications to request via REST API (<http>).

If you have used a pipeline to build your model, you can create an inference pipeline that performs the same steps for new data input, not the sample data used in training. You can publish the inference pipeline as a web service. A real-time inference pipeline must have at least one Web Service Input module and one Web Service Output module. The Web Service Input module is normally the first step in the pipeline. The pipeline performs the same steps for new data input. The Web Service Output module is normally the final step in the pipeline.

In Azure Machine Learning, when you publish a real-time inferencing model, you deploy the model as a web service running in a Docker container. There are different ways you can deploy the container:

- **Managed online endpoints** Online endpoints offer a fully managed compute with autoscaling that decrease your effort in infrastructure management. This option is recommended for real-time inference.
- **Azure Kubernetes Service online endpoints (AKS)** AKS is the Microsoft implementation of Kubernetes and is used to run highly scalable real-time inference as a web service for production. AKS gives you the full control over scaling, networking, and security.
- **Azure Container Instance (ACI)** ACI is used to run a single Docker container. You can use this to expose your prediction model as a web service for testing and low-volume production scenarios. This option is recommended only for testing or legacy scenarios.
- **IoT Edge** Azure IoT Edge supports the running of containers containing machine learning models on small devices, reducing the need for network traffic and reducing latency.

You can also use ONNX to export your model and deploy it on other platforms, such as on a mobile device.

Skill 2.4: Describe capabilities of no-code machine learning with Azure Machine Learning

Azure Machine Learning has two tools for building machine learning models with low or no code. This section explains how you use these Azure Machine Learning tools to build and deploy machine learning models.

This skill covers how to:

- [Describe Azure Automated Machine Learning](#)
- [Describe Azure Machine Learning Designer](#)

Describe Azure Automated Machine Learning

Azure Automated Machine Learning (Automated ML) finds the best algorithm for the dataset. A number of iterations are run to evaluate different algorithms and features. The results are analyzed and ranked, and an explanation is produced that interprets the best model for the data, parameters, and task.

Figure 2-49 shows the Automated ML process, where data is fed into Automated ML, which then outputs a best-fit trained model.

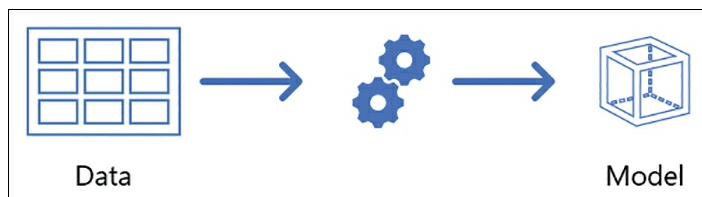


FIGURE 2-49 Automated Machine Learning

Automated ML trades data science skills with compute power and time. A data scientist will evaluate and choose the features, algorithm, and hyperparameters for the model. Automated ML simply tries all combinations of algorithms, features, and parameters and then ranks the results by the chosen metric.

Clicking Automated ML (left side of the Azure Machine Learning Studio) lists all previous ML runs and enables you to create a new run. Clicking + New Automated ML job opens the Submit an Automated ML Job Wizard (Figure 2-50).

Submit an Automated ML job

- ✔ Training method
- 2 Basic settings
- 3 Task type & data
- 4 Task settings
- 5 Compute
- 6 Review

Basic settings

Let's start with some basic information about your training job.

Job name * ⓘ 👁

Experiment name *

Select existing Create new

New experiment name * 👁

Description

Tags

: Add

Back
Next
Cancel

FIGURE 2-50 Submit an Automated ML Job Wizard, step 1

The first step in the wizard is to name the job and the experiment to which it belongs. This helps organize multiple runs under a common experiment and makes tracking results easier. You can either select an existing experiment or create a new one at this stage.

The second step is to select the type of task—classification, regression, time-series forecasting, natural language processing, or computer vision—and to choose the dataset to be used for training. You can select a dataset already registered in the workspace or register a new dataset at this point (Figure 2-51).

Submit an Automated ML job

- ✔ Training method
- ✔ Basic settings
- 3 Task type & data
- 4 Task settings
- 5 Compute
- 6 Review

Task type & data

Choose the type of task that you would like your model to perform and the data to use for training. [Learn more](#)

Select task type * ⓘ *

Regression
▼

Select data

Make sure your data is preprocessed into a supported format.

+ Create
↻ Refresh
 Show supported data assets only
↺ Reset view

Filter
Columns

Name	Type	Created on ↓	Modified on
<input checked="" type="checkbox"/> StudentData	Table	Apr 10, 2025 6:38 PM	Apr 10, 2025 6:38 PM

⏪ ⏩
Page
1
of 1
>
25/Page

Back
Next
Cancel

FIGURE 2-51 Submit an Automated ML Job, step 2

Note Ensure Sufficient Training Data Size

To successfully submit an Automated ML job, the training dataset must contain at least 50 rows. If the dataset is too small, the job will fail to run. Always verify that the dataset meets this minimum size requirement before starting the job.

In the third step, shown in [Figure 2-52](#), you define the target column (the label the model will learn to predict) and configure how the data will be split for training and validation. Options include K-fold cross-validation and train-validation split. You can also specify test data if available.

Submit an Automated ML job

✓ Training method
✓ Basic settings
✓ Task type & data
4 Task settings
5 Compute
6 Review

Task settings

Task type
Regression

Data
StudentData (View data)

Target column *
Score (Integer)

⚙ View additional configuration settings ⚙ View featurization settings

> Limits

Validate and test

You can choose a validation type and select test data as an optional step.

Validation type ⓘ
Automatic

Test data ⓘ
None

Back Next Cancel

FIGURE 2-52 Submit an Automated ML Job, step 3

 **Exam Tip**

Automated ML supports only supervised learning, as you must specify a label for the model to fit the data to. Expand the Limits section to reveal advanced options such as the maximum number of trials, maximum concurrent trials, and maximum number of nodes, which allow you to control the scale and duration of the training run. When one of these thresholds is met, the training job will end ([Figure 2-53](#)).

Submit an Automated ML job

[View additional configuration settings](#) [View featurization settings](#)

✓ Training method
✓ Basic settings
✓ Task type & data
④ Task settings
⑤ Compute
⑥ Review

▼ Limits

Max trials ⓘ
Enter total trials

Max concurrent trials ⓘ
Enter concurrent trials

Max nodes ⓘ
Enter max nodes

Metric score threshold ⓘ
Enter metric score threshold

Experiment timeout (minutes) ⓘ
Enter timeout in minutes

Iteration timeout (minutes) ⓘ
Enter timeout in minutes

Enable early termination ⓘ

Validate and test

FIGURE 2-53 Submit an Automated ML Job, step 3, Limits section

Clicking View Additional Configuration Settings in this step (Figure 2-52) opens the Additional Configuration panel (Figure 2-54). Here, you can select the metric used to evaluate and rank the models that are generated by Automated ML. By default, Automated ML will evaluate all algorithms available, and you can exclude specific algorithms if needed.

Additional configuration ✕

Primary metric ⓘ

NormalizedRootMeanSquaredError ▼ *

Explain best model ⓘ

Enable ensemble stacking ⓘ

Use all supported models

Blocked models ⓘ

A list of models that Automated ML will not use during training. ▼

Save Close

FIGURE 2-54 Submit an Automated ML Job, step 3, additional configuration settings

The final step is to select the compute type and compute instance on which the job will run. You can choose from existing compute targets or create a new one during this step. After you select the compute, you can submit the job ([Figure 2-55](#)).

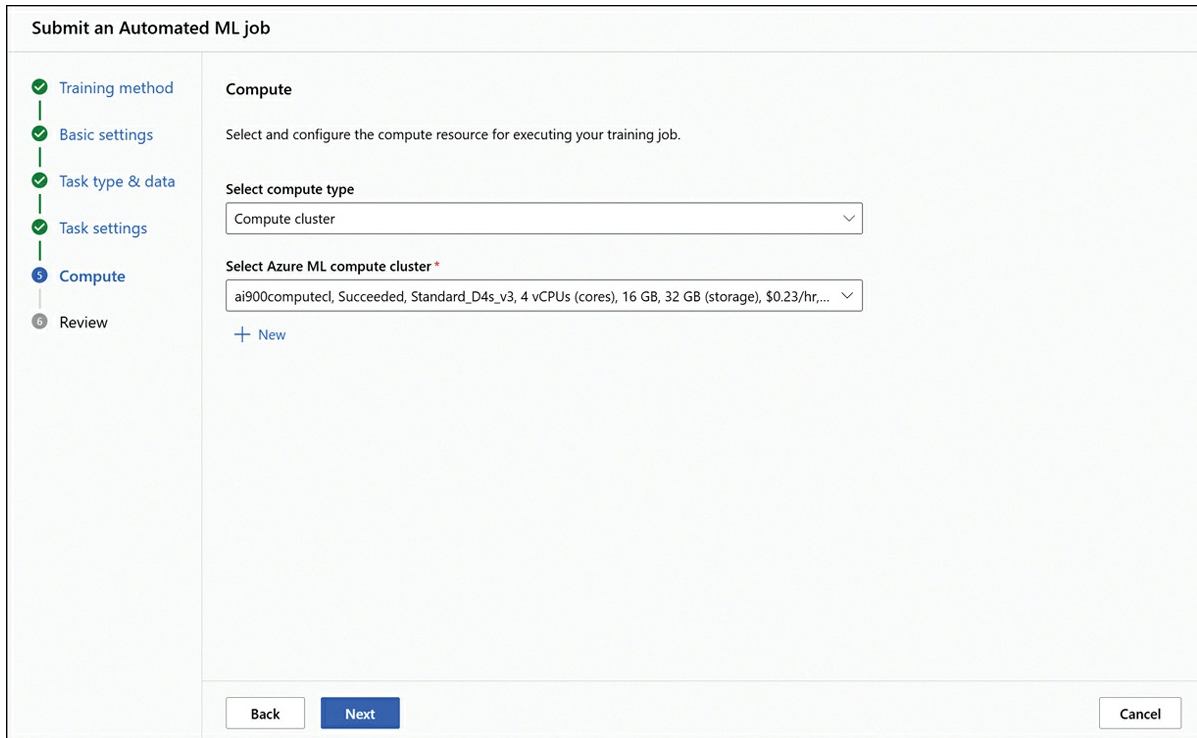


FIGURE 2-55 Submit an Automated ML Job, task type

The summary page for the Automated ML run (Figure 2-56) shows the results of the run, including the run's status and the best model. Selecting the Models + Child jobs tab enables you to review each of the models and evaluate the metrics, as shown in Figure 2-57.

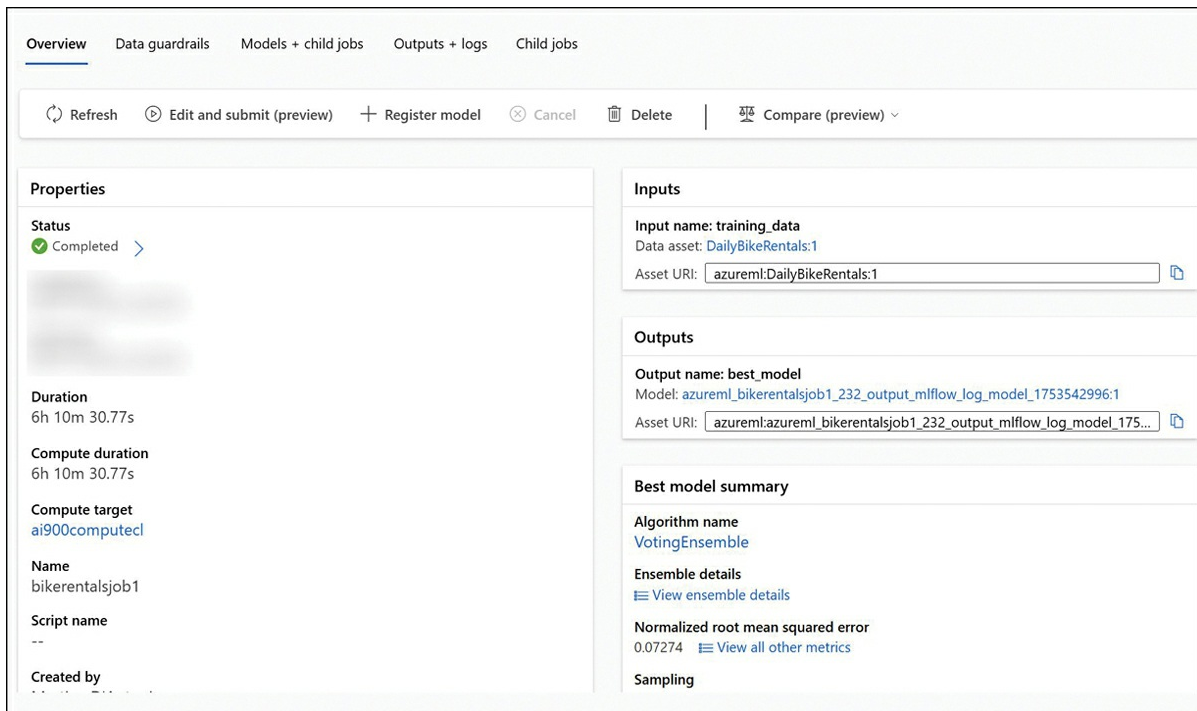


FIGURE 2-56 Automated ML run details

Algorithm name	Responsible AI	Normalized ro...	Sampling	Created on	Duration
VotingEnsemble	View responsible AI ...	0.07274	100.00 %	Apr 13, 2025 8:12 PM	1m 50s
SparseNormalizer, XGBoostRegressor		0.07666	100.00 %	Apr 13, 2025 4:33 PM	1m 1s
SparseNormalizer, XGBoostRegressor		0.07955	100.00 %	Apr 13, 2025 5:45 PM	4m 23s
MaxAbsScaler, XGBoostRegressor		0.07964	100.00 %	Apr 13, 2025 2:12 PM	28s
SparseNormalizer, XGBoostRegressor		0.08026	100.00 %	Apr 13, 2025 5:07 PM	59s
MaxAbsScaler, LightGBM		0.08049	100.00 %	Apr 13, 2025 2:12 PM	32s
SparseNormalizer, XGBoostRegressor		0.08132	100.00 %	Apr 13, 2025 8:10 PM	59s
SparseNormalizer, XGBoostRegressor		0.08237	100.00 %	Apr 13, 2025 5:27 PM	55s
StandardScalerWrapper, LightGBM		0.08279	100.00 %	Apr 13, 2025 2:33 PM	54s

FIGURE 2-57 Automated ML models

Chapter 1 discussed the principle of Transparency and how you sometimes need to be able to explain how an AI algorithm has generated its results. Automated ML generates an explanation for the model that was identified as the best model. The explanation allows you to understand why the model was selected and how the model works. It enables you to meet regulatory requirements and provides transparency to users.

Click View Explanation to see several different insights into the best model—for example, the metrics shown in Figure 2-58.

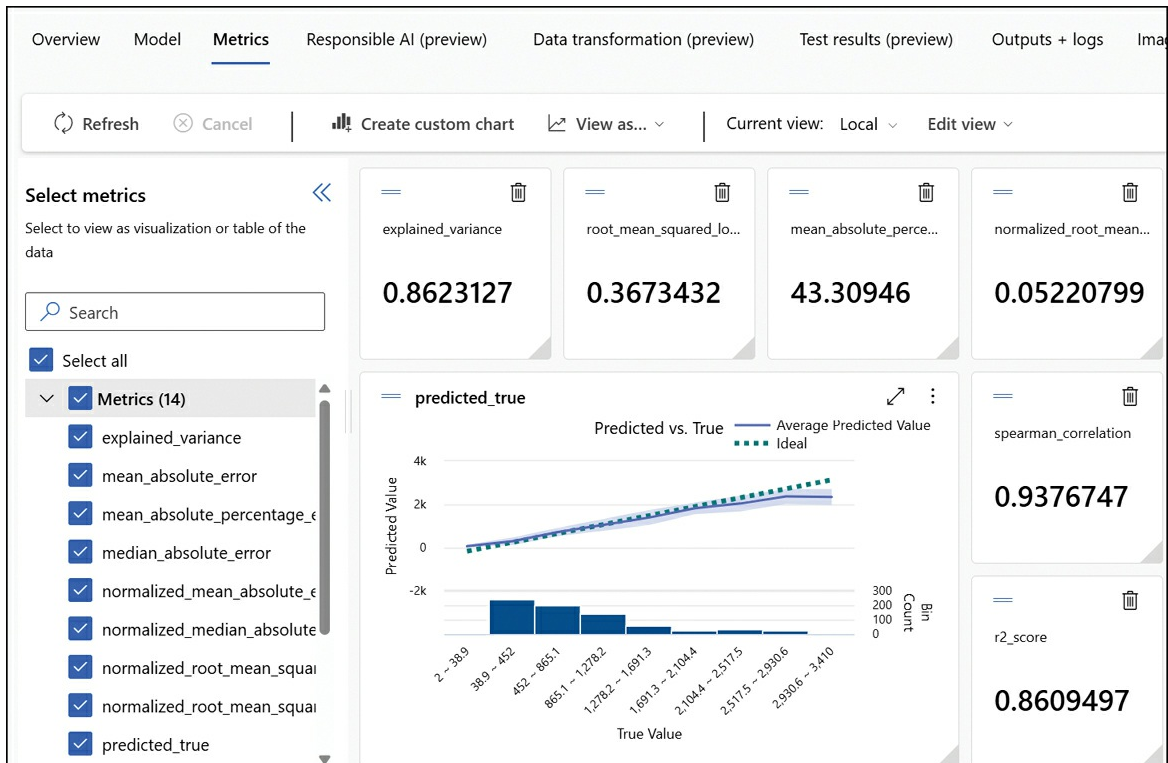


FIGURE 2-58 Explain the best model

You can use the information shown in the explanations to interpret the model and build responsible models.

You can select the model and deploy it from within Automated ML to either Azure Container Instances or to Azure Kubernetes Service. This will register the model and deploy it as a web service.

Describe Azure Machine Learning Designer

Azure Machine Learning Designer is a visual tool for creating machine learning pipelines. Azure Machine Learning Designer simplifies the process of building, testing, and deploying models, offering modules for data preparation, training, data visualization, and model evaluation.

Designer overview

Select Designer in the Azure Machine Learning Studio navigation pane to view a list of existing pipelines or create a new one. You can add either use a prebuilt pipeline or provide your own code as a custom component (Figure 2-59).

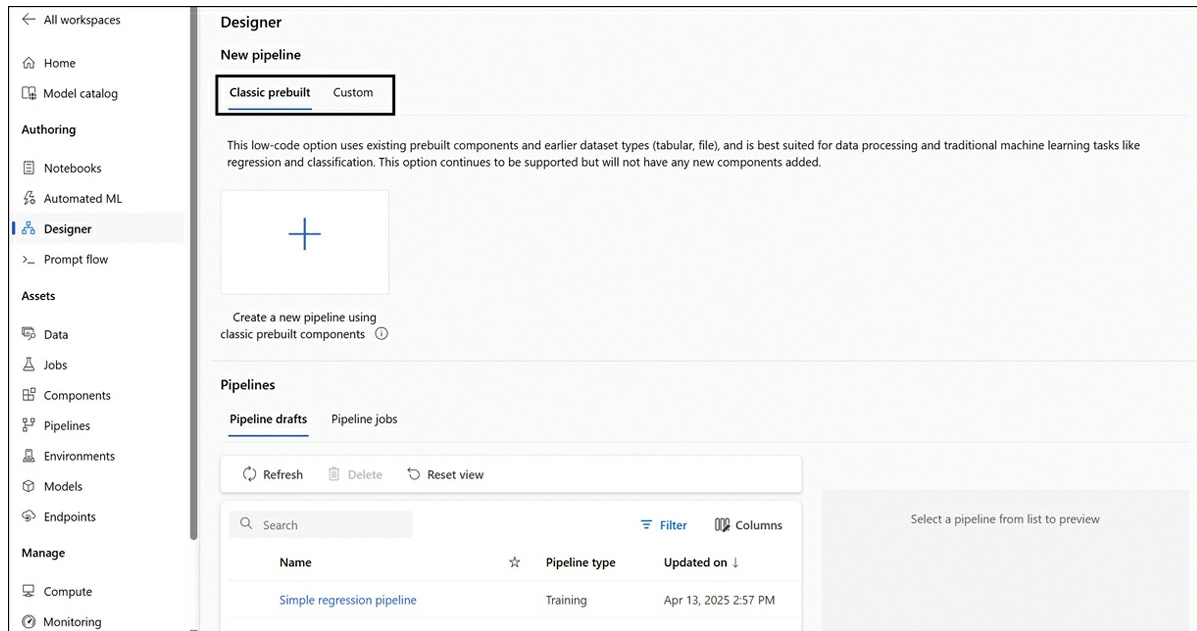


FIGURE 2-59 Azure Machine Learning Designer

To help you get familiar with the interface and the main elements of Azure ML Designer, select Classic Prebuilt and open a previously created pipeline. (You'll learn to build and configure a pipeline step by step in the next section.) With a pipeline open, you can see an overview of the canvas, components, and settings (Figure 2-60).

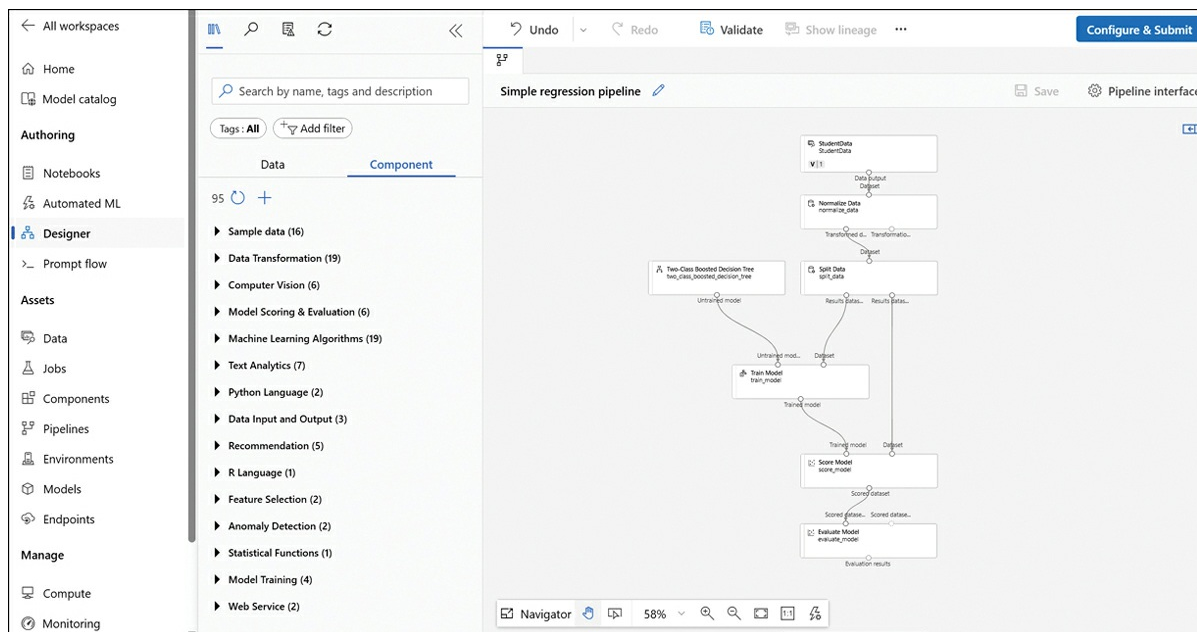


FIGURE 2-60 Sample pipeline in Azure Machine Learning Designer

Notice the two tabs on the left: The Data tab displays the datasets available in your workspace, and the Component tab lists all the modules that you can add to the canvas, including various processing steps, model training components, and a selection of example datasets. To the right, you can see the workflow for the pipeline, with a dataset as the first step at the top and the final evaluate step at the bottom. While you can use the Designer as a no-code tool, choosing the Custom option when creating a new pipeline allows you to insert blocks that execute custom Python or R code within the workflow.

Clicking Configure & Submit will run the pipeline. To run a pipeline, however, you need an experiment. You can select an existing experiment or create

a new experiment at this step. You must also specify the compute cluster on which the pipeline will run. As the pipeline is running, the currently active step is highlighted, so you can follow the progress of the pipeline run, as shown in [Figure 2-61](#).

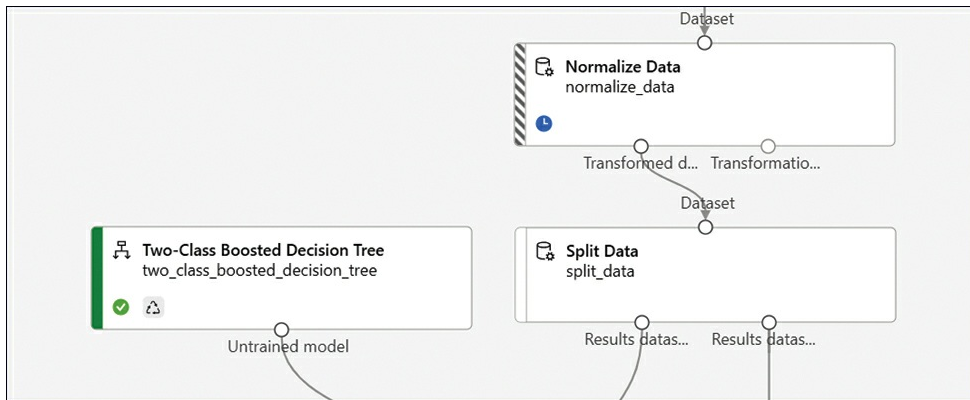


FIGURE 2-61 Pipeline in progress

You can click a step to see its properties. In [Figure 2-62](#), for example, the Split Data step properties show the mode for splitting the data and the proportion to split.

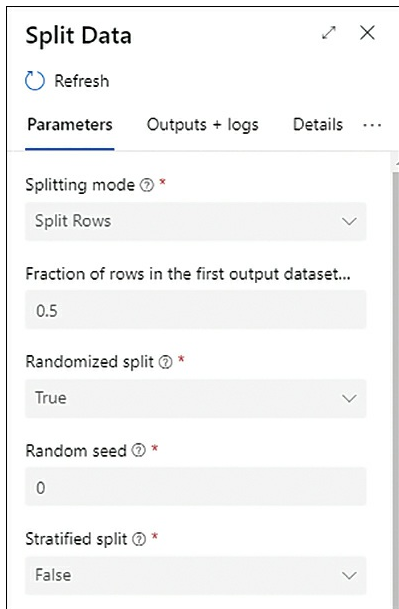


FIGURE 2-62 Step properties

When a pipeline run finishes, you can view the metrics generated for the model ([Figure 2-63](#)) by right-clicking the Evaluate Model step.

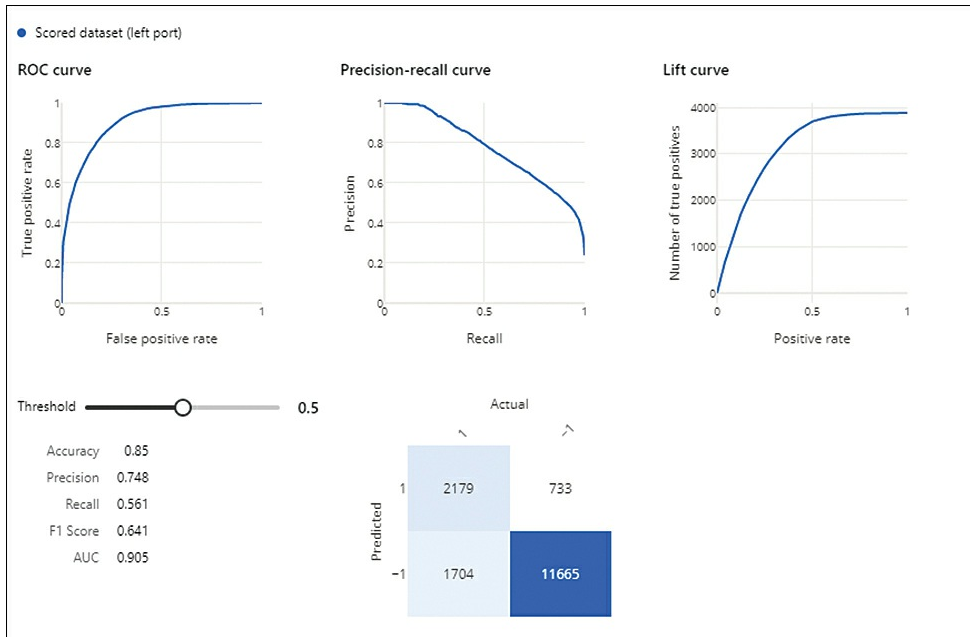


FIGURE 2-63 Evaluation metrics

Create a new model with Designer

To create a new model, you need to first create a blank training pipeline (Figure 2-64) to ingest and preprocess a dataset, and then train and evaluate the model. Training pipelines always start with a dataset, and expanding the Datasets tab will show you those that are available. For example, Figure 2-65 shows three datasets, including the student exam results dataset you registered earlier in this chapter. To add a dataset, simply drag it to the canvas.

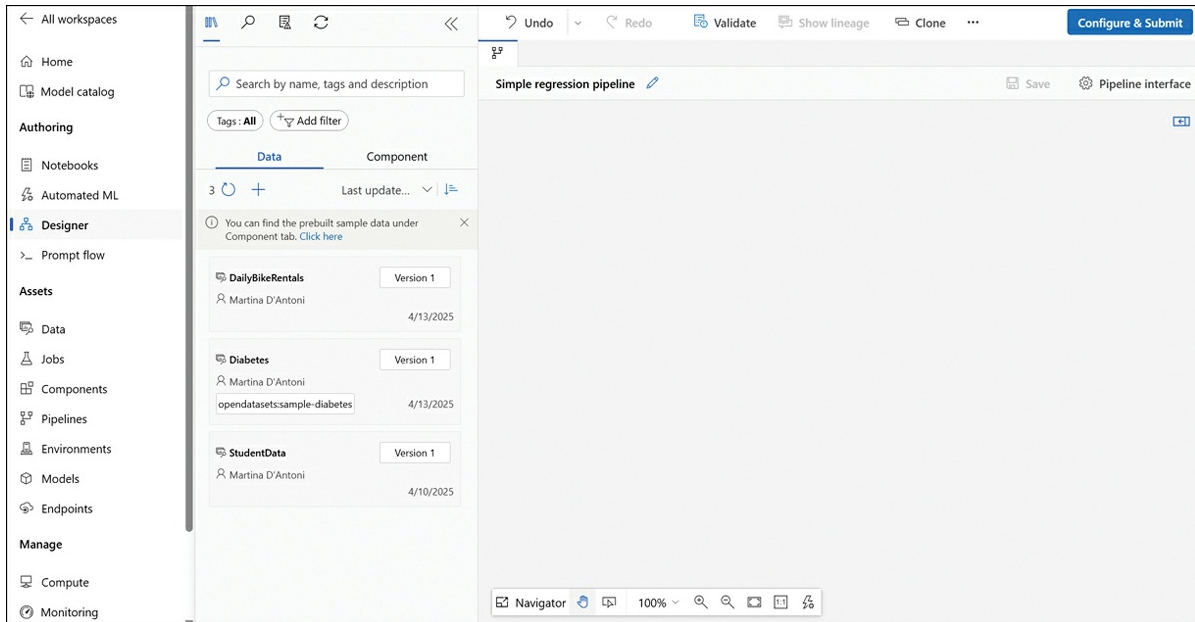


FIGURE 2-64 Blank pipeline in Azure Machine Learning Designer

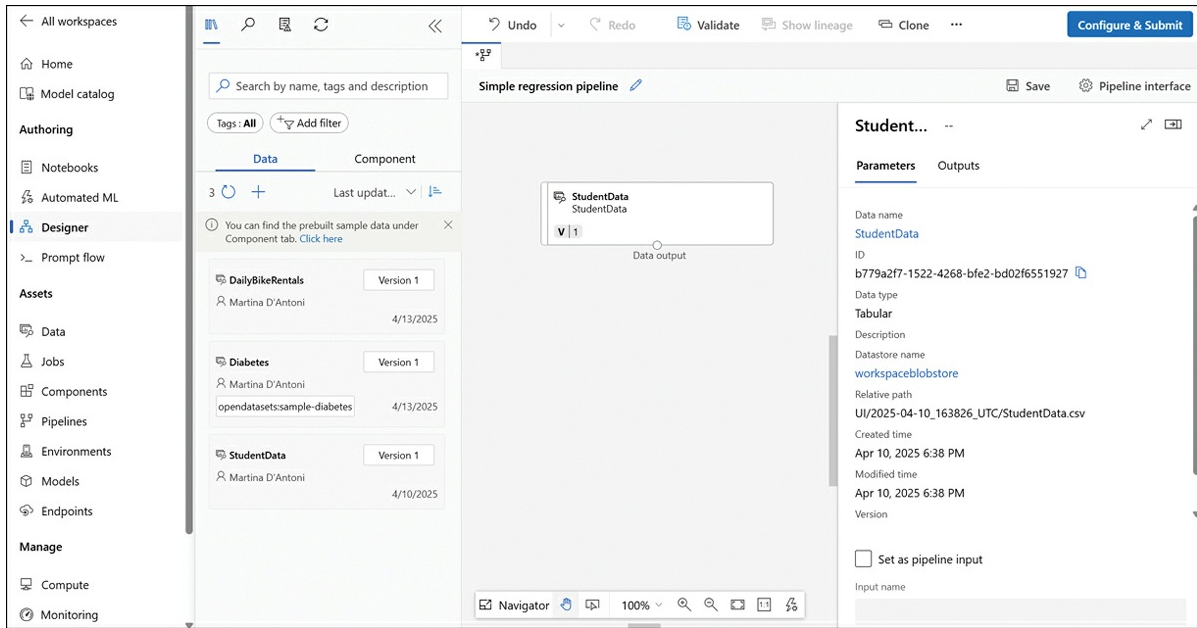


FIGURE 2-65 Dataset added to the Designer canvas

Before training, data often must be cleansed. There are many modules available to transform and cleanse data. If you have numeric features, you should normalize them, for example. You can find the Normalize Data transformation in the Data Transformation section of the Component tab. Drag it to the canvas, then drag the output from the dataset step to the top of the normalize step, as shown in Figure 2-66.

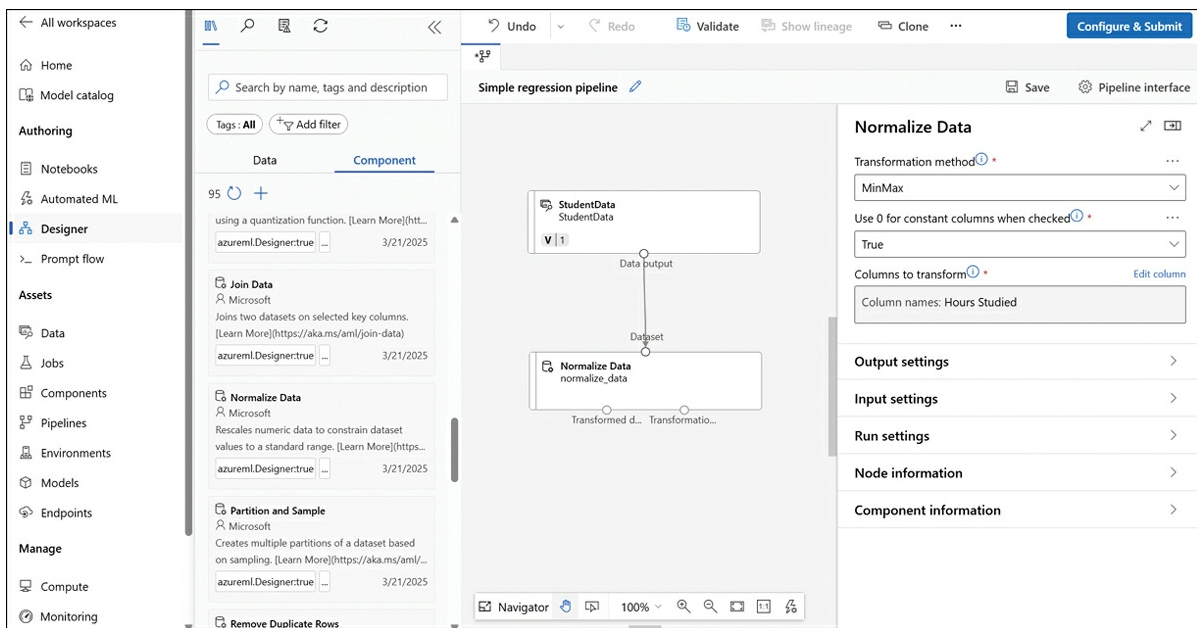


FIGURE 2-66 Normalize Data added to the Designer canvas

In the Normalize Data properties, next select the Transform Method, such as MinMax, which scales all values to a range between 0 and 1. Next, specify the numeric columns you want to normalize, such as Hours Studied for the example.

Next, you can add a Split Data module to randomly split the dataset into training and validation subsets. This is a common step in a supervised learning pipeline so that you can evaluate the model using data not used for training. You can specify the ratio of data used for training and for testing. The Split Data component, like Normalize Data, is located in the Data Transformation section within the Component tab.

The training data subset is connected to the Train Model module. You select the label in the Train Model module, which is also connected to an algorithm, as shown in Figure 2-67. You can find the Train Model component in the Model Training section within the Component tab and select the algorithm to be used from the Machine Learning Algorithms section.

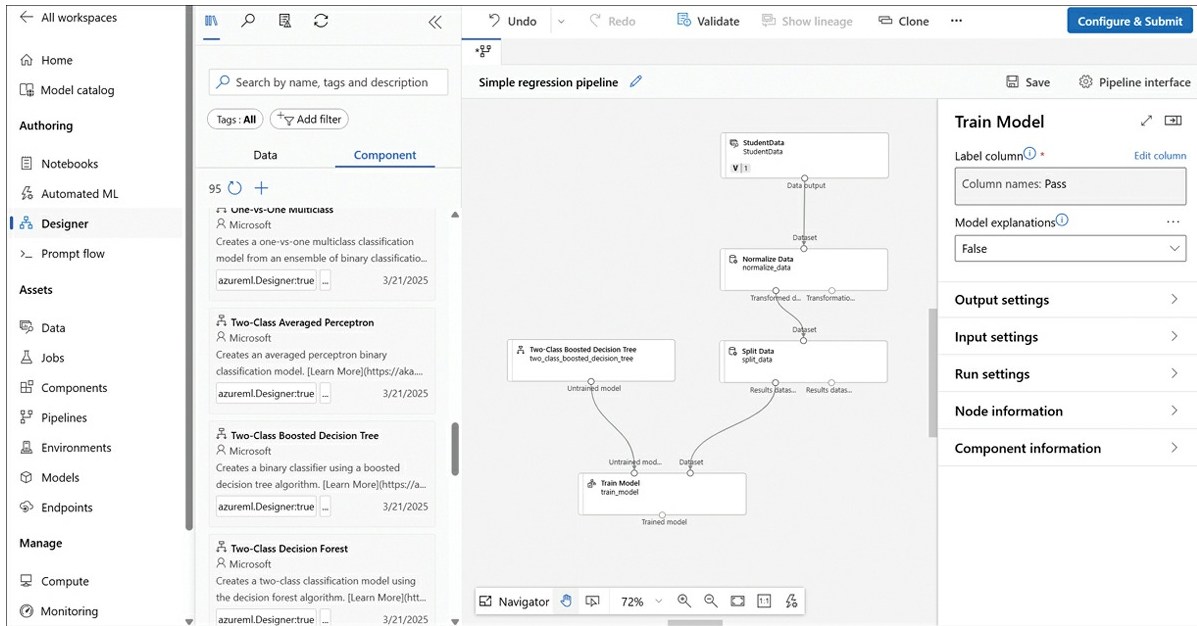


FIGURE 2-67 Train Model added to the Designer canvas

After training, you need to add a Score Model module to score the testing dataset from the Split Data step, as shown in Figure 2-68. The Score Model component is located in the Model Scoring and Evaluation section within the Component tab.

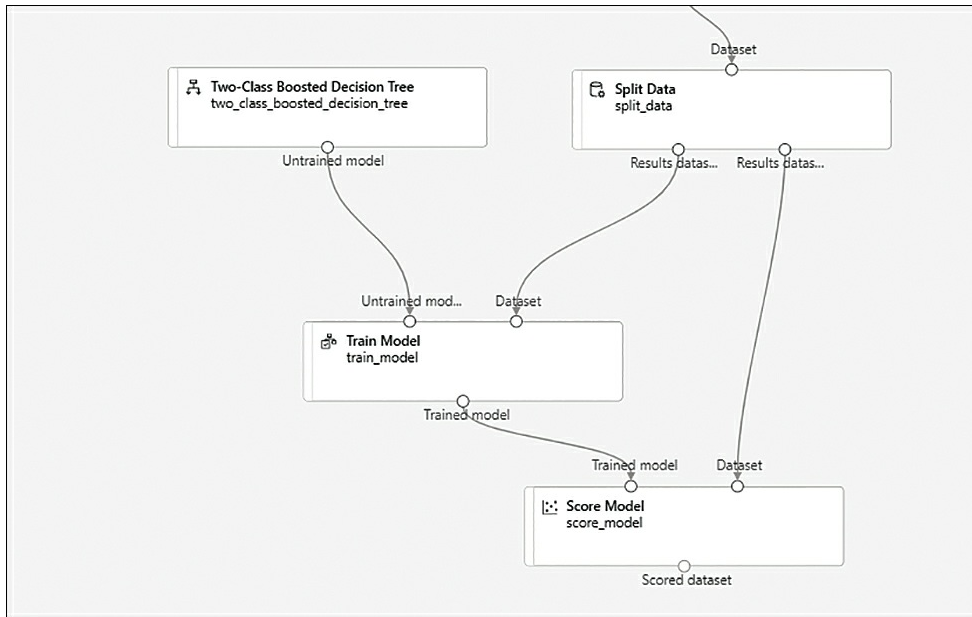


FIGURE 2-68 Score Model added to the Designer canvas

The Score Model module is required before you can evaluate the model. It is the component used to generate the actual prediction.

Add the Score Model module between the Train Model and Evaluate Model steps in the pipeline. Score Model has two inputs, one from the Train Model and the other from Split Data. You connect the second output from the Split Data step to the second input on the Score Model.

The testing data includes the actual label values. You can compare the actual values to the predicted labels generated in the Score Model step and use the Evaluate Model module to calculate the metrics to help you determine the performance, or accuracy, of the model.

Figure 2-69 shows the complete pipeline.

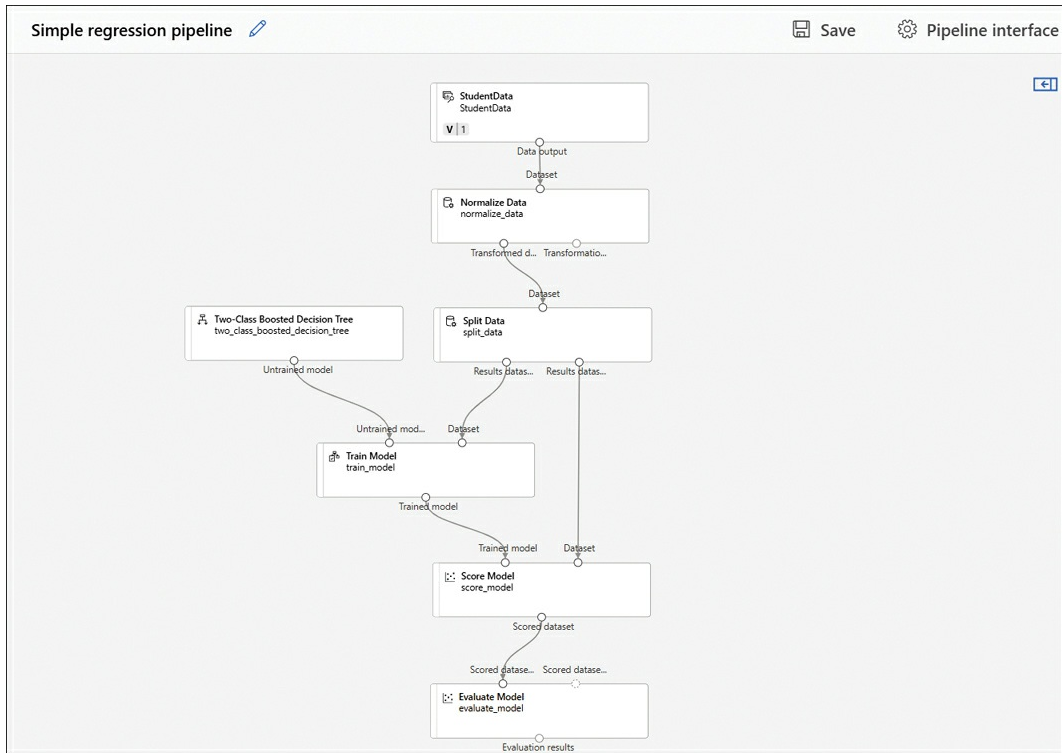


FIGURE 2-69 Evaluate Model added to the Designer canvas

This pipeline is the common workflow to train a supervised model, such as regression or classification. You may add steps between the dataset and normalize data to select columns, cleanse data, and transform data depending on your dataset and requirements.

Clicking Configure & Submit (above Designer canvas on right) opens a new window to start a run of the pipeline, as shown in [Figure 2-70](#).

Set up pipeline job 🔗

- 1 Basics
- 2 Inputs & outputs
- 3 Runtime settings
- 4 Review + Submit

Basics

Experiment name
 Select existing Create new

New experiment name * 👁

Job display name

Job description

Job tags

<input type="text" value="Name"/>	:	<input type="text" value="Value"/>	<input type="button" value="Add"/>
-----------------------------------	---	------------------------------------	------------------------------------

FIGURE 2-70 Set up a pipeline run

To run a pipeline, you need an experiment. You can either use an existing experiment or create a new experiment. In addition to selecting an experiment, you must also specify the compute cluster on which the pipeline will run, as shown in [Figure 2-71](#).

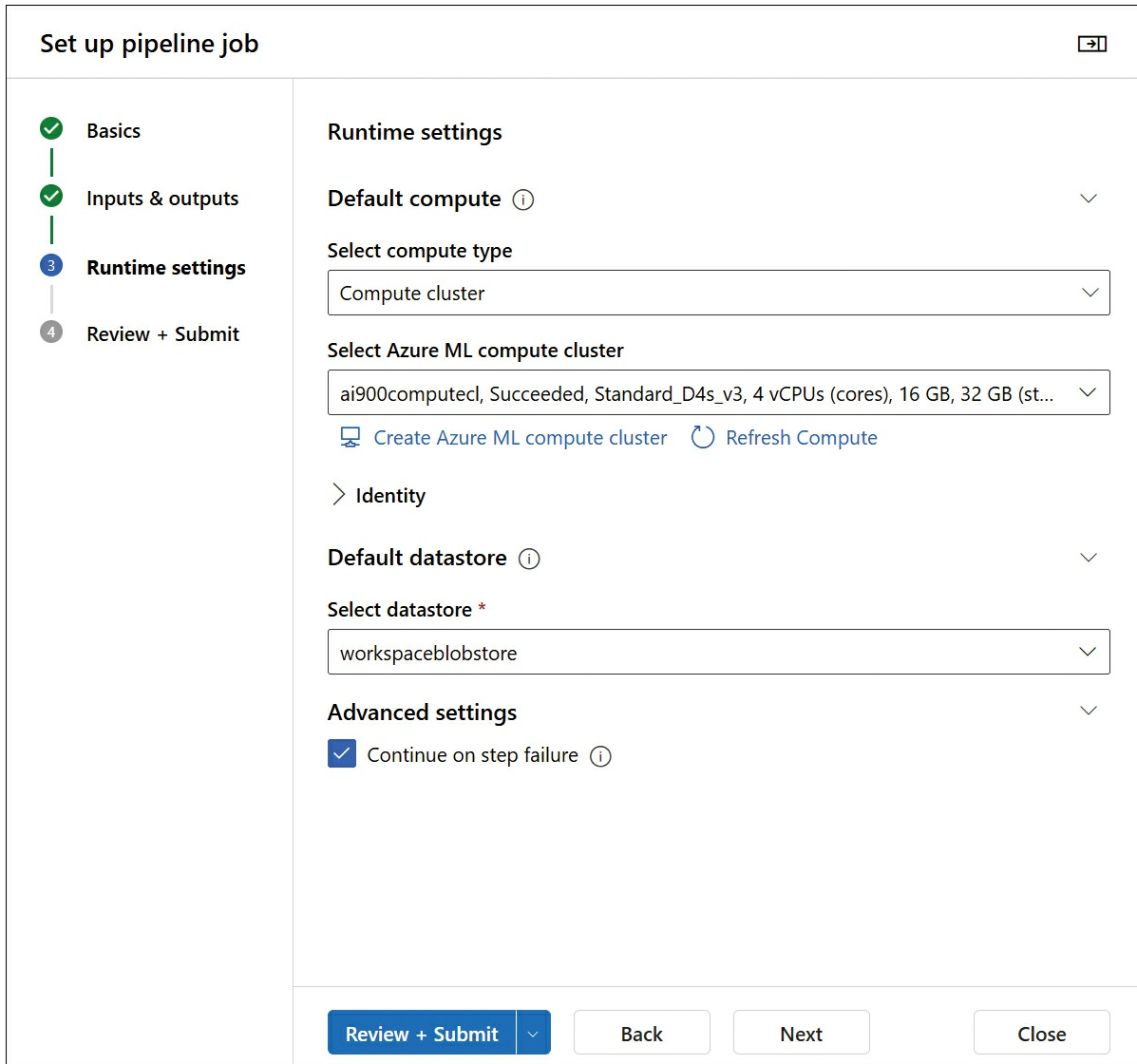


FIGURE 2-71 Set up a pipeline run

Clicking Review + Submit queues the pipeline to be run.

Once the pipeline run has completed, you can use the metrics to determine the accuracy of the model.

Deploy a model with Designer

Once you are confident with the model's performance, you can register the model. To do so, select the Outputs + Logs tab in the properties on the Train Model step, as shown in [Figure 2-72](#).

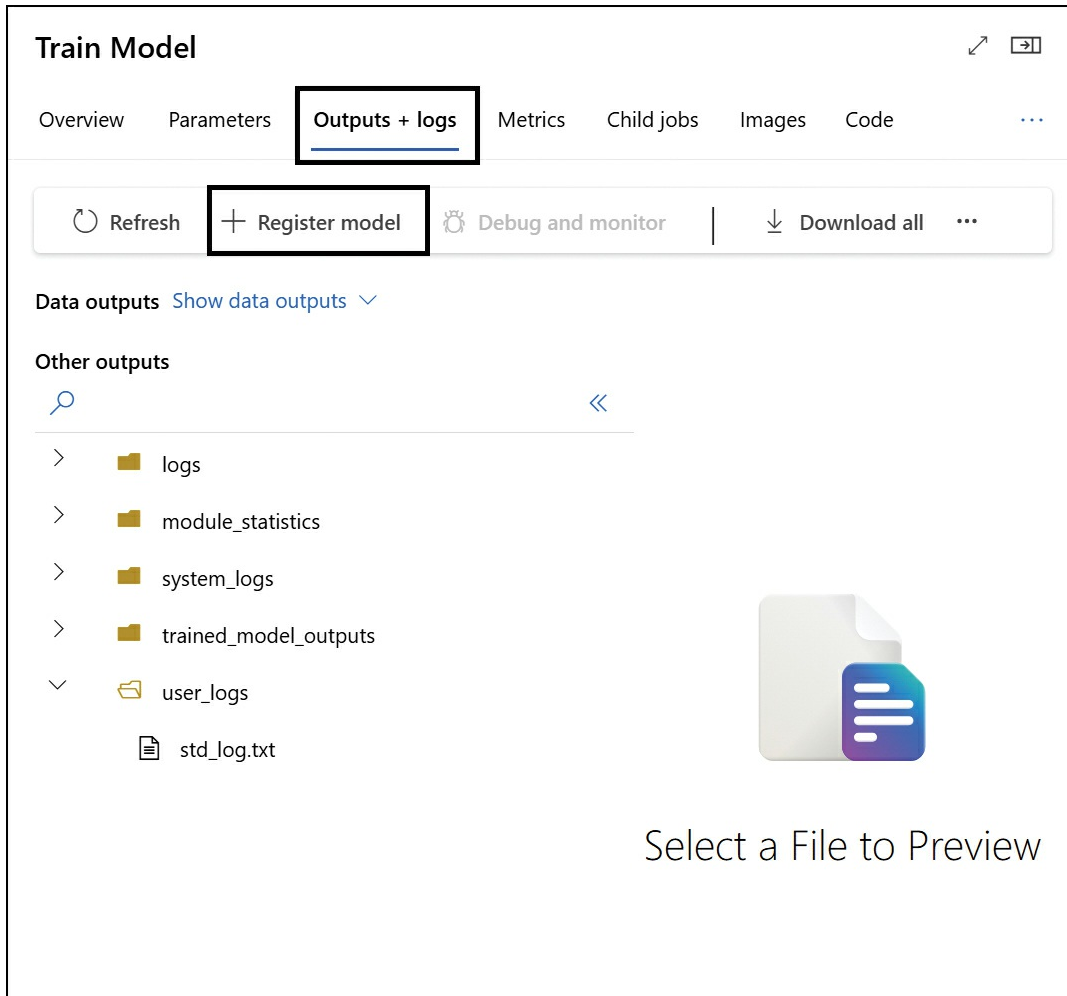


FIGURE 2-72 Train Model properties

Click Register Model to open a wizard in which you can specify the model type (Triton, MLFlow, or Unspecified), select the output of the job from which to register the model, and define the model's name.

You can see a list of models in the Machine Learning Studio, as shown in Figure 2-73.

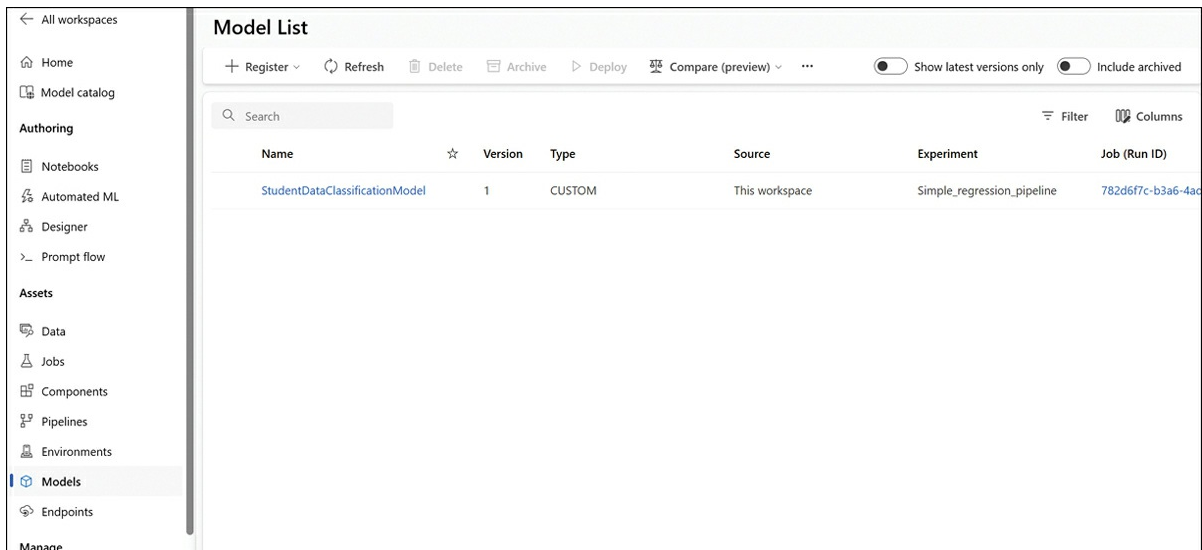


FIGURE 2-73 List of models

To use the model, select it from the models list, then select a model type for Use This Model, as shown in [Figure 2-74](#).

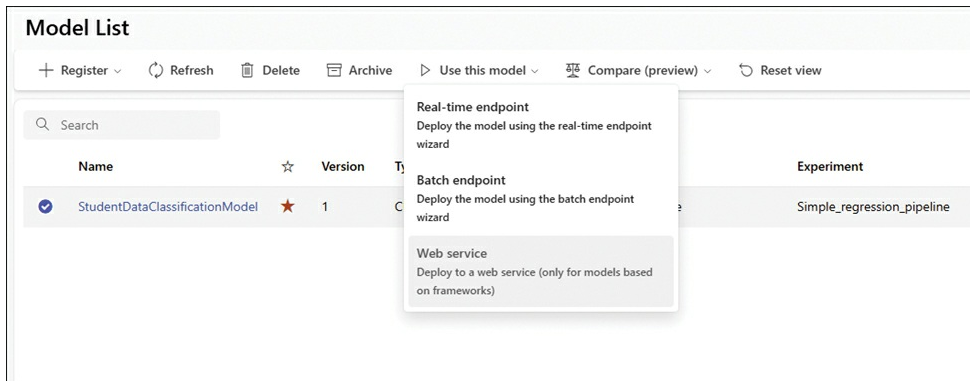


FIGURE 2-74 Deploy a model from the models list

You will need to write a scoring script in Python, which is the actual code executed when new requests are sent to the inference endpoint.

The downside of this approach is that you must manually reimplement all the transformations (such as normalization) that were performed in the pipeline.

A simpler alternative is to run the pipeline and then select Create Inference Pipeline, as shown in [Figure 2-75](#).

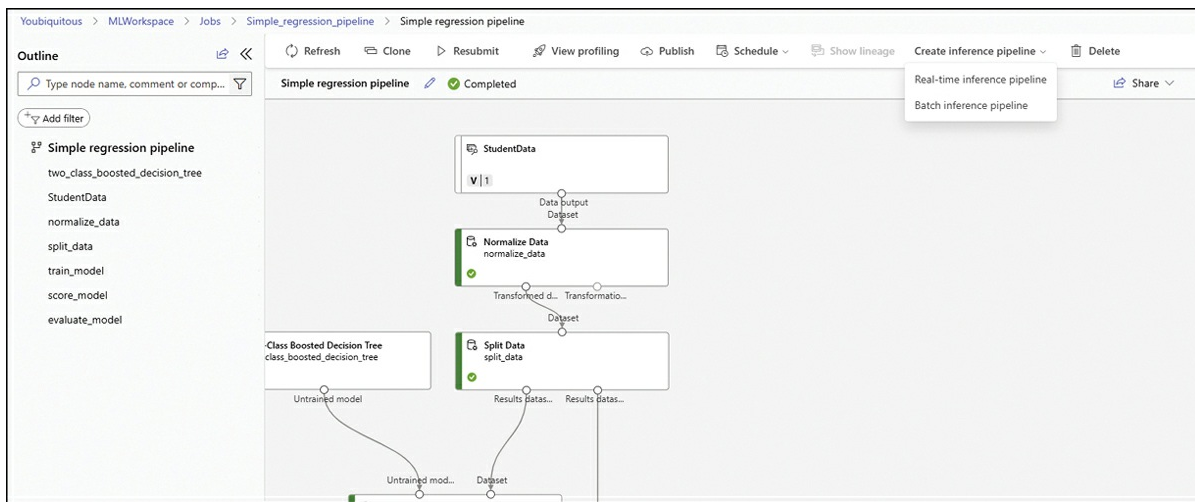


FIGURE 2-75 Deploy a model from the pipeline designer

From here, you should add the Web Service Input component and optionally remove the Evaluate Model component to create a final pipeline like [Figure 2-76](#).

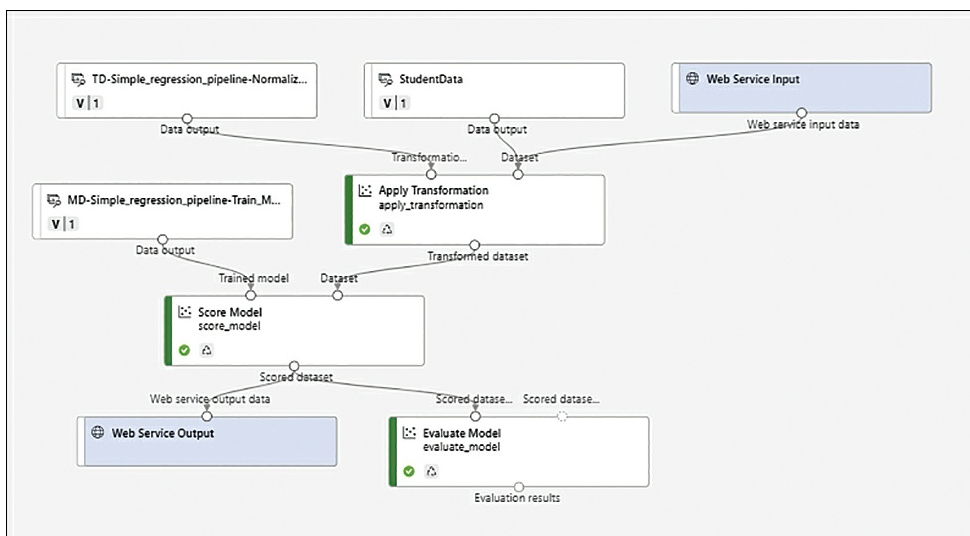


FIGURE 2-76 The final pipeline

Note Passing the Target Column

With this configuration, you should provide the model with the column(s) you want to predict in the input schema. The model won't use that column, so you can pass it an empty or null value.

The last step is clicking the Deploy button and choosing the inference endpoint name and compute type between the Azure Container Instance or Azure Kubernetes Service cluster. The generated endpoint will be available in the Endpoints section.

Chapter summary

In this chapter, you learned the fundamental concepts of machine learning and how they relate to modern AI workloads. You explored key types of models, data preparation techniques, evaluation methods, and how Azure supports building and deploying machine learning solutions. Here are the key concepts from this chapter:

- Machine learning is the basis for modern AI.
- Machine learning uses data to train models.
- Feature is the name for the data used as inputs to a model.
- Label is the name for the data that the model predicts.
- Supervised learning requires data with features and labels to train the model.
- Unsupervised learning trains with data without labels.
- Regression models predict a value along a range of values.
- Classification models predict a discrete class or category.
- The Transformer architecture is the most widely used architecture for language models.
- Regression and classification models are both examples of supervised learning.
- Clustering models group data by similarities in the data into discrete categories.
- Clustering is an example of unsupervised learning.
- Feature selection is the process of selecting a subset of relevant features for training a machine learning model.
- Feature engineering is the process of creating new features from raw data to increase the predictive power of the machine learning model.
- Normalization is a common process in data preparation that changes all numeric data to use the same scale to prevent bias.
- You split your dataset into a training dataset and a testing dataset.
- Training a model combines the training dataset with an algorithm.
- After training is complete, you score and then evaluate the model.
- Scoring is performed using the testing dataset. Scoring compares the actual values with the predicted values from the model.
- Evaluation generates metrics that determine how well the model performs.
- The metrics for regression models calculate the difference between the actual and predicted values using averages and means.
- The metrics for classification models measure whether the prediction is a true or false positive or negative. A confusion matrix represents the number of true/false positive/negative predictions. The metrics for classification models are ratios of these predictions.
- You need an Azure subscription to use Machine Learning on Azure.
- You need to create an Azure Machine Learning workspace to build and train models.
- The Azure Machine Learning workspace manages assets associated with machine learning, including datastores, datasets, experiments, models, and compute.
- Compute instances support data preparation and building of models.
- Compute clusters support training of models.
- You can deploy a model to a managed online endpoint, Azure Container Instances, or Azure Kubernetes Service.
- Azure Machine Learning Studio can be used to author models using Python with Jupyter notebooks, Automated ML, or the Designer.
- .NET is not supported in Azure Machine Learning Studio.
- You can build and manage models using Visual Studio Code with .NET and Python.
- Automated Machine Learning (Automated ML) uses the power of the cloud to create multiple models for the same data using different algorithms and then determining which is the best fit model.
- Azure Machine Learning Designer is a graphical user interface that allows you to drag modules into a workflow to build a model.

Thought experiment

In this thought experiment, demonstrate your skills and knowledge of the topics covered in this chapter. You can find the answers in the section that follows.

You work for Relecloud, an internet service provider that collects large volumes of data about its customers and the services they use. You are evaluating the potential use of machine learning models to improve business decision-making and operations. Relecloud uses Azure for virtual machines but has no experience of machine learning in Azure. You need to advise on the types of machine learning models to build and the languages and tools that you should use within your organization. You also need to explain to IT which Azure resources are required to be provisioned to use machine learning in Relecloud's applications. Answer the following questions:

1. Which machine learning type should you use to determine if a social media post has positive or negative sentiment?
2. Which type of machine learning groups unlabeled data using similarities in the data?
3. Which two datasets do you split your data into when building a machine learning model?
4. Which metrics can you use to evaluate a regression machine learning model?
5. Which metrics can you use to evaluate a classification machine learning model?
6. Which compute target should you use for development of machine learning models?
7. What is used to increase the predictive power of a machine learning model?
8. What should you do to prevent your model from being biased by one feature?
9. What should you do to enable a trained machine learning model to be measured for accuracy?
10. What should you do after training your model prior to deploying your model as a web service?
11. You use Automated ML to find the best model for your data. Which option should you use to interpret and provide transparency for the model selected?
12. Which two types of datasets can you register and use to train Automated Machine Learning (Automated ML) models?

Thought experiment answers

This section contains the solutions to the thought experiment. Each answer explains why the answer choice is correct.

1. The classification machine learning model is used to predict mutually exclusive categories. Classification involves learning using labels to classify data. Two-class classification algorithms can provide an either/or answer such as a social media post having a positive or negative sentiment.
2. The clustering machine learning model analyzes unlabeled data to find similarities in data points and groups them together into clusters. The clustering algorithm segments data into multiple groups based on similarities in the data.
3. The dataset is split into a training dataset and a testing dataset. The testing dataset is often referred to as the validation dataset.
4. Root mean squared error (RMSE) is used to evaluate regression models. RMSE is a measure of the difference between predictions and actual values. The closer RMSE is to 0, the better the model is performing. You can also use coefficient of determination.
5. Precision is a measure of the correct positive results. Precision is the number of true positives divided by the sum of the number of true positives and false positives. Precision is scored between 0 and 1, with closer to 1 being better. You can also use accuracy and F-score.
6. A compute instance is a configured development environment for machine learning.
7. Feature engineering is the process of creating new features from raw data to increase the predictive power of the machine learning model.
8. Normalization is the method that adjusts the values in numeric columns so that all numeric values are on a common scale, normally between 0 and 1. A dataset that has values using different scales can bias the model toward that feature.
9. You should score the model. After a model has been trained, the model should be evaluated using a different set of data. Scoring applies the validation dataset to the trained model to generate predictions that can be evaluated using metrics that measure the accuracy of the predictions.
10. After training your model, you need to create an inference pipeline that performs the same steps for new data input, not the sample data used in training.
11. Explain Best Model generates an explanation for the model that was identified as the best model.
12. File and tabular datasets can be used in Azure Machine Learning training workflows.

Chapter 3

Describe features of computer vision workloads on Azure

Azure AI Services is a suite of prebuilt AI services that developers can use to build AI solutions. AI Services meet common AI requirements that allow you to add AI to your apps more quickly with less expertise.

This chapter will begin with an overview of all AI Services, but then will focus on one of the major components of AI Services: the AI Vision service. Computer vision models, such as the Azure AI Vision service, can analyze a still image or video stream, and then provide an interpretation of it in computer-readable form. With the help of use cases, the chapter will outline the concepts involved in computer vision, then it will delve into how to use the Azure AI Vision service.

Chapter 4, “Describe features of natural language processing (NLP) workloads on Azure,” will elaborate on the other major components of AI Services: AI Language, AI Speech, and AI Translator.

Skills covered in this chapter:

- Skill 3.1: Identify common types of computer vision solutions
- Skill 3.2: Identify Azure tools and services for computer vision tasks

Skill 3.1: Identify common types of computer vision solutions

Computer vision is the processing of still images and video streams and extracting information from those images. Computer vision can interpret the image and provide detail and understanding about it in a computer-readable form. Computers can then perform further processing and analysis on this information. Many applications use computer vision to enhance user experience or to capture information about objects and people.

Microsoft Azure provides a set of services around computer vision as part of Azure AI Services. You can also use Azure Machine Learning to create your own image-processing models.

A focus of the Microsoft Azure AI Fundamentals certification is on the capabilities and features of computer vision and how computer vision can be applied in solutions. This requires you to understand the use cases for computer vision and to be able to differentiate the various services for computer vision in Microsoft Azure.

This skill covers how to:

- Describe AI Services
- Understand computer vision
- Describe image classification
- Describe object detection
- Describe optical character recognition
- Describe facial detection, recognition, and analysis

Describe AI Services

Azure AI Services are prebuilt machine learning models trained by Microsoft with massive volumes of data that developers, even those without machine learning skills, can use to build AI solutions. AI Services are focused on a subset of common AI requirements around processing images and analyzing text.

AI Services are available as a set of REST APIs that can easily be deployed and consumed by applications. Essentially, AI Services are off-the-shelf services that help you develop an AI-based solution more quickly and with less specialist expertise.

Overview of AI Services

A family of powerful, cloud-based AI services and APIs, Azure AI Services enables applications to see, hear, speak, search, understand, and assist with decision-making. The services are presented as individual offerings in the Azure portal, each designed to address specific AI scenarios so you can easily integrate the AI capabilities you need into your applications with minimal machine learning expertise. At this writing, the current catalog of Azure AI Services available includes:

- **Azure AI Agent Service** Combine generative AI models with tools that allow agents to access and interact with real-world data sources. Facilitates tool-calling agent orchestration and makes grounding possible by integrating with Bing Search.
- **Azure AI Model Inference** Perform model inference for flagship models in the Azure AI model catalog, with serverless access to a variety of AI foundation models.
- **Azure AI Search** Bring AI-powered vector and semantic search to your web and mobile apps. Offers full-text, vector, and hybrid search options for your own data and content.
- **Azure OpenAI Service** Access powerful language models, such as GPT-4, for natural language processing applications like content generation, code completion, and chatbots. Brings enterprise-grade security and compliance to foundation models.

- **Bot Service** Create intelligent bots and deploy them on multiple channels and platforms, to bring conversational interfaces to applications.
- **Content Safety** Detect potentially harmful or inappropriate content across text and images to maintain responsible use of AI applications. This service replaces the retiring Content Moderator service.
- **Custom Vision** Create custom image classifiers and object detectors without requiring deep machine learning knowledge. Train computer vision models with your own images to solve specific problems.
- **Document Intelligence** Extract structured data from documents like invoices, receipts, and contracts, converting unstructured content into insights. Formerly known as Form Recognizer.
- **Face** Detect and identify people and their attributes, such as emotions, in images. Useful for facial recognition in authentication scenarios.
- **Immersive Reader** Help users read and comprehend text with audio and visual cues. Makes text more legible for readers of any age and skill, enabling inclusive design and improved literacy.
- **Language** Build apps with industry-leading natural language understanding capabilities like entity recognition, sentiment analysis, key phrase extraction, summarization, and custom question answering. This service replaces the retired QnA Maker service.
- **Speech** Convert speech to text, convert text to speech, and perform real-time translation. Key features include Speech to Text (which transcribes audio into readable, searchable text), Text to Speech (which synthesizes natural-sounding speech), and Speech Translation (which converts spoken audio into translated text in real-time).
- **Translator** Translate more than 100 languages and dialects with neural machine translation. Supports real-time or batch processing of in-use, at-risk, and endangered languages.
- **Video Indexer** Extract actionable insights from your videos, such as speech, faces, and objects. It processes both the visual and audio tracks to automatically identify speakers, written and spoken text, faces, objects, and scenes.
- **Vision** Analyze content in images and video to extract information, detect objects, read text (OCR), generate image descriptions, and more. This service is also known as Azure AI Vision and Computer Vision. For simplicity, this book will refer to it as Vision.

Note Retired or soon-to-be Retired Services

Several services that were part of the original Azure Cognitive Services (renamed in Azure AI Services) have been deprecated and are scheduled for retirement. These include Anomaly Detector, Metrics Advisor, Personalizer, Content Moderator, Language Understanding (LUIS), and QnA Maker. Although some may still work for existing applications, you shouldn't use them for new projects, as they'll be turned off between 2025 and 2027. Most of their capabilities have been integrated into newer services like Azure AI Language, Azure AI Content Safety, and Azure Machine Learning.

Note Features Being Phased Out

Some features within active services are also being retired. Speaker Recognition will no longer be available after September 30, 2025, so plan to transition if you rely on it. Also, Spatial Analysis from the Vision family was retired on March 30, 2025. For those use cases, alternatives like Video Indexer or custom solutions are recommended.

Deploy AI Services

AI Services are easily deployed in Azure as resources. You can use the Azure portal, the CLI, or PowerShell to create resources for AI Services. There are even Azure Resource Manager (ARM) templates available to simplify deployment further. Once created, the APIs in AI Services will then be available to developers through REST APIs and client library SDKs.

You have two options when creating resources:

- Multi-service resource
- Single-service resource

With a multi-service resource, you have access to all the AI Services with a single key and https endpoint. Benefits of a multi-service resource are:

- Only one resource to create and manage
- Access to Vision, Language, and Speech services using a single API
- Consolidates billing across all services

With a single-service resource, you access a single AI Service with a unique key and https endpoint for each individual service. Benefits of a single-service resource are:

- Limits the services that a developer can use
- Dedicated API for each service
- Separate billing of services
- Free tier available

The multi-service resource is named AI Services in the Azure portal. To create a multi-service AI Services resource in the Azure portal, search for "AI Services" and select AI Services by Microsoft.

Figure 3-1 shows the service description for the AI Services multi-service resource.

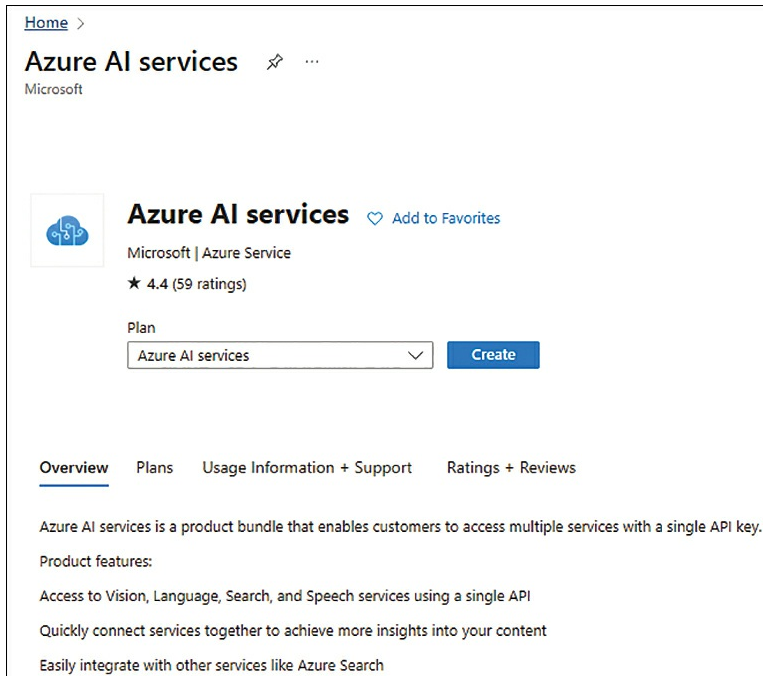


FIGURE 3-1 AI Services multi-service resource description

Click the Create button to open the Create Azure AI Services pane, as shown in Figure 3-2. Here, you can select the subscription, resource group, and region where the resource is to be deployed, as well as give the service a unique name. This name will be the domain name for your endpoint and so must be unique worldwide. You should then select your pricing tier. There is only one pricing tier for the multi-service resource, Standard S0. Click Review + Create to validate the options, then click Create to create the resource. The resource will be deployed in a few seconds.

Home > Azure AI services >

Create Azure AI services

Basics Network Identity Tags Review + create

Get access to Vision, Language, Search, and Speech Azure AI services with a single API key. Quickly connect services together to achieve more insights into your content and easily integrate with other services like Azure Search.

[Learn more](#)

Project Details

Subscription *

Resource group * [Create new](#)

Instance Details

Region

Name * ✓

Location ⓘ
 Location specifies the region only for included regional services. This does not specify a region for included non-regional services. [Click here for more details.](#)

Pricing tier *

[View full pricing details](#)

By checking this box I acknowledge that I have read and understood all the terms below *

Responsible AI Notice

Microsoft provides technical documentation regarding the appropriate operation applicable to this Azure AI service that is made available by Microsoft. Customer acknowledges and agrees that they have reviewed this documentation and will

FIGURE 3-2 Creating an AI Services resource

You can create an AI Services resource using the CLI as follows:

```
az cognitiveservices account create --name <unique name> --resource-group
<resource group name> --kind AIServices --sku S0 --location <region> --yes
```

To create a single-service resource for the Vision service using the Azure portal, you should search for “Vision” and create the resource. The options are the same as the multi-service resource, except you can select the Free pricing tier.

You can create a Vision resource using the CLI as follows:

```
az cognitiveservices account create --name <unique name> --resource-group
<resource group name> --kind Vision --sku F0 --location <region> --yes
```

If you want to create other single-service resources, use the following CLI command to find the correct values for kind:

```
az cognitiveservices account list-kinds
```

After your resource has been created, you need to obtain the REST API URL and the key to access the resource.

Use AI Services securely

Each resource you create will have a unique endpoint for the REST API and authentication keys. To use AI Services from your app, you need these details. You can easily view the endpoint and keys in the Azure portal: Navigate to the resource and select Keys and Endpoint, as shown in [Figure 3-3](#).

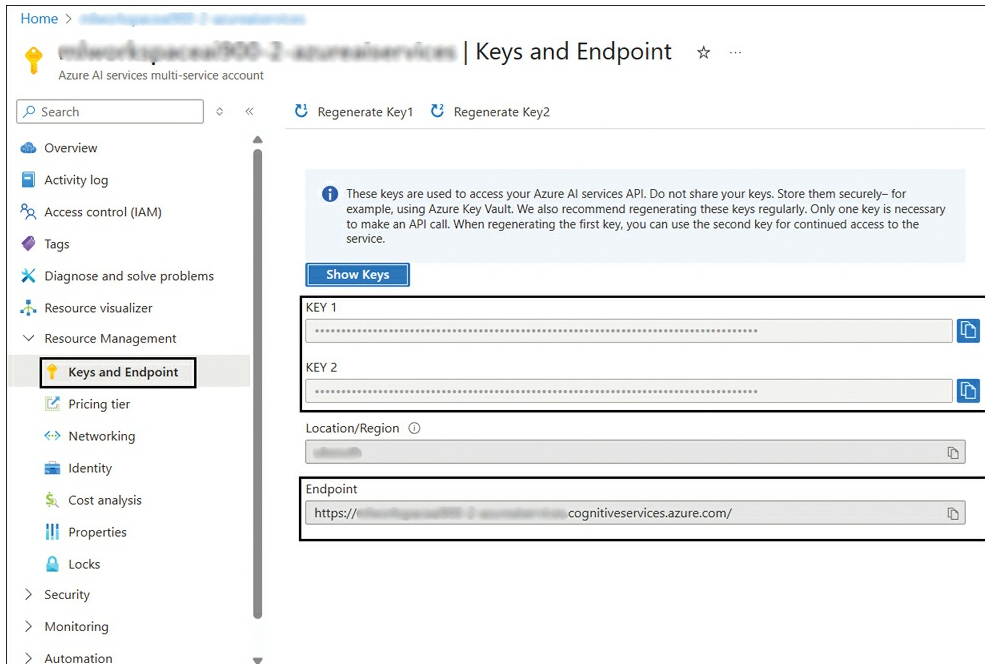


FIGURE 3-3 Keys and Endpoint

You can access the keys using the CLI as follows:

```
az cognitiveservices account keys list --name <unique name> --resource-group <resource group name>
```

Exam Tip

Practice creating single- and multi-service resources in the Azure portal and make sure you know where the endpoint and keys can be found.

Containers for AI Services

AI Services are also available as Docker containers for deployment to IoT devices and to on-premises systems. Containers provide advantages with hybrid and disconnected scenarios and allow higher throughput with lower latency of data.

Only some AI Services are available in containers.

Need More Review? Azure AI Services Containers

For more information on using AI Services with containers AI, see learn.microsoft.com/en-us/azure/ai-services/cognitive-services-container-support.

Understand computer vision

Computer vision is the interaction with the world through visual perception. A computer sees an array holding the color and intensity as number values. Computer vision analyzes these values using prebuilt models to detect and interpret the image. Computer vision also makes it easy for developers to process and label visual content in their apps. The Vision service API can describe objects in images, detect the existence of people, and generate human-readable descriptions and tags, enabling developers to categorize and process visual content.

Key features of computer vision

Some other key features of computer vision include the ability to:

- Categorize images
- Determine the image width and height
- Detect common objects including people
- Analyze faces
- Detect adult content

Use cases for computer vision

There are many uses for computer vision:

- In retail stores, a network of cameras can detect a shopper taking an object from the shelf and adding it to their basket.
- In vehicles, cameras can be used to detect pedestrians and cyclists, warning the driver of vulnerable road users.
- In healthcare, computer vision can analyze images of skin conditions to determine the severity with much higher accuracy than human specialists.
- In utilities, the positions of the panels on solar farms can be analyzed using cameras mounted on drones and the orientation changed to maximize efficiency.

Describe image classification

Image classification is a machine learning model that predicts the *category*, or *class*, the contents of an image belong to; 86 standard categories can be detected in an image. After a set of images is used to train the model, the model can then categorize new images. Note that categories are different from tags. *Tags* are based on the objects, people, and actions identified in the image.

Image classification can:

- Describe an image
- Categorize an image
- Tag an image

Figure 3-4 shows an example of image classification with a textual description of the image added to the bottom of the image.



FIGURE 3-4 Example of image classification

Detecting the color scheme in an image is another example of image classification. Three colors are classified in the image: the dominant foreground color, the dominant background color, and the accent color, which is the most vibrant color in the image.

Identifying products on a warehouse shelf is an example of image classification. The model will check for products against trained images added to the model.

Quality control on a manufacturing line is a similar example of image classification. Product labels and bottle caps can be verified to be correctly attached using image classification against a set of trained images of correctly labeled and sealed products.

Describe object detection

Object detection identifies and tags individual visual features (objects) in an image. Object detection can recognize many different types of objects.

Object detection will also return the coordinates for a box surrounding a tagged visual feature. Object detection is like image classification, but object detection also returns the location of each tagged object in an image.

Object detection can:

- Detect common objects
- Tag visual features
- Detect faces
- Identify brands and products
- Identify landmarks

Figure 3-5 shows an example of object detection. Three cats were identified as objects; the boxes drawn on the image indicate their coordinates.

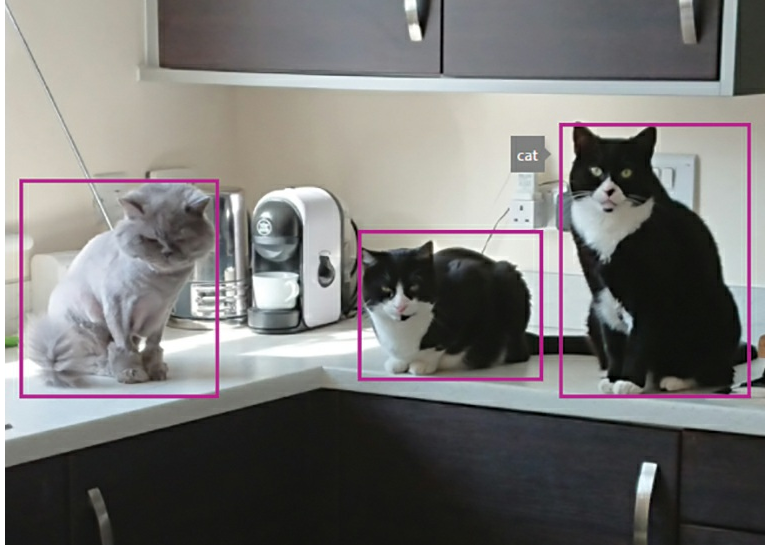


FIGURE 3-5 Example of object detection

Object detection can be used to detect objects in an image. For example, you could train computer vision to detect people wearing face masks. Facial detection does not include the ability to recognize that a face is covered with a mask, and masks may prevent faces from being recognized.

Evaluating compliance with building safety regulations is another example of object detection. Images of a building interior and exterior can be used to identify fire extinguishers, doors, and other access and emergency exits.

Describe optical character recognition

Optical character recognition (OCR) extracts small amounts of text from an image. OCR can recognize individual shapes as letters, numerals, punctuation, and other elements of text.

OCR can:

- Extract printed text
- Extract handwritten text

Using OCR, you can extract details from invoices that have been sent electronically or scanned from paper. You can then validate these details against the expected details in your finance system.

[Figure 3-6](#) shows an example of using OCR to extract text from an image.

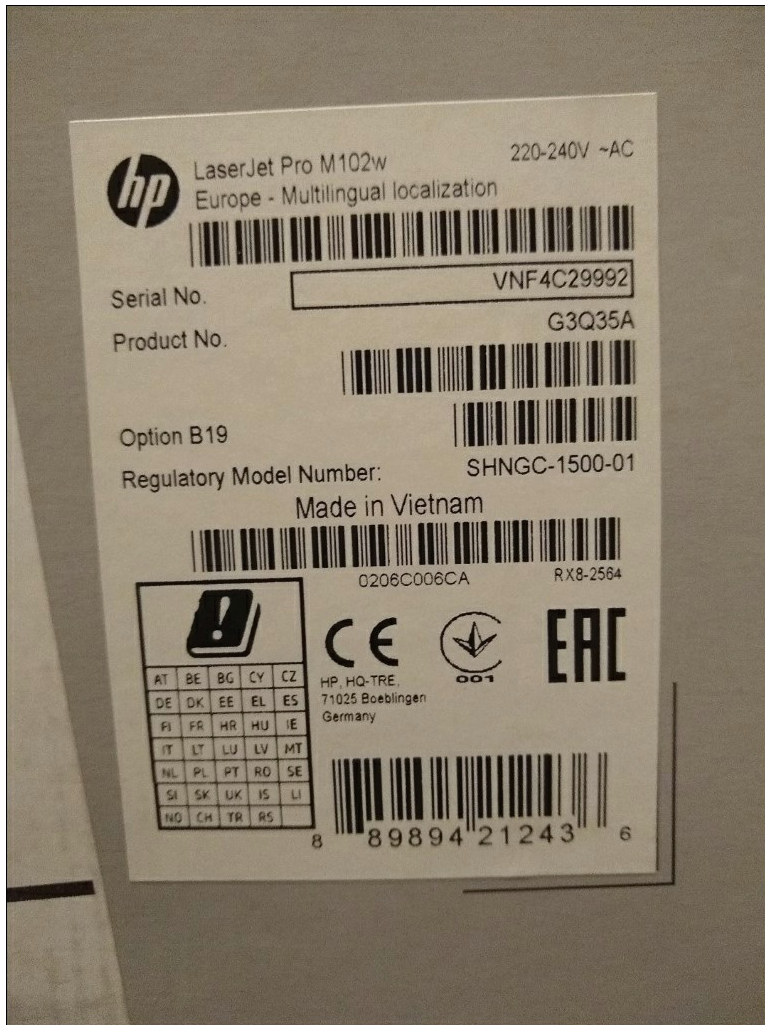


FIGURE 3-6 Example of image classification

The OCR service extracted the following pieces of text from the image:

- 220-240V ~AC
- hp
- LaserJet Pro M102w
- Europe - Multilingual localization
- Serial No.
- VNF 4C29992
- Product No.
- G3Q35A
- Option B19
- Regulatory Model Number
- SHNGC-1500-01
- Made in Vietnam

Describe facial detection, recognition, and analysis

Facial detection can provide a series of attributes about a face it has detected, including whether the person is wearing eyeglasses or has a beard. Facial detection can also estimate the type of eye covering, including sunglasses and swimming goggles.

Facial detection and recognition can:

- Detect faces

- Analyze facial features
- Recognize faces
- Identify famous people

Object detection includes the detection of faces in an image but provides only basic attributes of the face, including age and gender. Facial detection goes much further in analyzing many other facial characteristics, such as emotion.

Figure 3-7 shows an example of facial detection of the author.

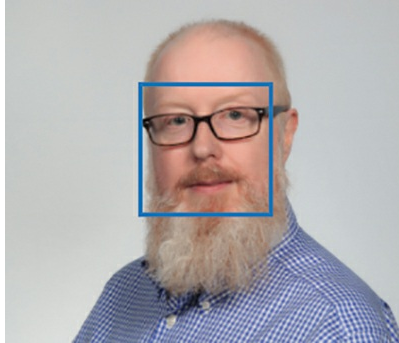


FIGURE 3-7 Example of facial detection

The facial detection identified the face, drew a box around the face, and supplied details such as wearing glasses, neutral emotion, not smiling, and other facial characteristics.

Customer engagement in retail is an example of using facial recognition to identify customers when they walk into a retail store.

Validating identity for access to business premises is an example of facial detection and recognition. Facial detection and recognition can identify a person in an image, and this can be used to permit access to a secure location.

Recognition of famous people is a feature of domain-specific content where thousands of well-known peoples' images have been added to the computer vision model. Images can be tagged with the names of celebrities.

Face detection can be used to monitor a driver's face. The angle, or head pose, can be determined, and this can be used to tell if the driver is looking at the road ahead, looking down at a mobile device, or showing signs of tiredness.

Now that you have learned about the concepts of computer vision, it's time to look at the specific Vision services provided by Azure AI Services.

Skill 3.2: Identify Azure tools and services for computer vision tasks

Azure AI Services provide trained computer vision models that cover most of the capabilities required for analyzing images and videos. This section describes the capabilities of the Vision service in Azure AI Services, which are a major focus of the Microsoft Azure AI Fundamentals certification. You will be required to understand how to use the Vision service and especially how to create your own custom models with the Custom Vision service.

Exam Tip

You will need to be able to distinguish between the Vision, Custom Vision, and Face services.

This skill covers how to:

- [Understand the capabilities of the Vision service](#)
- [Understand the Custom Vision service](#)
- [Understand the Face service](#)

Understand the capabilities of the Vision service

The Vision service in Azure AI Services provides several algorithms to analyze images. For instance, Azure AI Vision can:

- Detect and locate thousands of classes of common objects
- Detect and analyze human faces
- Generate a single sentence description of an image
- Generate a set of tags that relate to the contents of the image
- Identify images that contain adult, racy, or gory content
- Detect and extract the text from an image

To use Vision, you will need to create an AI Services multi-service resource, or a Vision single-service resource, as described earlier in this chapter. The following sections describe the capabilities of the APIs in the Vision service.

Note Service Names

Remember, the Vision service is also known as Computer Vision or Azure AI Vision.

Analyze image

The Analyze operation extracts visual features from the image content.

You can upload an image or, more commonly, specify a URL to where the image is stored. Then, you specify the features that you want to extract. If you do not specify any features, the image categories are returned.

Depending on the version you use (4.0 or 3.2), the available features and format differ slightly.

When you use Vision API v4.0 (latest), formulate the request URL as follows:

```
https://{endpoint}/computervision/imageanalysis:analyze[?features][&model-version]
[&language][&api-version]
```

When you use Vision API v3.2 (legacy), formulate the request URL as follows:

```
https://{endpoint}/vision/v3.2/analyze[?visualFeatures][&details][&language]
```

The URL for the image is contained in the body of the request.

The visual features that you can request include:

- **Read (v4.0 only)** Extracts readable text from images with Optical Character Recognition (OCR).
- **People Detection (v4.0 only)** Detects the people appearing within an image.
- **Objects (v4.0 and v3.2)** Detects various objects within an image, including their coordinates.
- **Tags (v4.0 and v3.2)** Tags the image with a detailed list of words related to the content.
- **Smart Crop (v4.0 and v3.2)** Analyzes the image to determine the most visually significant area, returning bounding box coordinates that match a given aspect ratio. For this operation, the SDK uses the term “Crop Suggestions.”
- **Captions (v4.0 only)** Describes the image content with a complete sentence.
- **Adult (v3.2 only)** Detects if the image is pornographic (adult), contains sexually suggestive content (racy), or depicts violence or blood (gory).
- **Brands (v3.2 only)** Detects well-known brands within an image.
- **Categories (v3.2 only)** Categorizes image content according to a taxonomy of 86 categories.
- **Color (v3.2 only)** Determines the accent color, dominant color, and whether an image is black and white.
- **Faces (v3.2 only)** Detects if there are human faces in the image with their coordinates, gender, and age.
- **ImageType (v3.2 only)** Detects if the image is clipart or a line drawing.

The `features` parameter (`visualFeatures` in v.3.2) can have multiple values, separated by commas.

The `details` parameter in v3.2 is used to extract domain-specific details:

- **Celebrities** Identifies celebrities in the image
- **Landmarks** Identifies landmarks in the image

The `language` parameter supports a few languages. The default is `en`, English. Currently, English is the only supported language for tagging and categorizing images.

If you’re using version 4.0, the Vision service supports a wider range of formats and larger file sizes. Specifically, your image must be in one of the following formats: JPEG, PNG, GIF, BMP, WEBP, ICO, TIFF, or MPO. The file size must be less than 20 megabytes, and the image dimensions must fall within a range of greater than 50 x 50 pixels but less than 16,000 x 16,000 pixels.

On the other hand, if you’re working with version 3.2, the supported formats are more limited: only JPEG, PNG, GIF, or BMP are accepted. Additionally, the file size must not exceed 4 megabytes, and the image must be at least 50 x 50 pixels in size.

Below is the JSON returned with version 3.2 for the image of the three cats used earlier in this chapter for these categories: `adult`, `color`, and `imageType` features.

```
{
  "adult": { "isAdultContent": false, "isRacyContent": false, "isGoryContent": false,
    "adultScore": 0.0034875324927270412, "racyScore": 0.007236192934215069, "goreScore":
    0.001163262757472694 },
  "color": { "dominantColorForeground": "Black", "dominantColorBackground": "Grey",
    "dominantColors": [ "Grey", "Black", "White" ],
    "accentColor": "666666", "isBwImg": true, "isBWImg": true },
  "imageType": { "clipArtType": 0, "lineDrawingType": 0 }
}
```

The category has been correctly identified with a confidence of 79.2%. There is no adult content in the image. The main colors are black, white, and grey.

The Analyze operation provides a generic image analysis returning many different visual features. There are other operations that extract other information from the image or provide more detail than that provided by the Analyze operation.

Captions (Description)

Image Analysis in version 4.0 offers two caption functionalities:

- **Caption** Generates a single sentence describing the entire image
- **Dense Captions** Offers more data by generating sentences describing up to 10 different regions with coordinates, in addition to the overall description

This functionality is available in English only.

You formulate the request URLs for Caption and Dense Captions as follows:

```
https://{endpoint}/computervision/imageanalysis:analyze?api-version=2024-02-01&features=caption
https://{endpoint}/computervision/imageanalysis:analyze?api-version=2024-02-01&features=densecaptions
```

The following JSON shows the dense captions for the image of the three cats mentioned before.

```
{
  "denseCaptionsResult": {
    "values": [
      {
        "text": "a group of cats sitting on a counter",
        "confidence": 0.9143637418746948,
        "boundingBox": {
          "x": 0, "y": 0, "w": 1148, "h": 842}
      },
      {
        "text": "a black and white cat",
        "confidence": 0.9195271134376526,
        "boundingBox": {
          "x": 830, "y": 181, "w": 282, "h": 416}
      },
      {
        "text": "a cat sitting on a table",
        "confidence": 0.8038703203201294,
        "boundingBox": {
          "x": 7, "y": 267, "w": 299, "h": 314}
      },
      {
        "text": "a close up of a white mug",
        "confidence": 0.7983858585357666,
        "boundingBox": {
          "x": 364, "y": 412, "w": 73, "h": 58}
      },
      {
        "text": "a cat cuddling with another cat",
        "confidence": 0.7859628200531006,
        "boundingBox": {
          "x": 519, "y": 359, "w": 356, "h": 206}
      },
      {
        "text": "a close-up of a handle",
        "confidence": 0.7525227665901184,
        "boundingBox": {
          "x": 189, "y": 0, "w": 927, "h": 149}
      }
    ]
  },
}
```

There are multiple captions with the region coordinates and a single sentence describing the image with a confidence of 91.4%.

As do the previous features, the Describe Image feature in version 3.2 supplies a general description and a list of tags, but without any coordinates. You formulate the request URL as follows:

```
https://{endpoint}/vision/v3.2/analyze?visualFeatures=Description[?maxCandidates]
[&language]
https://{endpoint}/vision/v3.2/describe[?maxCandidates][&language]
```

The first URL uses the more flexible /analyze endpoint, allowing you to specify multiple visual features (such as Description, Tags, Objects, and more) in a single request. In contrast, the second URL uses the dedicated /describe endpoint, which is limited to image description only and cannot be combined with other analysis types in the same call.

The parameter maxCandidates specifies the number of descriptions to return. The default is 1. The default language is English.

Below is the JSON result for the same image.

```
{
  "description": {
    "tags": ["cat", "sitting", "white", "wall", "black", "indoor", "sink"],
    "captions": [{"text": "a group of cats sitting on a counter top", "confidence": 0.5289103388786316}]}
}
```

The confidence level of the image description is 52.9%, which is less compared with the latest version analyzed above.

Detect objects

The Detect operation detects objects in an image and provides coordinates for each object detected.

You formulate the request URL for version 4.0 as follows:

```
https://{endpoint}/computervision/imageanalysis:analyze?api-version=2024-02-01&features=objects
```

Following is the JSON returned:

```
{
  "objectsResult": {
    "values": [
      {
        "boundingBox": {
          "x": 826, "y": 191, "w": 287, "h": 419},
        "tags": [{"name": "cat", "confidence": 0.846}]}
    ]
  }
}
```

```

    {"boundingBox": {"x": 24, "y": 275, "w": 299, "h": 326}, "tags": [{"name":
"cat", "confidence": 0.822}}],
    {"boundingBox": {"x": 529, "y": 355, "w": 274, "h": 223}, "tags": [{"name":
"cat", "confidence": 0.783}}]
  ]
}

```

The Detect operation identified three cats with a high level of confidence and provided the coordinates for each cat.

The corresponding API calls for version 3.2 are the following:

```

https://{endpoint}/vision/v3.2/analyze?visualFeatures=Objects
https://{endpoint}/vision/v3.2/detect

```

The first URL uses the more flexible `/analyze` endpoint, allowing you to specify multiple visual features (such as `Description`, `Tags`, `Objects`, and more) in a single request. In contrast, the second URL uses the dedicated `/detect` endpoint, which is limited to object detection only and cannot be combined with other analysis types in the same call.

Content tags

The Tags operation generates a list of tags, based on the image and the objects in the image. Tags are based on objects, people, and animals in the image, along with the placing of the scene (setting) in the image.

The tags are provided as a simple list with confidence levels.

You formulate the request URL for version 4.0 as follows:

```

https://{endpoint}/computervision/imageanalysis:analyze?api-version=2024-02-01&features=tags

```

Following is the JSON returned:

```

{
  tagsResult: {
    "values": [
      {"name": "indoor", "confidence": 0.9943850040435791},
      {"name": "cat", "confidence": 0.9880032539367676},
      {"name": "animal", "confidence": 0.9813541173934937},
      {"name": "wall", "confidence": 0.9745256900787354},
      {"name": "black and white", "confidence": 0.9260863065719604},
      {"name": "home appliance", "confidence": 0.9126243591308594},
      {"name": "small to medium-sized cats", "confidence": 0.9064431190490723},
      {"name": "felidae", "confidence": 0.8941723704338074},
      {"name": "mammal", "confidence": 0.8829941749572754},
      {"name": "white", "confidence": 0.7592796087265015},
      {"name": "sitting", "confidence": 0.750069260597229},
      {"name": "black", "confidence": 0.7427221536636353},
      {"name": "appliance", "confidence": 0.7241742610931396},
      {"name": "kitchen", "confidence": 0.5594900250434875}
    ]
  }
}

```

The Tags operation generated a list of tags in order of confidence. The indoor tag has the highest confidence score of 99.4%, with kitchen the lowest score of 55.9%.

The corresponding API call for version 3.2 is the following:

```

https://{endpoint}/vision/v3.2/analyze?visualFeatures=Tags[?language]
https://{endpoint}/vision/v3.2/tag[?language]

```

The first URL uses the more flexible `/analyze` endpoint, allowing you to specify multiple visual features (such as `Description`, `Tags`, `Objects`, and more) in a single request. In contrast, the second URL uses the dedicated `/tag` endpoint, which is limited to image tagging only and cannot be combined with other analysis types in the same call.

Thumbnail generation

The Get Thumbnail operation generates a thumbnail image by analyzing the image, identifying the area of interest, and smart cropping the image.

You formulate the request URL for version 4.0 as follows:

```

https://{endpoint}/computervision/imageanalysis:analyze?api-version=2024-02-01&features=smartCrops[&smartcrops-aspect-ratios]

```

An optional parameter, `smartcrops-aspect-ratios` lets you specify one or more desired crop shapes (width divided by height). If not specified, the service chooses one automatically (between 0.5 and 2.0).

Following is the JSON returned:

```

{
  "smartCropsResult": {
    "values": [
      {"aspectRatio": 1.5, "boundingBox": {
        "x": 48, "y":

```

```
35, "w": 1052,
    "h": 702}}}]}}
```

This reduced the image size by a little throughout the region of interest, adjusting it from a width of 1148 and height of 842 to a cropped width of 1052 and height of 702.

Alternatively, you formulate the request URL for version 3.2 as follows:

```
https://{endpoint}/vision/v3.2/generateThumbnail[?width][&height][&smartCropping]
```

The generated thumbnail will differ depending on the parameters you specify for `height`, `width`, and `smartCropping`.

Both `width` and `height` are numeric values, and `smartCropping` is either 0 or 1.

But in this case, the response contains a binary JPG image.

Domain-specific content

Additionally, version 3.2 offers two models in Vision that have been trained on specific sets of images:

- **Celebrity** Recognizes famous people
- **Landmark** Recognizes famous buildings or outdoor scenery

You formulate the request URL as follows:

```
https://{endpoint}/vision/v3.2/models/{model}/analyze[?language]
```

The model is either celebrities or landmarks. English is the default language.

[Figure 3-8](#) is a photograph of the Belém tower in Lisbon, Portugal. This is a famous sixteenth-century landmark, the place from where explorers set sail.



FIGURE 3-8 Example of a landmark

The JSON returned includes the name of the celebrity or landmark:

```
"landmarks": [{"name": "Belém Tower", "confidence": 0.9996672868728638 }]
```

These domain-specific models can also be used by the Analyze operation by using the `details` parameter.

The Analyze operation can also detect commercial brands from images using a database of thousands of company and product logos.

Optical character recognition (OCR)

OCR is the extraction of printed or handwritten text from images. You can extract text from images and documents. Microsoft provides two main OCR methods: Vision (optimized for images) and Document Intelligence (optimized for text-heavy documents).

Vision (Image Analysis v4.0) is ideal for quick text extraction from “in-the-wild” images, such as labels, signs, or posters. Its key features are:

- **Fast, synchronous API** Real-time results for user-facing apps
- **Wide language support** Covers many global languages, including mixed handwritten and printed text
- **Unified approach** Extract text plus other image insights (such as object detection) in one call

You formulate the request URL as follows:

```
https://{endpoint}/computervision/imageanalysis:analyze?
api-version=2024-02-01&features=read
```

[Figure 3-9](#) is an image containing a quote from the Greek philosopher Democritus.



FIGURE 3-9 Quote printed in an image

The JSON returned includes the pieces of text from the image:

```
{
  "readResult": {
    "blocks": [
      {
        "lines": [
          {
            "text": "NOTHING",
            "boundingPolygon": [{"x": 26, "y": 27}, {"x": 374, "y": 27}, {"x": 374, "y": 83}, {"x": 26, "y": 83}],
            "words": [{"text": "NOTHING", "boundingPolygon": [{"x": 29, "y": 27}, {"x": 352, "y": 30}, {"x": 350, "y": 83}, {"x": 27, "y": 84}], "confidence": 0.993}],
          },
          {
            "text": "EXISTS",
            "boundingPolygon": [{"x": 26, "y": 124}, {"x": 376, "y": 124}, {"x": 376, "y": 209}, {"x": 26, "y": 209}],
            "words": [{"text": "EXISTS", "boundingPolygon": [{"x": 26, "y": 124}, {"x": 360, "y": 124}, {"x": 360, "y": 213}, {"x": 28, "y": 211}], "confidence": 0.994}],
          },
          {
            "text": "EXCEPT",
            "boundingPolygon": [{"x": 28, "y": 254}, {"x": 380, "y": 254}, {"x": 380, "y": 331}, {"x": 28, "y": 330}],
            "words": [{"text": "EXCEPT", "boundingPolygon": [{"x": 28, "y": 255}, {"x": 367, "y": 332}, {"x": 31, "y": 331}], "confidence": 0.994}],
          },
          {
            "text": "ATOMS",
            "boundingPolygon": [{"x": 29, "y": 376}, {"x": 374, "y": 376}, {"x": 374, "y": 451}, {"x": 29, "y": 451}],
            "words": [{"text": "ATOMS", "boundingPolygon": [{"x": 34, "y": 377}, {"x": 363, "y": 377}, {"x": 363, "y": 452}, {"x": 36, "y": 453}], "confidence": 0.997}],
          },
          {
            "text": "AND EMPTY",
            "boundingPolygon": [{"x": 35, "y": 481}, {"x": 362, "y": 481}, {"x": 362, "y": 529}, {"x": 35, "y": 529}],
            "words": [{"text": "AND", "boundingPolygon": [{"x": 40, "y": 482}, {"x": 147, "y": 482}, {"x": 146, "y": 530}, {"x": 39, "y": 529}], "confidence": 0.999}],
          },
          {
            "text": "EMPTY",
            "boundingPolygon": [{"x": 176, "y": 482}, {"x": 352, "y": 482}, {"x": 353, "y": 530}, {"x": 176, "y": 530}],
            "confidence": 0.998}],
          },
          {
            "text": "SPACE.",
            "boundingPolygon": [{"x": 41, "y": 561}, {"x": 360, "y": 561}, {"x": 360, "y": 638}, {"x": 41, "y": 638}],
            "words": [{"text": "SPACE.", "boundingPolygon": [{"x": 42, "y": 561}, {"x": 355, "y": 562}, {"x": 357, "y": 640}, {"x": 42, "y": 640}], "confidence": 0.99}],
          },
          {
            "text": "Everything else",
            "boundingPolygon": [{"x": 27, "y": 667}, {"x": 379, "y": 667}, {"x": 379, "y": 718}, {"x": 27, "y": 718}],
            "words": [{"text": "Everything", "boundingPolygon": [{"x": 29, "y": 669}, {"x": 268, "y": 667}, {"x": 268, "y": 719}, {"x": 28, "y": 718}], "confidence": 0.993}],
          },
          {
            "text": "else",
            "boundingPolygon": [{"x": 289, "y": 667}, {"x": 376, "y": 667}, {"x": 377, "y": 719}, {"x": 289, "y": 719}],
            "confidence": 0.993}],
          },
          {
            "text": "is opinion.",
            "boundingPolygon": [{"x": 134, "y": 723}, {"x": 382, "y": 723}, {"x": 382, "y": 777}, {"x": 134, "y": 777}],
            "words": [{"text": "is", "boundingPolygon": [{"x": 135, "y": 723}, {"x": 172, "y": 724}, {"x": 172, "y": 778}, {"x": 135, "y": 778}], "confidence": 0.993}],
          },
          {
            "text": "opinion.",
            "boundingPolygon": [{"x": 183, "y": 724}, {"x": 381, "y": 724}, {"x": 381, "y": 776}, {"x": 183, "y": 776}],
            "confidence": 0.994}]]]]}}}
```

The response includes pieces of text that are split into individual lines. Each line has a bounding polygon showing where it is placed on the image. In each line, there are words, each with its own bounding box and a numeric confidence score.

The corresponding API call for version 3.2 is the following:

```
https://{endpoint}/vision/v3.2/ocr
```

OCR for Documents with Document Intelligence

The Document Intelligence (Read model) service is targeted primarily at documents that contain a lot of text, for example, PDFs, Microsoft Word documents, or multipage TIFF services. This service falls outside the umbrella of Vision.

The prebuilt-read model is still the high-resolution OCR engine that is specifically tuned for dense documents, including PDFs, Microsoft Word documents, and multipage TIFFs.

It can detect paragraphs, lines, individual words, and bounding boxes, as well as process handwriting.

Some of the key features are:

- Support for PDFs, Microsoft Office files, HTML, and images
- High-resolution OCR (and code the layout orientation) with layout detection (paragraphs, text lines, bounding boxes)
- Can handle large and multipage documents asynchronously (up to 2,000 pages)
- Translates PDFs into searchable documents

Note Service Rebranding

In July 2023, Microsoft rebranded Azure AI Form Recognizer as Azure AI Document Intelligence.

Suppose you need to analyze the example image in [Figure 3-10](#) using Document Intelligence.



FIGURE 3-10 Document Intelligence

First, create the associated request URL as follows, where `output` specifies the document format and `api-version` assigns the model version:

```
https://{endpoint}/documentintelligence/documentModels/prebuilt-read:analyze[?output] [&api-version]
```

Because this is an asynchronous operation, you need to retrieve the output using the following request:

```
https://{endpoint}/documentintelligence/documentModels/prebuilt-read/analyzeResults/ {resultId}[?api-version]
```

The output will be similar to but longer than the example that follows. Due to the extreme level of detail the request produces, the output snippet here shows only the first instance of each component (words, lines, paragraphs, and so on).

```
{  
  "status": "succeeded",  
  "createdDateTime": "2025-06-20T18:53:34Z",  
  "lastUpdatedDateTime": "2025-06-20T18:53:35Z",  
  "analyzeResult": {  
    "apiVersion": "2024-11-30",  
    "modelId": "prebuilt-read",  
    "stringIndexType": "textElements",  
  
    "content": "contoso-receipt.png 1688 x 3000\nLayers V\nContoso\nContoso\n123 Main  
Street\n"
```

```

nRedmond 4WA 98052\n123-456-7890\n6/10/2019 13:59\nSales Associate: Paul\
n1 Surface Pro 4\
6\n256GB | Intel Core i5\n8GB RAM (Black)\n999.00\n1 SurfacePen\n$ 99.99\
nSub-Total\n$ 1098.99\nTax\n104.40\nTotal\n1203.39",

"pages": [{
  "pageNumber": 1,
  "angle": 0.29,
  "width": 1466,
  "height": 1295,
  "unit": "pixel",

  // First word example
  "words": [{
    "content": "Al-900",
    "polygon": [939, 14, 984, 14, 984, 31, 939, 30],
    "confidence": 0.848,
    "span": { "offset": 0, "length": 6 }
  }],
  // First line example
  "lines": [{
    "content": "Al-900 Demo - [ contoso-receipt.png ]",
    "polygon": [939, 14, 1192, 14, 1192, 32, 939, 31],
    "spans": [{ "offset": 0, "length": 37 }]},
    "spans": [{ "offset": 0, "length": 335 }]}],
  // First paragraph example
  "paragraphs": [{
    "content": "Al-900 Demo - [ contoso-receipt.png ]",
    "spans": [{ "offset": 0, "length": 37 }],
    "boundingRegions": [{
      "pageNumber": 1,
      "polygon": [939, 14, 1192, 14, 1192, 32, 939, 32]}]}],
  "styles": [],
  // Format used for indexing text content
  "contentFormat": "text"}]

```

The `status` field indicates whether the analysis was successful. The `createdDateTime` and `lastUpdatedDateTime` fields show when the request started and when it finished, respectively.

Inside `analyzeResult`, `content` gives you the complete extracted text, which is useful when you just need everything in a plain text format.

If you need more detail, `pages` has `words`, `lines`, and `paragraphs` with coordinates and the confidence score for each page.

The `modelId` shows which model was used (in this case it was `prebuilt-read`, meaning general OCR).

If `styles` is empty, this means no bold or handwritten text formatting was detected.

Use `content` if you just need text quickly; use `pages` if you need structure and key coordinates.

Vision vs. Document Intelligence

There are two main services that perform text extraction from images and documents:

- Vision
- Document Intelligence

You should understand the differences between these services.

Vision is optimized for quick text extraction from everyday images like photos, signs, and labels. Vision works synchronously and gives you real-time results. Vision is ideal when you need fast basic text recognition from simple images.

Document Intelligence is tuned for text-heavy content like PDFs, Microsoft Word documents, and complex multi-page files. Document Intelligence runs at higher resolution and can handle large documents asynchronously. Document Intelligence recognizes document structure including paragraphs, tables, and form fields, not just plain text extraction.

Use Vision for quick image text extraction. Use Document Intelligence for comprehensive document analysis and structured data extraction.

Understand the Custom Vision service

The Custom Vision service is an alternative to the pretrained Vision service. Custom Vision enables you to build, train, and deploy a custom image recognition model based on images you provide.

In Custom Vision, you define the labels for your model and a set of sample images. You tag your images with your labels. The Custom Vision service uses a machine learning algorithm to analyze these sample images. Custom Vision trains and evaluates the custom model.

You can then deploy your model with an endpoint and key and consume this model in your apps in a similar way to the Vision service.

Custom Vision supports two different types of modes:

- **Image classification** Tags an image using the labels defined for the model.
- **Object detection** Identifies objects using the tags and provides the coordinates of objects in an image. Object detection is a type of classification model.

A model can be built for only one of these two types.

Custom Vision uses a web portal (www.customvision.ai) where you can create your model, upload your images, label the images or the objects, train the model, test and evaluate the model, and finally deploy the model.

To use Custom Vision, you will need to create either an AI Services multi-service resource, or a Custom Vision service resource, as described earlier in

this chapter. There are two Custom Vision services: Training and Prediction. You will require both services.

Creating a Custom Vision model

The process for creating a Custom Vision model is as follows:

1. Specify the model type.
2. Upload own images.
3. Define your labels.
4. Label the images, or identify the object in the images.
5. Train the model.
6. Evaluate the model.
7. Deploy the model.

Custom Vision exercise

The following steps take you through creating a custom object detection model to identify fruit from images.

You can download the fruits dataset aka.ms/fruit-objects. Extract the image files. [Figure 3-11](#) shows all 33 images.

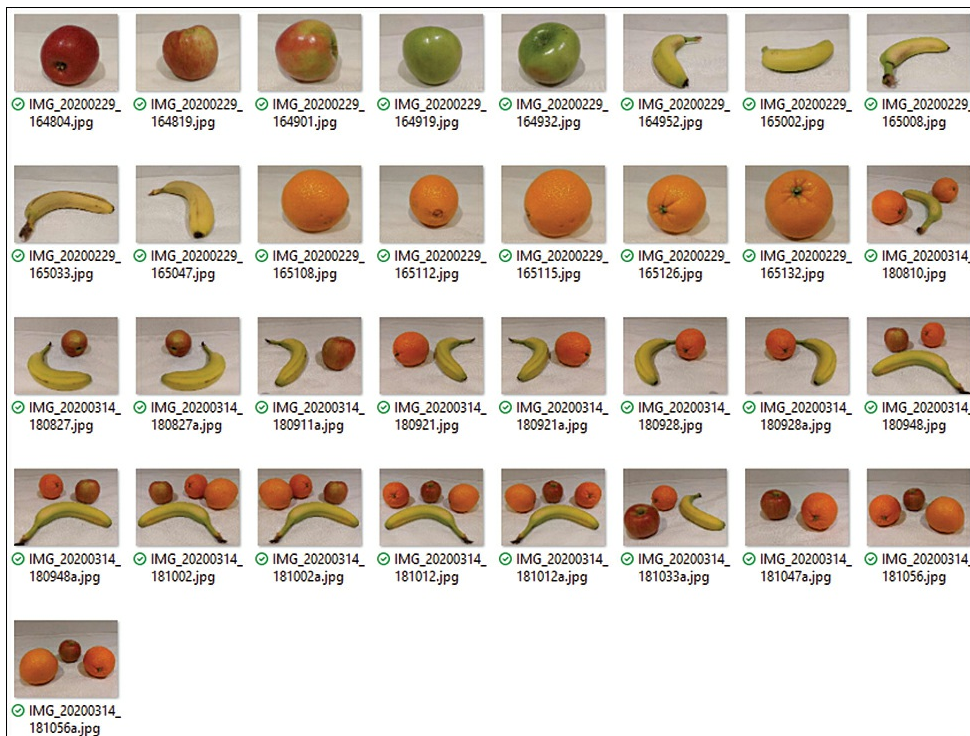


FIGURE 3-11 Images of fruit

You will need to use 30 of the images to train your model; set aside three images for testing your model after you have trained it.

First, you need to create a Custom Vision service. [Figure 3-12](#) shows the pane in the Azure portal for creating a Custom Vision service. For Choose Options, select Both because you need training and prediction services. Next, select the subscription and resource group, then give the service a unique name. This name will be the domain name for your endpoint and must be unique worldwide.

Create Custom Vision ...

Basics Network Tags Review + create

Customize and embed state-of-the-art computer vision for specific domains. Build frictionless customer experiences, optimize manufacturing processes, accelerate digital marketing campaigns -- and more. No machine learning expertise is required.

[Learn more](#)

Create options * Both Prediction Training

Project Details

Subscription * ⓘ

Resource group * ⓘ [Create new](#)

Instance Details

A training resource and a prediction resource will be created in same region.

Region ⓘ

Name * ⓘ ✓

Training Resource

Select pricing for training Resource.

Training pricing tier * ⓘ

[View full pricing details](#)

Prediction Resource

Select pricing for prediction Resource.

Prediction pricing tier * ⓘ

[View full pricing details](#)

FIGURE 3-12 Creating an AI Services resource

In the Training Resource section, select the region where the Training resource is to be deployed, as well as your pricing tier: Free F0 or Standard S0. Next, select the region and pricing tier for the Prediction resource.

Click Review + Create to validate the options, then click Create to create the resource. Because you selected Both, two resources will be deployed: the Training resource with the name you provided and the Prediction resource with “-Prediction” appended to the Training resource’s name.

Next, you need to navigate to the Custom Vision web portal, www.customvision.ai, and sign in with the credentials for your Azure subscription.

You will need to create a new project, name it, and select your Custom Vision training resource (or you can use a multi-service Cognitive Service resource).

Next, you select Object Detection as the Project Type and General for the Domain, as shown in [Figure 3-13](#).

FIGURE 3-13 New Custom Vision project

The domain is used to train the model. You should select the most relevant domain that matches your scenario. If none of the domains are applicable, use the General domain.

The domains for image classification are:

- General
- Food
- Landmarks
- Retail
- General (compact)
- Food (compact)
- Landmarks (compact)
- Retail (compact)
- General [A1]
- General [A2]
- General (compact) [S1]

The domains for object detection are:

- General
- Logo
- Products on Shelves
- General (compact)
- General (compact) [S1]
- General [A1]

Compact domains are lightweight models that are designed to run locally—for example, on mobile platforms.

The General [A1] and General [A2] domains are suitable for a wide range of use cases and are optimized for high accuracy. Among them, the General [A2] model offers faster inference and reduced training time, making it a better choice when speed and efficiency are priorities.

For object detection tasks, the General (compact) domain requires specialized postprocessing logic. If you prefer a model that does not include this postprocessing, opt for General (compact) [S1] instead.

Need More Review? Domains

For more explanation as to which domain to choose, see docs.microsoft.com/azure/cognitive-services/custom-vision-service/select-domain.

After you create the project, you should create your tags. In this exercise, you will create three tags:

- Apple
- Banana
- Orange

Next, upload your training images. [Figure 3-14](#) shows the Custom Vision project with the images uploaded and untagged.

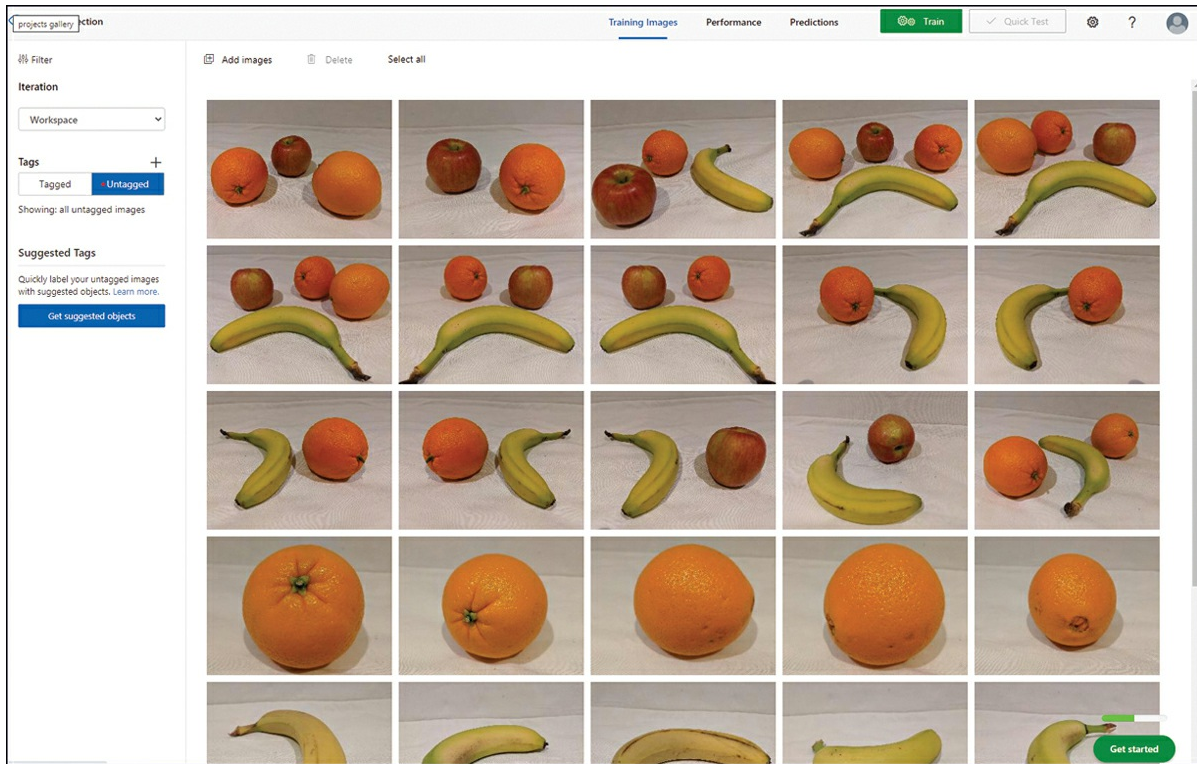


FIGURE 3-14 Custom Vision project with uploaded images

You now need to click each image. Custom Vision will attempt to identify objects and highlight the object with a box. You can adjust and resize the box, and then tag the objects in the image, as shown in [Figure 3-15](#).

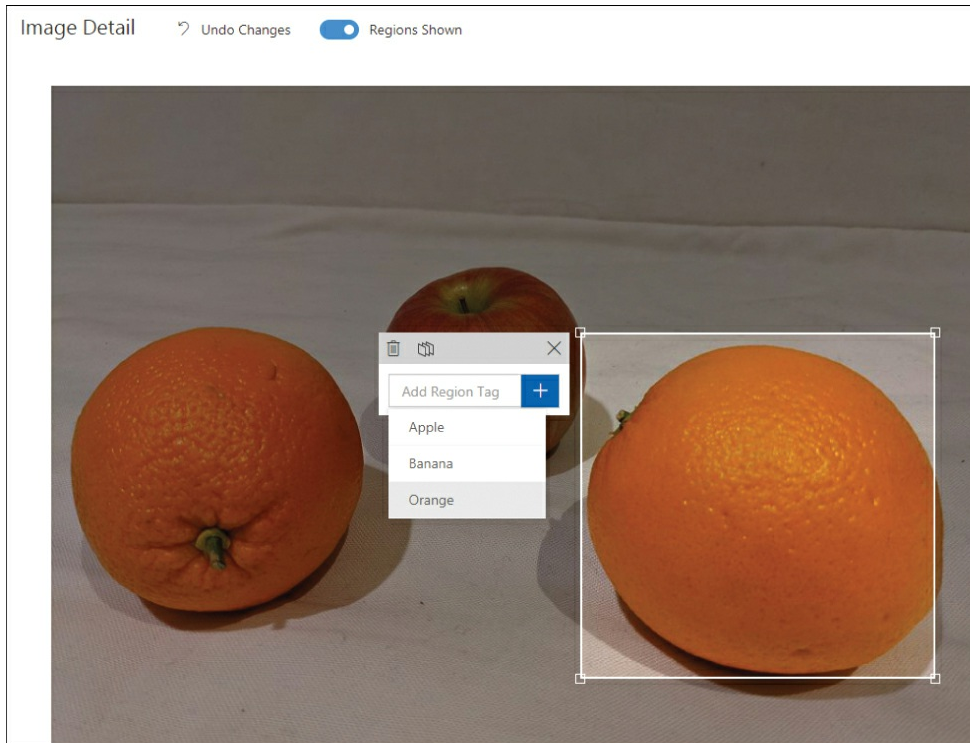


FIGURE 3-15 Tagging objects

Repeat tagging the objects for all the training images.

You will need at least 10 images for each tag, but for better performance, you should have a minimum of 30 images. To train your model, use a variety of images with different lighting, orientation, sizes, and backgrounds.

Select the Tagged button (at left) to see your tagged images.

You are now ready to train your model. Click the Train button at the top of the project window. There are two choices:

- **Quick Training** Training will take a few minutes.
- **Advanced Training** Specifies the amount of time to spend training the model from 1 to 24 hours.

Select the Quick Training option, and click Train.

When training has completed, the model's performance is displayed. There are two key measures that indicate the effectiveness of the model:

- **Precision** The percentage of predictions that the model correctly detected. This is a value between 0 and 1 and is shown as a percentage (the higher the better).
- **Recall** The percentage of the predictions that the model was correct. This is a value between 0 and 1 and is shown as a percentage (the higher the better).

[Figure 3-16](#) shows the results after training the model.

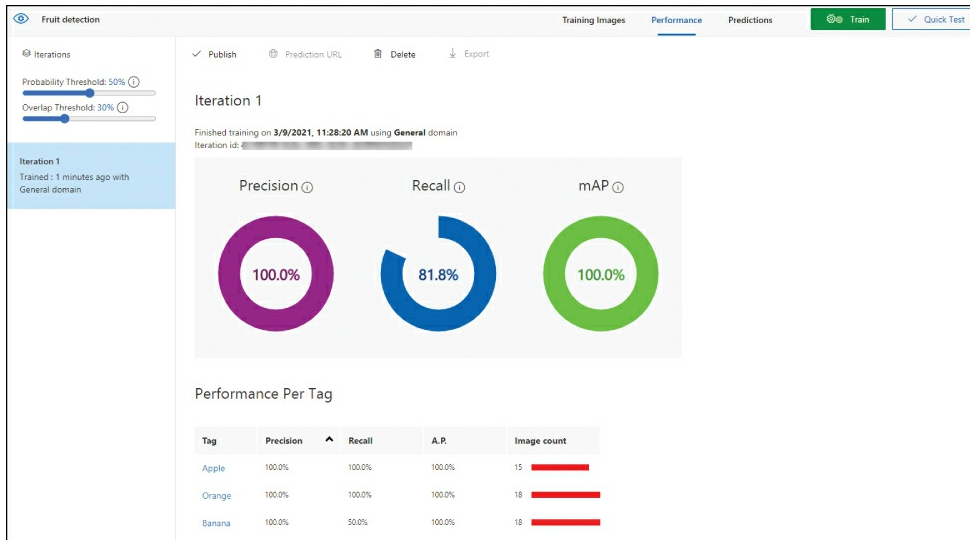


FIGURE 3-16 Model performance

You can use the Quick Test option to check your model. You should upload one of the three images you put aside. The image will be automatically processed, as shown in [Figure 3-17](#).

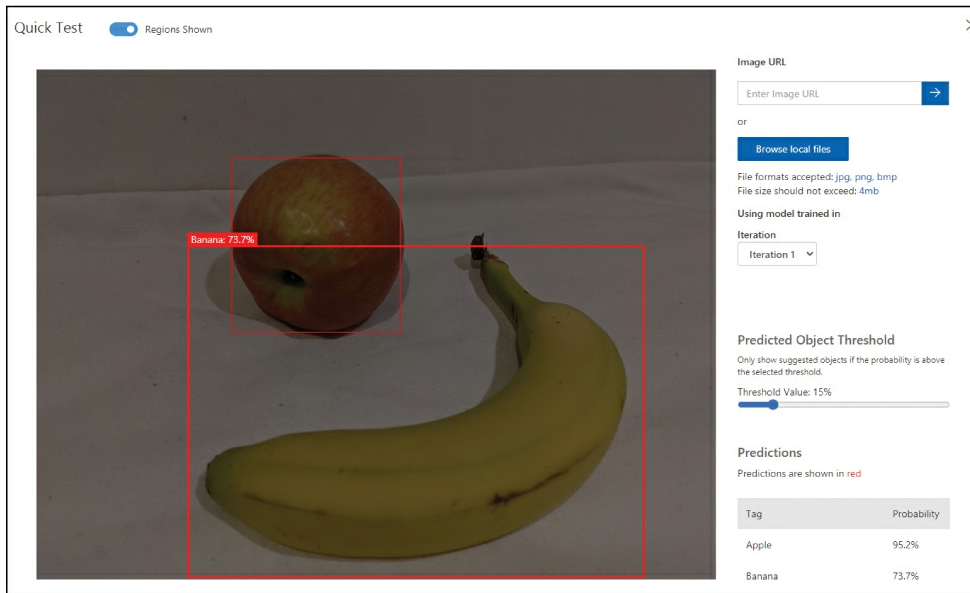


FIGURE 3-17 Quick Test

The model has identified both the apple and the banana and drawn boxes around the pieces of fruit. The objects are tagged, and the results have high confidence scores of 95.2% and 73.7%.

To publish your model, click the Publish button at the top of the Performance tab ([Figure 3-16](#)). You will need to name your model and select a Custom Vision Prediction resource.

Note Published Endpoint

You cannot use a multi-service AI Services resource for the published endpoint.

Publishing will generate an endpoint URL and key so that your applications can use your custom model.

Vision vs. Custom Vision

It is important that you understand the differences of capabilities of the prebuilt Vision service compared with the capabilities of Custom Vision.

Vision uses prebuilt models trained with many thousands of images and supports invoking multiple features in a single request. The service has the following capabilities:

- People Detection
- Objects
- Tags
- Smart Crop
- Captions
- Content Moderation
- Brands
- Categories
- Color
- ImageType
- Faces
- OCR

Custom Vision uses images and tags that you supply to train a custom image recognition model. Custom Vision has only two capabilities:

- Object detection
- Image classification

Understand the Face service

While the Vision service includes face detection, it provides only basic information about the person. The Face service performs more detailed analysis of the faces in an image. The Face service can examine facial characteristics, compare faces, and even verify a person's identity. If you want to do analysis around the characteristics of faces or compare faces, you should use the Face service instead of Vision.

Facial recognition has many use cases, such as security, retail, aiding visually challenged people, disease diagnosis, school attendance, and safety.

The Face service contains several advanced face algorithms, enabling face attribute detection and recognition. The Face service examines facial landmarks including noses, eyes, and lips to detect and recognize faces.

The Face service is only available to Microsoft managed customers and partners. To request access, compile the form at aka.ms/facerecognition.

To prevent stereotyping, discrimination, or unfair treatment, Microsoft has retired the use of facial attribute inference features like emotion and gender detection. The attributes that remain available for predictions (age, smile, facial hair, hair, makeup) are limited in capabilities.

The Face service can detect attributes of the face, such as the following:

- Smile
- Age
- Facial hair
- Hair
- Makeup

The Face service can perform facial recognition:

- Similarity matching
- Identity verification

Another interesting capability is Liveness detection, which can determine if an input video is real or spoof.

You can deploy the Face service in the Azure portal by searching for Face when creating a new resource. You must select your region and resource group, provide a unique name, and select the pricing tier: Free F0 or Standard S0.

You can create Face resources using the CLI as follows:

```
az cognitiveservices account create --name <unique name> --resource-group <resource
group name>
--kind Face --sku F0 --location <region>
```

Detection

The Face service detects the human faces in an image and returns their boxed coordinates. Face detection extracts face-related attributes, such as head pose, emotion, hair, and glasses.

The Face service examines 27 facial landmarks, as shown in [Figure 3-18](#). The location of eyebrows, eyes, pupils, nose, mouth, and lips are the facial landmarks used by the Face service.

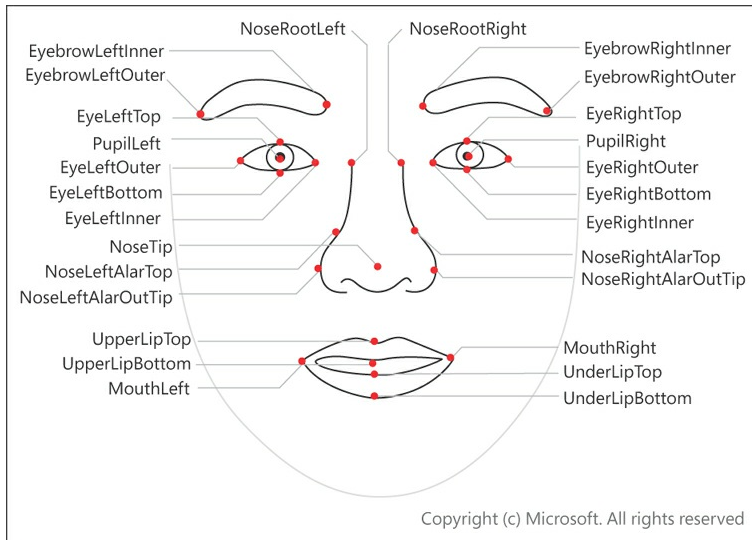


FIGURE 3-18 Facial landmarks

Facial detection provides a set of features, or attributes, about the faces it has detected:

- **Age** The estimated age in years.
- **Glasses** Whether the given face has eyeglasses and the type of eye covering (`NoGlasses`, `ReadingGlasses`, `Sunglasses`, or `Swimming Goggles`).
- **Hair** Whether the face has hair (and its color) or is bald.
- **Facial hair** Whether the face has facial hair.
- **Makeup** Whether the eyes and/or lips have makeup as either true or false.
- **Smile** Whether the face is smiling. A value of 0 means no smile and a value of 1 is a clear smile.
- **Occlusion** Whether there are objects blocking parts of the face. True or false is returned for `eyeOccluded`, `foreheadOccluded`, and `mouthOccluded`.
- **Blur** How blurred the face is in the image. This has a value between 0 and 1 with an informal rating of low, medium, or high.
- **Exposure** The level exposure of the face between 0 and 1 with an informal rating of `underExposure`, `goodExposure`, or `overExposure`.
- **Noise** The level of visual noise detected in the face image. This has a value between 0 and 1 with an informal rating of low, medium, or high.
- **Head pose** The orientation of the face. This attribute is described by the pitch, roll, and yaw angles in degrees.

The request URL is formulated as follows:

```
https://{endpoint}/face/v1.0/detect[?returnFaceId][&returnFaceLandmarks]
[&returnFaceAttributes] &
[&returnRecognitionModel][&recognitionModel][&detectionModel][&faceIdTimeToLive]
```

The parameters you can specify include the following:

- **returnFaceId** True or false to indicate if the API should return IDs of detected faces.
- **returnFaceLandmarks** True or false to indicate if the API should return facial landmarks.
- **returnFaceAttributes** A comma-separated list of the attributes you want returned (`age`, `gender`, `headPose`, `smile`, `facialHair`, `glasses`, `emotion`, `hair`, `makeup`, `occlusion`, `accessories`, `blur`, `exposure`, and `noise`).
- **returnRecognitionModel** When true, the API response includes the recognition model version used to process the request. False is the default.
- **recognitionModel** You should use the `recognitionModel` if you want to use the Recognition operations described in the next section. There are three recognition models you can use: `recognition_01`, `recognition_02`, and `recognition_03`. The default model is `recognition_01`. The latest model, `recognition_03`, is recommended because its accuracy is higher than the older models.
- **detectionModel** There are three detection models you can use: `detection_01`, `detection_02`, and `detection_03`. The default is `detection_01`. The `detection_02` model should be used for images with small, side, and blurry faces. The `detection_03` model has better results on small faces. Facial attributes are not available for `detection_02` and `detection_03`.
- **faceIdTimeToLive** Specifies how long (in seconds) the `faceId` should be stored for use in further operations like verification or identification.

The detection model returns a `faceId` for each face it detects. This ID can then be used by the face recognition operations described in the next section.

The JSON returned using the detect operation on the image of the author in Figure 3-7 follows:

```
{  "faceId": "aa2c934e-c0f9-42cd-8024-33ee14ae05af",
  "faceRectangle": {  "top": 613,  "left": 458,  "width": 442,  "height": 442  },
  "faceAttributes": {  "hair": {  "bald": 0.79,  "invisible": false,  "hairColor":
[  {  "color": "gray",  "confidence": 0.98  },  {  "color": "brown",  "confidence": 0.7  },
{  "color": "blond",  "confidence": 0.7  }
]
```

```
  "confidence": 0.47  }, { "color": "black", "confidence": 0.45  },
{ "color": "other", ↵
  "confidence": 0.28  }, { "color": "red", "confidence": 0.04  },
{ "color": "white", ↵
  "confidence": 0.0  } ]  },

"smile": 0.011,

"headPose": { "pitch": 2.9, "roll": -2.2, "yaw": -9.3  },

"age": 53.0,

"facialHair": { "moustache": 0.9, "beard": 0.9, "sideburns": 0.9  },

"glasses": "ReadingGlasses",

"makeup": { "eyeMakeup": false, "lipMakeup": false  },
```

As you can see, the attributes are mainly correct except for the hair color. This is expected as the image in [Figure 3-7](#) was a professionally taken photograph with good exposure and a neutral expression.

Liveness

The Face service includes a liveness detection capability to determine whether a face in a video stream is real (live) or fake (spoof). It is a critical component of biometric authentication systems for thwarting spoofing attacks using photos, videos, masks, or other forms of spoofing.

Liveness detection ensures that the system is interacting with a physically present subject during authentication. It is particularly important in such applications as digital banking, remote ID verification, and secure access control.

Recognition

The Face service can recognize known faces. Recognition can compare two faces to determine if they are similar (similarity matching) or belong to the same person (identity verification).

There are four operations available in facial recognition:

- **Verify** Evaluates whether two faces belong to the same person. The Verify operation takes two detected faces and determines whether the faces belong to the same person. This operation is used in security scenarios.
- **Identify** Matches faces to known people in a database. The Identify operation takes one or more faces and returns a list of possible matches with a confidence score between 0 and 1. This operation is used for automatic image tagging in photo management software.
- **Find Similar** Extracts faces that look like a person's face. The Find Similar operation takes a detected face and returns a subset of faces that look similar from a list of faces you supply. This operation is used when searching for a face in a set of images.
- **Group** Divides a set of faces based on similarities. The Group operation separates a list of faces into smaller groups on the similarities of the faces.

You should not use the Identify or Group operations to evaluate whether two faces belong to the same person. You should use the Verify operation instead.

Exam Tip

Ensure that you can determine the scenario for each of the four facial recognition operations.

Video Indexer service

Video Indexer is part of Azure AI Services and built on such services as Face, Translator, Vision, and Speech. Video Indexer uses AI to generate insights about stored videos to analyze video and audio content without requiring machine learning knowledge.

Video Indexer has many use cases, such as content management, compliance monitoring, accessibility, security, and media search.

Video Indexer can generate insights from video content with over 30 AI models (including video and audio content analysis).

The Video Indexer service can work on the following tasks:

- Speech-to-text
- Translation
- Face detection and identification
- Brand detection
- OCR
- Keywords and topics
- Scene segmentation
- Emotion detection
- Sentiment analysis

Every insight contains a timestamp providing you a way to skip straight to what happened in the video at that moment.

Video Indexer can be used in two ways: directly from the Video Indexer portal, or created as an Azure resource (as shown in [Figure 3-19](#)).

Home > Azure AI Video Indexer >

Create a Video Indexer resource

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group * [Create new](#)

Resource name *

Region *

Existing content

Storage account * [Create a new storage account](#)

Advanced AI features

Unlock generative AI: Connect your Azure Video Indexer resource to Azure OpenAI for more capabilities. [Learn about the benefits of connecting to Azure OpenAI](#)

Azure OpenAI Resource

Managed identities

The managed identity will be used by the Video Indexer account to access Azure resources.

System-assigned

User-assigned

I have all the rights to use the content/file, and agree that it will be handled per the [Online Services Terms](#) and the [Microsoft Privacy Statement](#).

[Review + create](#) [Previous](#) [Next: Networking >](#)

FIGURE 3-19 Deploy Video Indexer

When you create the Video Indexer resource in Azure, you need to select the region, the resource group, a unique name, a connection to Storage V2 account in the same region, and a pricing tier. Video Indexer is billed based on the amount of video processed and the type of analysis used (audio-only, video-only, or audio and video).

Vision vs. Face service vs. Video Indexer

There are three services that perform an element of facial detection:

- Vision
- Face
- Video Indexer

It is important that you understand the differences of the capabilities of the prebuilt Vision service compared with the capabilities of the Face service and the Video Indexer service.

Video Indexer is part of Azure Media Services and uses AI Services, including the Face service, to extract insights from videos. Video Indexer can detect and identify people and brands.

Exam Tip

You will need to be able to distinguish between Vision, Face, and Video Indexer.

Vision can detect faces in images but can provide only basic information about the person from the image of the face, such as the estimated age and gender.

The Face service can detect faces in images and can provide information about the characteristics of the face. The Face service can also perform:

- Facial analysis
- Face identification
- Pose detection

The Video Indexer service can detect faces in video images but can also perform face identification.

Here are some examples of the differences between these services:

- Face can detect the angle a head is posed at. Vision can detect faces but is not able to supply the angle of the head.
- Video Indexer can detect faces but does not return the attributes Face can return.
- The Face service is concerned with the details of faces. The Video Indexer service can detect and identify people and brands but not landmarks.
- Custom Vision allows you to specify the labels for an image. The other services cannot.
- Vision can identify landmarks in an image. The other services cannot.

Chapter summary

In this chapter, you learned some of the general concepts related to computer vision workloads on Azure. You learned about the types of computer vision, and you learned about the services in Azure AI Services related to computer vision. Here are the key concepts from this chapter:

- AI Services are prebuilt machine learning models available through REST APIs.
- AI Services enable applications to see, hear, speak, search, understand, and build intelligence into your applications quickly and easily.
- AI Services can be deployed with either a multi-service resource or single-service resource.
- AI Services can be deployed through the Azure portal or with the CLI.
- You need both the endpoint and key to use an Azure AI Services resource.
- Computer vision analyzes still images and video streams and can detect and classify images.
- Image classification categorizes images based on the content of the image.
- Object detection identifies and tags individual visual features in an image.
- Optical character recognition (OCR) extracts text from an image.
- Facial detection uses the characteristics of a face to provide attributes about the face.
- The Vision service can perform many operations on an image. Analyzing the image can detect objects, describe the image in a single sentence, tag the objects in the image, identify brands and landmarks, extract text, and identify inappropriate content.
- Object detection is a type of classification model.
- The Custom Vision service enables you to build, train, and deploy a custom image recognition model based on images you provide when the prebuilt Vision service does not explain your domain.
- Custom Vision creates a custom image recognition model.
- Custom Vision requires you to upload your images, tag the images, train your model, and evaluate your model.
- Custom Vision can perform image classification or object detection only.
- The Vision service can detect faces in images but only provides basic information about the person and face. The Face service provides more detailed facial analysis.
- The Face service uses facial landmarks to analyze and identify faces.
- The Face service can detect faces and extract attributes about the face.
- The Face service can perform facial recognition.
- Only the Verify operation should be used to identify a person from an image of their face.

Need More Review? Hands-On Labs

For more hands-on experience with Vision, complete labs 1 to 6 at microsoftlearning.github.io/AI-900-AIFundamentals.

Thought experiment

To apply what you have learned in this chapter and demonstrate your knowledge of its topics, try this thought experiment. You can find the answers in the section that follows.

You work for Fabrikam, Inc., a vehicle insurance company. Fabrikam is interested in processing the many images and documents that customers and assessors send to the company using AI.

Fabrikam wants to evaluate how AI Services can improve their document processing time accuracy.

Fabrikam has recently created an app for customers to send in details of incidents and upload photographs of damage to their vehicles. Fabrikam wants the app to assess the level of damage from the photographs taken.

The app requests that customers take a photo of the driver after an incident. The app also requests that customers take several pictures of the scene of an incident, showing any other vehicles involved and the street. Customers are able to upload dashcam videos as evidence for their claims. Customers can also upload scanned images of their claim form, which contains a diagram explaining the incident.

Insurance adjusters have a mobile app with which they can assess and document vehicle damage. Fabrikam wants the app to assess the cost of repairs based on photographs and other information about the vehicle.

Answer the following questions:

1. Assessing the damage to a vehicle from a photograph is an example of which type of computer vision?
2. You need to capture the license plates of the vehicles involved in an incident. Which type of computer vision should you use?
3. You need to confirm that the driver is insured to drive the vehicle. Which type of computer vision should you use?
4. You need to automatically identify the vehicles in the image. Which type of computer vision should you use?
5. Can you use the free tier to create a single resource for all these requirements?
6. You are unable to process some high-quality photographs that customers upload. Can you configure Vision to process these images?
7. Which service should you use to assess the level of damage to a vehicle from a photograph?
8. Which model type should you use to assess the level of damage?
9. Which service should you use to process the scanned claim detail and diagram that the customer has uploaded?

Thought experiment answers

This section contains the solutions to the thought experiment. Each answer explains why the answer choice is correct.

1. Image classification is a machine learning model that predicts the category, or class, the contents of the image belong to. The categories are the level of damage involved.
2. Optical character recognition (OCR) extracts text from an image. You should use OCR to read the license plate of the vehicle but would not be able to assess the level of damage to the vehicle.
3. Identifying people in an image is an example of facial detection and recognition.
4. Object detection will identify and tag the vehicle and may be able to identify the manufacturer and model but will not be able to assess the level of damage to the vehicle.
5. No, you cannot use the free tier with a multi-service resource. You must create resources for each Vision service if you want to use the free tier.
6. No, the Vision service supports file sizes less than 4MB only.
7. You should use the Custom Vision service to assess damage. A set of images can be used to train a custom model. A new image can then be categorized using the model. You would train the model with sets of vehicle images with differing levels of damage with categories (tags) that you define. The model will then be able to place any new image in one of the categories.
8. You should use the image classification model type rather than object detection. Image classification categorizes the images.
9. You should use the Vision OCR capability. This capability can process both images and documents and is able to match form fields to data items, extracting the data in a structured format that your application can process.

Describe features of natural language processing (NLP) workloads on Azure

Natural language processing (NLP) is the processing of text and speech to extract meaning. NLP can interpret written text and spoken audio and provide detail and understanding about the language used in a computer-readable form.

Natural language processing services are provided by Azure AI Services. In this chapter, the concepts involved in NLP will be outlined with use cases, followed by how to use the various features of the Azure AI Language service.

Language has been a major focus for AI. For many organizations, processing is centered around text and documents, and there are large gains to be made by applying AI. There are large amounts of text held in computer-readable formats that can be used to train models. AI can assist users in handling a significant volume of text data and audio streams. For instance, AI can:

- Mine text for patterns and trends
- Transform unstructured text into more structured formats
- Generate medical codes by analyzing healthcare documents for specific phrases and terms
- Sift social media, blog feeds, and web pages to monitor trends
- Interpret requests and decide on the best action to take, acting as a digital assistant
- Translate text from one language into other languages

This chapter describes the prebuilt language-related AI capabilities provided by Azure AI Services and explains how to use them.

Skills covered in this chapter:

- [Skill 4.1: Identify features of common NLP workload scenarios](#)
- [Skill 4.2: Identify Azure tools and services for NLP workloads](#)

Skill 4.1: Identify features of common NLP workload scenarios

Natural language processing is concerned with understanding written and spoken language. There are countless scenarios for using NLP in applications for organizations across many industries.

Humans can instinctively read and understand the meaning of a piece of unstructured text, but NLP does not understand the text in the same way. NLP is the process of gaining insights into text and extracting some meaning from that text.

A major aspect of the Microsoft Azure AI Fundamentals certification is the capabilities and features of natural language processing, including how language services can be applied in various scenarios. This requires you to understand the use cases for natural language processing and to be able to differentiate the various language services in Microsoft Azure.

This skill covers how to:

- [Describe natural language processing](#)
- [Describe language modeling](#)
- [Describe key phrase extraction](#)
- [Describe named entity recognition](#)
- [Describe sentiment analysis](#)
- [Describe speech recognition and synthesis](#)
- [Describe translation](#)

Describe natural language processing

Natural language processing (NLP) applies AI algorithms and models to text in documents, email messages, and other sources, as well as to speech, and then extracts attributes about the text to gain insights from the text.

NLP can be used to classify or summarize documents by identifying the subjects they contain. The output of NLP can be used for further text-based processing or searching.

NLP can extract key phrases from a document, as well as recognize questions and intent from text and extract those into requests and actions. NLP can perform sentiment analysis on a document to decide how positively or negatively the language is used.

NLP can interpret spoken language and synthesize speech responses.



Exam Tip

You will need to be able to map the NLP workloads on the scenarios presented or identify which type of NLP workload applies to the requirements described.

Text analytics techniques

NLP is concerned with applying the following processing techniques to extract meaning and gain insights:

- Analysis of text
 - **Tokenize** Splitting text into words and phrases
 - **Statistical** Analyzing the terms used, including the frequency of the appearance of individual words
 - **Frequency** Identifying the frequency of individual words and of phrases
 - **Part of speech tagging (PosTag)** Assigning parts of speech (noun, verb, or adjective) to each word.
 - **Sentiment analysis** Scoring the text for sentiment, as having a positive or negative feeling
 - **Language detection** Detecting the predominate language used in the text
- Language modeling
 - **Semantic modeling** Identifying the relationships between words
 - **Named entity recognition (NER)** Identifying objects (places, dates, or quantities) in the text
 - **Topic detection** Combining entities into topics to describe the important topics present in the text
- Analysis of speech
 - **Conversion of audio into text** Analyzing and interpreting speech and converting it into text
 - **Conversion of text into audio** Analyzing text, identifying phrases, and synthesizing those phrases into spoken audio
- Translation
 - **Conversion of languages** Automatic translation between languages for both text and speech

In [Figure 4-1](#) the Immersive Reader tool demonstrates the use of speech tagging. Each word is identified with a tag (adjective, noun, or verb) and is color coded.

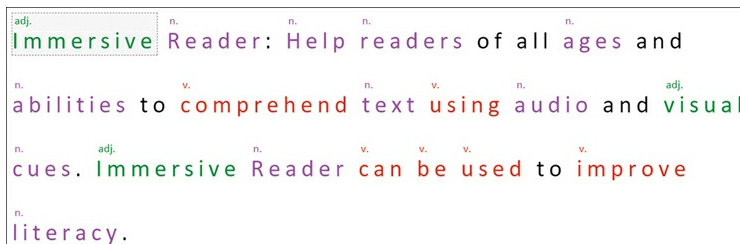


FIGURE 4-1 Immersive Reader

Use cases for NLP

There are many use cases for NLP. A few examples are:

- A company can search for terms in documents related to their products. For instance, a patent attorney could use NLP to look for examples of prior use when evaluating new patents, or an attorney could protect existing patents by searching for terms in their patents against patents submitted by others.
- An organization can monitor social media posts for sentiment and then flag negative sentiment. An example would be an airline that monitors X for posts that mention the airline's name. Any negative post could be flagged, and the follower count of the poster could be used to determine the level of response required.
- A call center can use NLP to analyze phone calls. NLP can transcribe the call and identify which voice is speaking.
- Organizations can classify emails in customer support. Emails can be prioritized for the urgency of response, if the email is SPAM, or even if an email requires a response at all.

Now, you will examine the major NLP workloads.

Describe language modeling

A common use of NLP is conversations between humans and chatbots or personal digital assistants. To have human-like interactions such as these requires the computer to interpret the meaning of the text supplied, understand what is being requested, and what the request is about. It requires a language model.

Note Language Model

A *language model* is a core component in NLP. The language model uses the statistical techniques described previously to analyze the structures and patterns in text. For instance, a language model will look at the probability of a sequence of words. This is more than extracting the nouns and verbs in a sentence. It is having a model of the language to be able to understand the text supplied. It is more than rules for grammar; humans do not always follow the complex rules for grammar, and these rules vary over time and context.

Language models are trained to learn the structure of a language by analyzing enormous amounts of text. Language modeling aims to interpret the intent from a text statement and extract key information to discover the overall meaning from the text.

Language modeling is at the heart of creating NLP-based solutions and forms the basis for the other workloads described in this section. Converting a command into smart actions is an example of language modeling. Language modeling interprets the purpose of a text command and turns that command into an intent that can be converted into a smart action for a device to run.

Language modeling does not detect the language in which the text is written. That is a task for language *detection*, not language modeling. Language modeling can be performed in many languages but only one language at a time. Similarly, language modeling does not determine the emotion in a text statement. Determining the emotion in a text statement is *sentiment analysis*, not language modeling.

Describe key phrase extraction

Key phrase extraction is the evaluation of a piece of text to identify the main talking points, or popular mentions, contained in the text. Key phrase extraction returns a list of talking points. Each talking point is normally a single word or a short phrase.

An example use for key phrase extraction is for website reviews. You can extract key phrases from all the reviews and then add these elements as tags to allow other website users to filter reviews. For example, for the review text, “The food was delicious and there were wonderful staff,” the key phrase extraction returns the talking points: “food” and “wonderful staff.”

Describe named entity recognition

Named entity recognition (NER) is the identification of entities in a piece of text. Entities are of a type, or class (such as people, places, organizations, numbers, or personal information), and are objects that are already known by the AI workload. Some entities like dates have subtypes, such as time or duration.

Some common entities available with NER are:

- People
- Places
- Organizations
- Dates and times
- Date
- Duration
- Time
- Quantities
- Age
- Temperature

NER can also find personal information, including email addresses, telephone numbers, social security numbers, driver license numbers, and passport numbers.

NER can be used to extract brand information in emails and social media posts. You can use NER to find mentions of your product names.

You can use NER to scan news feeds for mentions of people or organizations and tag those documents. For example, you can create a curated list of daily news articles that are relevant to your company or clients.

Describe sentiment analysis

A common use for NLP is to score text for sentiment. *Sentiment analysis* is the determination if a piece of text has a positive or negative feeling or emotion.

Sentiment analysis can be used to listen to what customers think of your products and services by analyzing social media for positive or negative sentiment. There are many third-party applications—for example, Hootsuite—that monitor and track social media using sentiment analysis, allowing you to monitor trends and to take action.

Describe speech recognition and synthesis

NLP is not just for text; it includes the spoken word. Speech covers both *speech recognition* (detecting and interpreting spoken input and converting into text) and *speech synthesis* (generating spoken output from text).

Speech recognition

Speech recognition does not simply recognize words; it must find patterns in the audio. Speech recognition uses an acoustic language model that converts the audio into phonemes. A *phoneme* is the smallest unit of sound in speech. When we teach children to read, we teach how letters are represented by sounds. In English, however, there are 26 letters in the alphabet, but there are 44 phonemes, and similar words can be pronounced very differently.

The word “scone” (a baked good) is an example of the different phonemes. Scone has two common pronunciations in England: one that rhymes with “cone” and another that rhymes with “gone.” If you are in Scotland, you might hear it pronounced as “skoon.”

Once the phonemes have been identified, the acoustic model then maps phonemes to words, using statistical algorithms that predict the most probable sequence of words based on the phonemes.

An example of speech recognition can be found in Microsoft PowerPoint. The Presenter Coach tool monitors your speech and uses speech recognition to give you a statistical report for a rehearsal of your presentation. It will tell you if you have used filler words or euphemisms, and it detects if you are just reading the text from the slide. It will also provide suggestions to improve your delivery.

Other common NLP scenarios for speech recognition include interactive voice response in call centers, transcribing telephone calls, and in-home automation.

Speech synthesis

Synthesized speech does not generate a sound for each word. It converts the text into phrases, and then using the acoustic model, converts the phrases into smaller prosodic units and then into phonemes. A voice is then applied to convert the phonemes into audio speech. The voice defines the pitch, speaking rate, and intonation for the generated audio.

You find speech synthesis in personal digital assistants, like Siri and Cortana, that respond vocally to your requests.

Common NLP scenarios for speech synthesis are broadcasting arrivals and departures at airports, reading out text messages while you are driving your car, and screen reading software applications for people with visual impairments.

Describe translation

There are many languages in the world, and the ability to convert text from one language into another language is a feature of NLP.

Translation is very difficult for humans to do. (Disclaimer: My wife is a translator from Italian to English.) Languages do not have simple word-for-word translations; for instance, there are words in Italian that can be one of multiple words in English, depending on the context. There are Italian words that do not have a simple translation, such as *ciofecca*, which means a poor quality and badly prepared drink, such as coffee. Another example is that *volume* in Italian means the space inside an object, whereas in English generally the word refers to a measurement of that space—or to a level of sound or a book in a series. The German word *schadenfreude* has no direct translation and generally means to derive pleasure from someone else's misfortune. In Finnish, the word *kalsarikännit* means the feeling you get when sitting at home getting drunk in your underwear. As you can see, translation is hard and relies on context and a lot of knowledge.

Most human translators translate from one language into their native language; they do not translate the other way around, as they need to make the text understandable to the native speaker. You have probably all seen instruction manuals that were poorly translated by a non-native speaker.

To translate from one language to another requires models for both languages and to be able to understand the context of how each language is used. This involves understanding rules of grammar, use of informal language, and large dictionaries and glossaries.

AI-powered text translation uses large amounts of translated text to train the translation models. You will find that the translations are better when more examples of the text are available in each language, so the results of translation between English and a language are often better than translation between two other languages. Translation results tend to be better on news and marketing documents than on highly technical documents, as again, there are more examples of the former available to train the models on.

Translation includes the conversion of both text and audio speech from one language into another:

- **Text** Text translation translates the text documents from one language to another language.
- **Speech** Speech translation translates spoken audio from one language to another.

Now that you have an understanding of the common NLP workloads, it's time to turn your attention to the services in Azure AI Services for the major NLP workloads.

Skill 4.2: Identify Azure tools and services for NLP workloads

Azure AI Services provide pretrained NLP models that cover most of the capabilities required for analyzing text and speech. This section describes the capabilities of the main language services within Azure AI Services.

Available for both text and speech, these language services allow developers to add features such as sentiment detection, speech recognition, and language understanding to their applications without needing skills in machine learning.

These text services extract meaning from unstructured text:

- **Azure OpenAI** Offers natural language processing through powerful general-purpose generative models like GPT for tasks such as content generation, code completion, and conversational AI. Azure OpenAI Service is explained in [Chapter 5, “Describe features of generative AI workloads on Azure.”](#)
- **Document Intelligence** Extracts structured data from documents like invoices, receipts, and contracts.
- **Language** Provides capabilities such as entity recognition, sentiment analysis, key phrase extraction, summarization, and more. This service replaces legacy offerings like Text Analytics and LUIS.
- **Translator** Detects and translates text in real time, or in a batch, across more than 100 languages.
- **Immersive Reader** Helps readers of all ages and abilities to comprehend text using audio and visual cues.

These speech services allow you to add speech processing into your apps:

- **Speech to Text** Transcribes audio into text in real time or from audio files.
- **Text to Speech** Synthesizes text into spoken audio.
- **Speech Translation** Converts audio into text and translates into another language in real time. Speech Translation leverages the Translator service.

- **Speaker Recognition** Identifies people from the voices in an audio clip.

The Microsoft Azure AI Fundamentals certification encompasses the capabilities of natural language processing services. This requires you to understand the capabilities of the services described in this section.

This skill covers how to:

- Identify the capabilities of the [Azure AI Language service](#)
- Identify the capabilities of the [Azure AI Speech service](#)
- Identify the capabilities of the [Azure AI Translator service](#)



Exam Tip

You need to be able to map the service to the scenario presented and to identify the operation for a service to use.

Language services are used alongside other AI Services, such as OCR described in [Chapter 3](#), “Describe features of computer vision workloads on Azure,” and with bots and generative AI services described in [Chapter 5](#).

To use the language services, you will need to create an Azure AI multi-service resource, or a language single-service resource, as described in [Chapter 3](#).

The following sections describe the capabilities of the main language APIs provided by Azure AI Services.

Note Generative AI vs. AI Services

Even with the rise of generative AI, prebuilt services like Text Analytics, Translator, and Language are often more efficient, cost-effective, and easier to integrate.

For tasks like sentiment analysis or entity recognition, they could offer cheaper, faster, more reliable solutions without the need for prompt engineering and formatting rules.

Identify the capabilities of the Azure AI Language service

The Language service is an Azure AI Service that performs a series of natural language processing operations on text. The Language service can detect the sentiment of sentences or whole paragraphs. With it, you can extract key phrases and such entities as people, places, and things from a piece of text.

To deploy the Language service, create a new resource in the Azure portal and select Language. You must then select your region, select the resource group, provide a unique name, and select the pricing tier: Free F0 or Standard.

You can create Language resources using the CLI, as follows:

```
az cognitiveservices account create --name <unique name> --resource-group <resource group name> --kind Language --sku F0 --location <region>
```

The following sections describe the capabilities of the operations available with the Language service. You will see that the request URL stays the same for Detect Language, Sentiment Analysis and Opinion Mining, Key Phrase Extraction, and Named Entity Recognition (NER), while the JSON payload in the request body changes depending on the specific service being called. In particular, the main differences are in the `kind` property and, when applicable, the presence of an additional `parameters` property.

Detect Language

The Language service supports a wide range of languages. A key operation is the identification of the language used in a document. Language can detect a wide range of languages, variants, dialects, and some regional/cultural languages—for example, French (`fr-FR`) vs. Canadian French (`fr-CA`).

The *Detect Language* operation returns the name of the language, the ISO language code, and a level of confidence between 0 and 1. If there is more than one language in the document, then the predominant language is returned.

Formulate the request URL as follows:

```
https://{endpoint}/language/:analyze-text?api-version=2023-11-15-preview
```

You supply a JSON document containing text to be analyzed formatted as follows:

```
{
  "kind": "LanguageDetection",
  "parameters": {
    "modelVersion": "latest"
  },
  "analysisInput": {
    "documents": [
      {
        "id": "1",
        "text": " We went to Contoso Steakhouse located at midtown NYC last week
for a dinner
party, and we adore the spot! They provide marvelous food and they
have a great
menu. The chief cook happens to be the owner (I think his name is John
```

```

Doe) and he is ↵
super nice, coming out of the kitchen and greeted us all. We enjoyed
very much dining ↵
in the place! The Sirloin steak I ordered was tender and juicy, and
the place was ↵
impeccably clean. You can even pre-order from their online menu at ↵
www.contososteakhouse.com call 312-555-0176 or send email to ↵
order@contososteakhouse.com! The only complaint I have is the food
didn't come fast ↵
enough. Overall I highly recommend it!"
    }
  ]
}

```

The following JSON is returned:

```

{
  "kind": "LanguageDetectionResults",
  "results": {
    "documents": [
      {
        "id": "1",
        "warnings": [
          ],
        "detectedLanguage": {
          "name": "English",
          "iso6391Name": "en",
          "confidenceScore": 1.0,
          "script": "Latin",
          "scriptCode": "Latn"
        }
      }
    ],
    "errors": [
      ],
    "modelVersion": "2024-11-01"
  }
}

```

This operation found that the language was English, an ISO code `en`, with a confidence score of 100%.

Sentiment Analysis and Opinion Mining

Sentiment Analysis analyzes the emotion for each sentence in a piece of text and for the whole document.

Sentiment is a classification model that evaluates the emotion of the text as to how positive or negative it is. The operation returns a sentiment score between 0 and 1 (1 is the most positive) and a sentiment label (positive, negative, neutral, or mixed).

Opinion Mining is a feature of sentiment analysis that provides more detailed insight into opinions expressed in the text. Also known as *aspect-based sentiment analysis*, it identifies not just the overall sentiment of a sentence or document, but also the sentiment associated with specific targets or aspects, such as a product feature or service attribute.

Sentiment Analysis and Opinion Mining use a model that has been pretrained on hundreds of millions of examples of text. Currently, both Sentiment Analysis and Opinion Mining can be evaluated for more than 90 languages.

Formulate the request URL as follows:

```
https://{endpoint}/language/:analyze-text?api-version=2023-11-15-preview
```

Note that, as mentioned before, the URL used to make the request is the same as the one used for Language Detection; what changes is the structure of the JSON payload sent to the service. In this case, the JSON structure containing the document to be analyzed is as follows:

```

{
  "kind": "SentimentAnalysis",
  "parameters": {
    "modelVersion": "latest",
    "opinionMining": "False"
  },
  "analysisInput": {
    "documents": [
      {
        "id": "1",
        "language": "en",
        "text": "The food and service were unacceptable. The concierge was
awesome, however "
      }
    ]
  }
}

```

The value of the `opinionMining` parameter determines whether the output will include only the overall sentiment analysis or a more detailed analysis using the Opinion Mining feature.

Following is the JSON returned with `OpinionMining` set to `False` for a document containing the sentence “The food and service were unacceptable. The concierge was awesome, however.”:

```

{
  "kind": "SentimentAnalysisResults",
  "results": {
    "documents": [
      {
        "id": "1",

```

```

"sentiment": "mixed",
"confidenceScores": {
  "positive": 0.49,
  "neutral": 0.01,
  "negative": 0.5
},
"sentences": [
  {
    "sentiment": "negative",
    "confidenceScores": {
      "positive": 0.0,
      "neutral": 0.0,
      "negative": 1.0
    },
    "offset": 0,
    "length": 40,
    "text": "The food and service were unacceptable. "
  },
  {
    "sentiment": "positive",
    "confidenceScores": {
      "positive": 0.97,
      "neutral": 0.02,
      "negative": 0.0
    },
    "offset": 40,
    "length": 35,
    "text": "The concierge was awesome, however."
  }
],
"warnings": [
]
},
"errors": [
],
"modelVersion": "2025-01-01"
}

```

This operation found that the overall sentiment of the paragraph is mixed with a positive score of 49%, a neutral score of 1%, and a negative score of 50%. The first sentence is 100% negative, while the second sentence is 97% positive.

If the same request is made with the `opinionMining` parameter set to `true`, the returned JSON is:

```

{
  "kind": "SentimentAnalysisResults",
  "results": {
    "documents": [
      {
        "id": "1",
        "sentiment": "mixed",
        "confidenceScores": {
          "positive": 0.49,
          "neutral": 0.01,
          "negative": 0.5
        }
      }
    ],
    "sentences": [
      {
        "sentiment": "negative",
        "confidenceScores": {
          "positive": 0.0,
          "neutral": 0.0,
          "negative": 1.0
        },
        "offset": 0,
        "length": 40,
        "text": "The food and service were unacceptable. ",
        "targets": [
          {
            "sentiment": "negative",
            "confidenceScores": {
              "positive": 0.01,
              "neutral": 0.0,
              "negative": 0.99
            },
            "offset": 4,
            "length": 4,
            "text": "food",
            "relations": [
              {
                "relationType": "assessment",
                "ref": "#/documents/0/sentences/0/assessments/0"
              }
            ]
          }
        ]
      },
      {
        "sentiment": "negative",
        "confidenceScores": {
          "positive": 0.01,
          "neutral": 0.0,
          "negative": 0.99
        },
        "offset": 13,
        "length": 7,
        "text": "service",

```

```

        "relations":[
          {
            "relationType":"assessment",
            "ref":"#/documents/0/sentences/0/assessments/0"
          }
        ]
      },
      "assessments":[
        {
          "sentiment":"negative",
          "confidenceScores":{
            "positive":0.01,
            "negative":0.99
          },
          "offset":26,
          "length":12,
          "text":"unacceptable",
          "isNegated":false
        }
      ]
    },
    {
      "sentiment":"positive",
      "confidenceScores":{
        "positive":0.97,
        "neutral":0.02,
        "negative":0.0
      },
      "offset":40,
      "length":35,
      "text":"The concierge was awesome, however.",
      "targets":[
        {
          "sentiment":"positive",
          "confidenceScores":{
            "positive":1.0,
            "negative":0.0
          },
          "offset":44,
          "length":9,
          "text":"concierge",
          "relations":[
            {
              "relationType":"assessment",
              "ref":"#/documents/0/sentences/1/assessments/0"
            }
          ]
        }
      ]
    }
  ],
  "assessments":[
    {
      "sentiment":"positive",
      "confidenceScores":{
        "positive":1.0,
        "negative":0.0
      },
      "offset":58,
      "length":7,
      "text":"awesome",
      "isNegated":false
    }
  ]
}
],
"warnings":[
]
},
"errors":[
],
"modelVersion":"2025-01-01"
}
}

```

As you can see, this approach provides a more fine-grained sentiment analysis compared to the previous case. Opinion Mining can detect a negative opinion about the food and service, and a positive opinion about the concierge. This level of granularity helps better understand what exactly users feel positively or negatively about.

Key Phrase Extraction

Key Phrase Extraction identifies the main talking points in unstructured text. The operation generates a list of relevant words and/or phrases that describes the subject of the document. It should be noted that the phrases are returned as a simple list without any context.

Key Phrase Extraction uses a model that has been pretrained on millions of examples of text. Currently, phrases can be extracted for more than 90 languages.

The request URL is the same as the one used for language detection and sentiment analysis:

```
https://{endpoint}/language/:analyze-text?api-version=2023-11-15-preview
```

You supply a JSON document containing text to be analyzed:

```

{
  "kind": "KeyPhraseExtraction",
  "parameters": {
    "modelVersion": "latest"
  },
  "analysisInput": {
    "documents": [
      {
        "id": "1",
        "language": "en",
        "text": "The food and service were unacceptable. The concierge was
awesome, however."
      }
    ]
  }
}

```

Following is the JSON returned:

```

{
  "kind": "KeyPhraseExtractionResults",
  "results": {
    "documents": [
      {
        "id": "1",
        "keyPhrases": [
          "food",
          "service",
          "concierge"
        ],
        "warnings": [
        ]
      }
    ],
    "errors": [
    ],
    "modelVersion": "2022-10-01"
  }
}

```

The key phrase extraction operation found three key phrases in the paragraph of text.

Named Entity Recognition

Named Entity Recognition (NER) extracts a wide set of prebuilt entities from documents. Entities are objects in the text such as people, places, organizations, date/time, numbers, and personally identifiable information (PII). For PII detection, Azure provides a separate API where the JSON request body must include the `kind` property set to `PiiEntityRecognition` while for all the others this property should be set to `EntityRecognition`.

Note Entity Recognition

Microsoft has referred to the Named Entity Recognition feature as *entity recognition* in the past but is now using the more common term in its documentation. In the exam, you may see usage of entity recognition. Named Entity Recognition and entity recognition refer to the same operation.

The categories of entities extracted by the Language service are:

- **Person** The names of people. This is not limited to famous people but can identify forenames and surnames in the text.
- **PersonType** The job type or job title.
- **DateTime** Dates and times of day including durations and date ranges.
- **Quantity** Numerical measurements and units including temperature, percentages, ages, and dimensions.
- **Location** A geographical feature, landmark, building, or city.
- **Organization** The names of companies, political groups, musical bands, sports teams, government bodies, and public organizations.
- **Event** The names of historical, social, and other events.
- **Product** Physical objects. Currently these are computing related.
- **Skill** Capabilities, skills, or expertise.
- **Address** Addresses including street, city, and postal code.
- **Phone number** Telephone numbers.
- **Email** Email addresses.
- **URL** URLs to websites.
- **IP** Network IP addresses.

If none of the predefined entity types meet your needs, Language provides a feature called *Custom Named Entity Recognition*. This is a cloud-based API service that leverages machine learning to allow you to build custom models tailored to recognize domain-specific entities in unstructured text, such as clauses in contracts or terms in financial documents.

By creating a Custom NER project, you can iteratively label data, train the model, evaluate its performance, and refine it before making it available for use. The accuracy and effectiveness of the model are strongly influenced by the quality of the labeled data. To simplify the process of building and

customizing your model, the service provides a dedicated web interface accessible through Language Studio.

The operation returns the entity, its category, and a confidence score between 0 and 1. Currently, entities can be extracted for more than 70 languages, including Arabic.

Note Entity Linking

The Language service also provides links to Wikipedia articles for well-known entities. To do so, in the JSON set the `kind` property in the request to `EntityLinking`.

Formulate the request URL as follows:

```
https://{endpoint}/language/:analyze-text?api-version=2023-11-15-preview
```

You supply a JSON document containing text to be analyzed. The JSON you send should look like this example (borrowed from Microsoft documentation):

```
{
  "kind": "EntityRecognition",
  "parameters": {
    "modelVersion": "latest"
  },
  "analysisInput": {
    "documents": [
      {
        "id": "1",
        "language": "en",
        "text": "I had a wonderful trip to Seattle last week."
      }
    ]
  }
}
```

Following is the JSON returned:

```
{
  "kind": "EntityRecognitionResults",
  "results": {
    "documents": [
      {
        "id": "1",
        "entities": [
          {
            "text": "trip",
            "category": "Event",
            "type": "Event",
            "offset": 18,
            "length": 4,
            "confidenceScore": 0.66,
            "tags": [
              {
                "name": "Event",
                "confidenceScore": 0.66
              }
            ]
          },
          {
            "text": "Seattle",
            "category": "Location",
            "type": "City",
            "offset": 26,
            "length": 7,
            "confidenceScore": 1.0,
            "tags": [
              {
                "name": "City",
                "confidenceScore": 1.0
              },
              {
                "name": "GPE",
                "confidenceScore": 1.0
              },
              {
                "name": "Location",
                "confidenceScore": 1.0
              }
            ]
          },
          {
            "text": "last week",
            "category": "Temporal",
            "type": "DateRange",
            "offset": 34,
            "length": 9,
            "confidenceScore": 0.8,
            "tags": [
              {
                "name": "DateRange",
                "confidenceScore": 0.8
              },
              {
                "name": "Temporal",

```

```

        "confidenceScore":0.8
      }
    ],
    "metadata":{
      "metadataKind":"TemporalSpanMetadata",
      "spanValues":[
        {
          "timex":"2025-W27",
          "begin":"2025-06-30",
          "end":"2025-07-07"
        }
      ]
    }
  ],
  "warnings":[]
}
],
"errors":[]
},
"modelVersion":"2025-02-01"
}
}

```

Three entities were extracted: Trip as an event with confidence score of 66%, Seattle as Location with a confidence score of 100%, and last week as a DateTime with a confidence score of 100%.

If you set kind to EntityLinking for the same text, the JSON returned is:

```

{
  "kind":"EntityLinkingResults",
  "results":{
    "documents":[
      {
        "id":"1",
        "entities":[
          {
            "bingId":"5fbbab6b8-85e1-4d41-9444-d9055436e473",
            "name":"Seattle",
            "matches":[
              {
                "text":"Seattle",
                "offset":26,
                "length":7,
                "confidenceScore":0.19
              }
            ]
          },
          {
            "language":"en",
            "id":"Seattle",
            "url":"https://en.wikipedia.org/wiki/Seattle",
            "dataSource":"Wikipedia"
          }
        ],
        "warnings":[]
      }
    ],
    "errors":[]
  },
  "modelVersion":"2021-06-01"
}

```

One Wikipedia article has been identified for the Seattle city.

Note Named Entity Recognition on Your Premises

Named Entity Recognition (and Custom Named Entity Recognition models) can be run locally using Docker containers, giving you full control over deployment and data processing within your own environment. This is useful especially for PII detection.

Summarize text-based content

The *Summarization* feature of Azure AI Language combines generative large language models with task-optimized encoder models to deliver high-quality, low-latency, and cost-efficient text summarization capabilities. This service supports three types of input: plain text, conversations, and native documents. Depending on the input type, the Summarization process adapts accordingly, handling free-form text blocks, spoken dialogue, or files in formats such as Microsoft Word, PDF, and plain text. This flexibility makes the Summarization feature suitable for a wide range of applications, from document processing to conversation analysis.

To help you understand how the Summarization feature works, walk through an example in which the input is a plain text block. The first step is to send a POST request to the following URL:

`https://{endpoint}/language/analyze-text/jobs?api-version=2023-04-01`

In the body of this request, you must include a JSON payload formatted as follows (example taken from Microsoft documentation):

```
{
```

```

"displayName": "Text ext Summarization Task Example",
"analysisInput": {
  "documents": [
    {
      "id": "1",
      "language": "en",
      "text": "At Microsoft, we have been on a quest to advance AI beyond existing
techniques, by taking a
more holistic, human-centric approach to learning and understanding. As Chief
Technology Officer
of Azure AI services, I have been working with a team of amazing scientists
and engineers to turn
this quest into a reality. In my role, I enjoy a unique perspective in viewing
the relationship among
three attributes of human cognition: monolingual text (X), audio or visual
sensory signals, (Y) and
multilingual (Z). At the intersection of all three, there's magic-what we call
XYZ-code as illustrated
in Figure 1-a joint representation to create more powerful AI that can speak,
hear, see, and
understand humans better. We believe XYZ-code will enable us to fulfill our
long-term
vision:
cross-domain transfer learning, spanning modalities and languages. The goal is
to have
pre-trained models that can jointly learn representations to support a broad
range of downstream
AI tasks, much in the way humans do today. Over the past five years, we have
achieved human
performance on benchmarks in conversational speech recognition, machine
translation,
conversational question answering, machine reading comprehension, and image
captioning. These
five breakthroughs provided us with strong signals toward our more ambitious
aspiration to
produce a leap in AI capabilities, achieving multi-sensory and multilingual
learning that is closer in
line with how humans learn and understand. I believe the joint XYZ-code is a
foundational
component of this aspiration, if grounded with external knowledge sources in
the downstream
AI tasks."
    }
  ]
},
"tasks": [
  {
    "kind": "ExtractiveSummarization",
    "taskName": "Text Extractive Summarization Task 1",
    "parameters": {
      "sentenceCount": 6
    }
  }
]
}

```

The `sentenceCount` parameter specifies how many sentences the text should be summarized into. If you do not specify the `sentenceCount` parameter, the model will determine the summary's length. Note that `sentenceCount` is just the approximation of sentence count of output summary, range 1 to 7.

The service will respond with a message containing a job id. From that response, you need to extract the `<your_job_id>` value and use it to poll the result by making a GET request to the following URL, replacing `<jobId>` with the actual ID:

```

https://<your-language-resource-endpoint>/language/analyze-text/
jobs/<your-job-id>?api-version=2023-04-01

```

The final response will contain the summarization result, in a format similar to this:

```

{
  "jobId": "49d097fa-8a39-4d02-a7f3-97c4f656a13b",
  "lastUpdatedDateTime": "2025-05-11T16:58:25Z",
  "createdDateTime": "2025-05-11T16:58:23Z",
  "expirationDateTime": "2025-05-12T16:58:23Z",
  "status": "succeeded",
  "errors": []
},
  "displayName": "Text ext Summarization Task Example",
  "tasks": {
    "completed": 1,
    "failed": 0,
    "inProgress": 0,
    "total": 1,
    "items": [
      {
        "kind": "ExtractiveSummarizationLROResults",
        "taskName": "Text Extractive Summarization Task 1",
        "lastUpdateDateTime": "2025-05-11T16:58:25.4900221Z",
        "status": "succeeded",
        "results": {
          "documents": [
            {
              "id": "1",
              "sentences": [
                "text": "At Microsoft, we have been on a quest to advance AI
beyond existing
techniques, by taking a more holistic, human-centric

```



```

"total": 2,
"items": [
  {
    "kind": "conversationalSummarizationResults",
    "taskName": "Conversation Task 1",
    "status": "succeeded",
    "results": {
      "conversations": [
        {
          "summaries": [
            {
              "aspect": "issue",
              "text": "Customer wants to connect their Smart Brew 300 to their
Wi-Fi. | The Wi-Fi ↵
connection didn't work."
            }
          ],
          "id": "conversation1"
        }
      ]
    }
  },
  {
    "kind": "conversationalSummarizationResults",
    "taskName": "Conversation Task 2",
    "status": "succeeded",
    "results": {
      "conversations": [
        {
          "summaries": [
            {
              "aspect": "resolution",
              "text": "Asked customer to check if the power light is blinking on and
off every second."
            }
          ],
          "id": "conversation1"
        }
      ]
    }
  }
]
}
}

```

This response includes a `jobId`, similar to the text summarization output, but the results contain summaries focused on specific aspects such as the `issue` and `resolution` from the conversation.

For the `document` input type, the process is essentially the same as the previous ones. However, instead of providing plain text or conversation data directly in the JSON request, the input here refers to documents that are stored in an Azure Blob Storage container. These documents are accessed via their corresponding URLs (which you would need to replace with your own source and target URLs in the Azure portal).

Once the request is made, the output is similar to the previous cases in that it includes a `jobId`, and after the job is processed, you can query for the result. The output will provide a URL to the summarized document, and other relevant information, such as the status of the job and any warnings or errors that occurred during processing.

In summary, while the general process remains the same—submitting a request and receiving a `jobId` to later retrieve the results—the key differences lie in the format of the input JSON and the structure of the output.

Answer Questions

The *Custom Question Answering* feature from Microsoft offers a cloud-based natural language processing (NLP) solution that enables you to create a natural conversational layer over your data. This tool is designed to extract relevant answers from customer queries or project content, making it easier to find information quickly and efficiently.

Custom Question Answering is particularly useful for building conversational applications such as chatbots, social media interactions, and speech-enabled desktop applications. This offering includes key features like enhanced relevance through a deep learning ranker, precise answers, and full support across regions.

You should consider using Custom Question Answering in several scenarios:

- **When you have static information** If your project relies on static data, such as documents (PDFs, URLs), the feature can provide relevant responses to user queries based on that content.
- **When you want to provide consistent answers** If multiple users ask the same question, this feature ensures that the same answer is returned consistently, maintaining uniformity across interactions.
- **When you want to filter static information based on meta-information** You can enrich your content with metadata tags to allow for additional filtering options. This metadata can include such categories as chit-chat, content type, purpose, format, and freshness, which are valuable for providing more relevant answers tailored to your users.
- **When managing a bot conversation** If your application involves guiding users through a predetermined conversation flow, the feature can help manage and automate this flow. Whether responding to commands or questions within multi-turn conversations, it ensures a smooth interaction by returning the appropriate responses from your knowledge base.

To make REST API calls to the Custom Question Answering (CQA) feature of the Language service, you must first create a dedicated project within Language Studio that uses the Custom Question Answering capability. This project serves as the foundation for the feature's operation and must include a knowledge base populated with relevant content, typically uploaded from files such as PDFs or URLs. The knowledge base is where the system draws information from to generate accurate and consistent answers to user queries.

First, sign in to Language Studio using your Azure credentials, which you can find at language.cognitive.azure.com. Once signed in, you have to select details for the Azure Directory and Azure Subscription settings, as well as provide a name for the language resource (Figure 4-2).

Select an Azure resource ✕

To access Language studio, you need an Azure resource. You can use an existing resource or create a new one. You can change the resource you use at any time. [Learn more about resources in Azure.](#)

Azure directory *

Note: Switching directory will cause the page to refresh.

Azure subscription *

Select an existing Azure subscription or [create a free account](#) and then refresh

Resource type *

Language or Azure Cognitive Services resources can be used for any capability except for translation, which requires a Translator resource

Resource name *

Select an existing resource or create a new one

[Create a new Language resource](#)

Pricing tier:

Managed identity:

[Create a new Language resource](#)

Done **Cancel**

FIGURE 4-2 Select Language for the Resource Type setting in Language Studio

Next, select Understand Questions and Conversational Language, then Custom Question Answering (Figure 4-3).

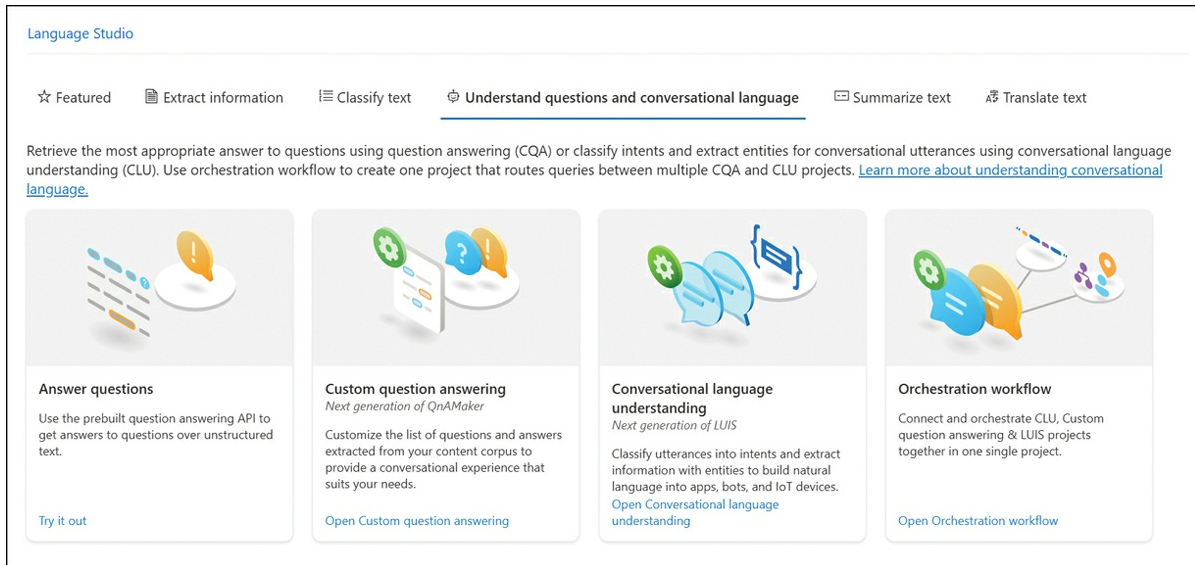


FIGURE 4-3 Understand Questions and Conversational Language section

If your Language resource is not yet connected to an Azure AI Search instance, you will be redirected to the Azure portal to complete the configuration. In the portal, from the Language Resource page, select Resource Management then Features (left), and select Enable the Custom Question Answering in the Features pane (right, Figure 4-4).

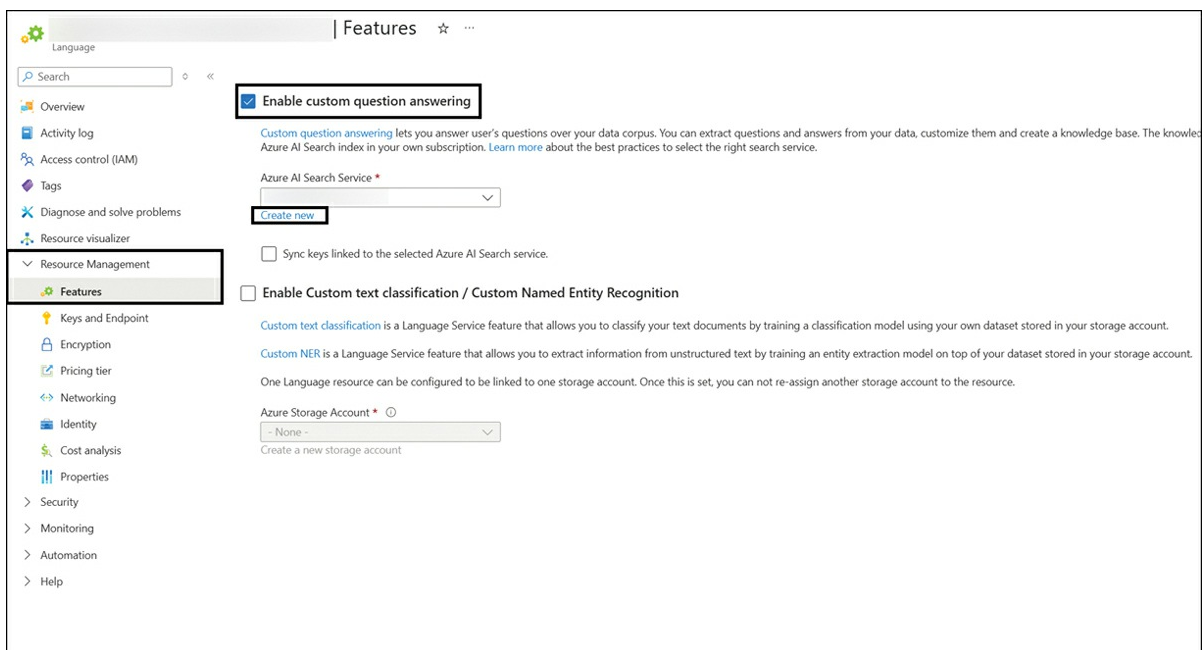


FIGURE 4-4 Connecting a Language resource to an Azure AI Search instance

At this stage, if you do not yet have an Azure AI Search resource, you will need to create one. To do so, select Create New and fill in the required fields (Figure 4-5): the name of the search resource, an existing or new resource group, the location (ensure it matches the location of the Language resource), and the pricing tier (Standard S1 or higher is required for semantic capabilities). Azure AI Search will be used to host the knowledge base and perform searches over it.

After you create the resource, return to the previous screen (Figure 4-5), select the newly created search resource from the list, and confirm the configuration by selecting Apply. If you already have a compatible search resource, simply select it from the list and confirm the connection.

Create a search service ...

Basics Scale Networking Tags Review + create

Project details

Subscription *

Resource group * [Create new](#)

Instance Details

Service name *

Location *

Pricing tier *

- Standard**
160 GB/Partition, max 12 replicas, max 12 partitions, max 36 search units
[Change Pricing Tier](#)

FIGURE 4-5 Creating a search service

Next, return to the Language Studio tab, refreshing it if needed, to register the change. You can now proceed to create a new project as shown in [Figure 4-6](#).

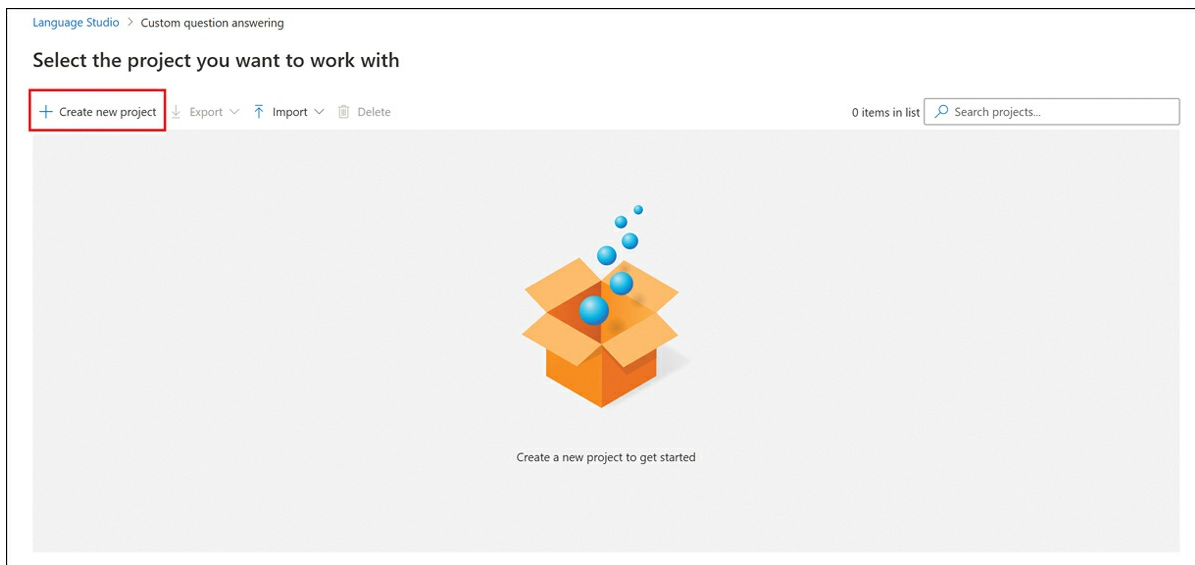


FIGURE 4-6 Project creation

Click [Create New Project](#) ([Figure 4-6](#)) to open the Create Widget Wizard. The first step is to choose a language setting for the resource. You can opt to select a language each time you create a new project for the resource or to select the language once and use it for all projects you create within the resource, as shown in [Figure 4-7](#).

Create a project
✕

- Choose language setting
- Enter basic information
- Review and finish

Choose language setting for resource ai900-azureai-languageservice.

Permanently set whether or not you can create projects in multiple languages using your Azure resource ai900-azureai-languageservice. [Learn more about projects in multiple languages and pricing.](#)

How do you want to select the language for projects in this resource?* ⓘ

I want to select the language when I create a project in this resource

When creating a project in this resource you will be able to select what language the data is in. Selecting this option will incur more costs. [Learn more about pricing](#)

I want to set the language for all projects created in this resource

All projects created in this resource will always use the same language for the data.

Select the language for all projects* ⓘ

English
▼

Back
Next
Create project
Cancel

FIGURE 4-7 Project creation, step 1

As shown in [Figure 4-8](#), you then need to name your project and, if you like, provide an optional description of the project. Leave Default Answer When No Answer Is Returned set to No Answer Found, and click Next to open the Review and Finish page. If you're satisfied with the configuration, click Create Project ([Figure 4-9](#)).

Create a project
✕

- Choose language setting
- Enter basic information
- Review and finish

Enter basic information

Enter the basic information for your custom question answering knowledge base such as name and description.

To change your resource go to [Settings](#)

Azure search resource

To change your resource go to [Azure Search](#)

Name *

Source-project

Description

Type project description

Source language * ⓘ

English
▼

Default answer when no answer is returned * ⓘ

No answer found

Back
Next
Create project
Cancel

FIGURE 4-8 Project creation, step 2

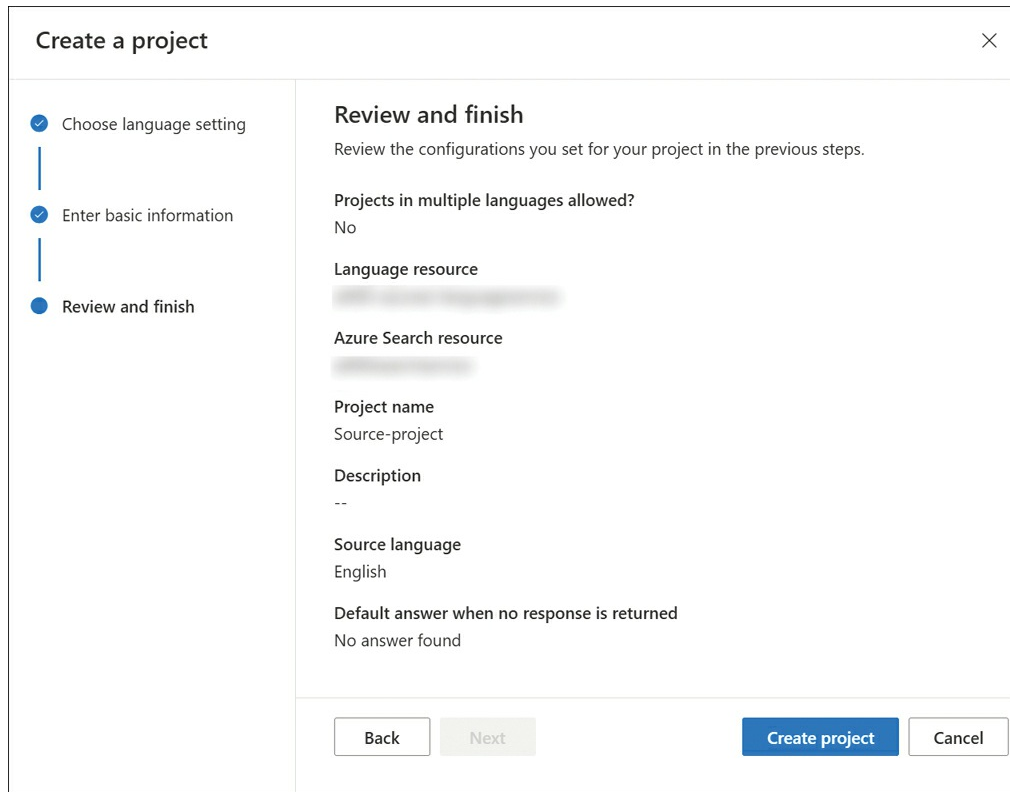


FIGURE 4-9 Project creation, step 3

With the project created, you'll be taken to the Manage Sources page (as shown in Figure 4-10).

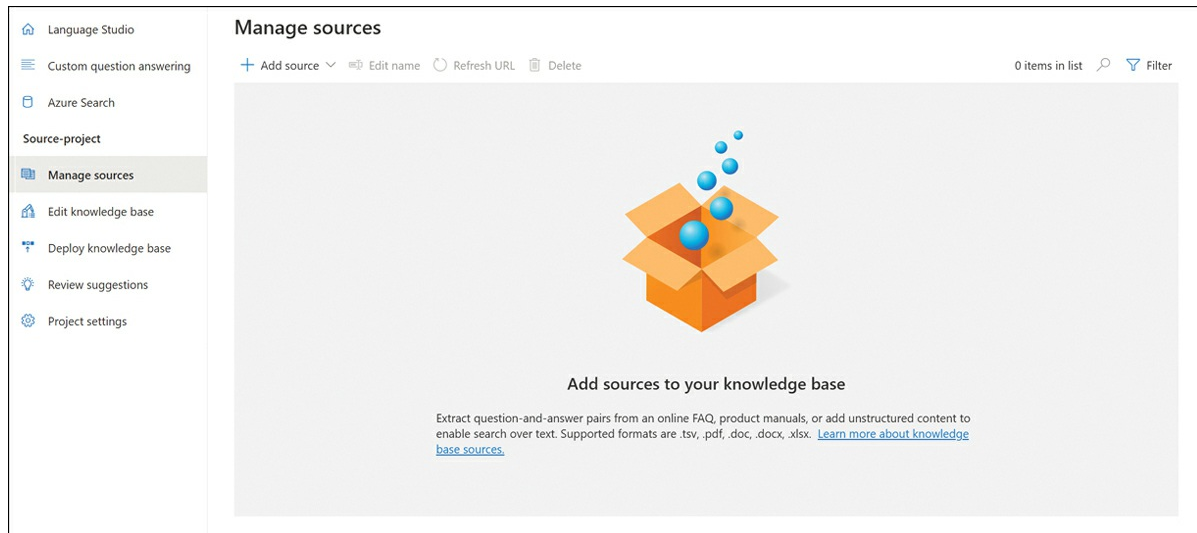


FIGURE 4-10 Manage Sources page

Here, select Add Source and choose whether to load files from your local computer or from URLs (as shown in Figure 4-11). After the sources are added, the process of searching through them will be handled by Azure AI Search.

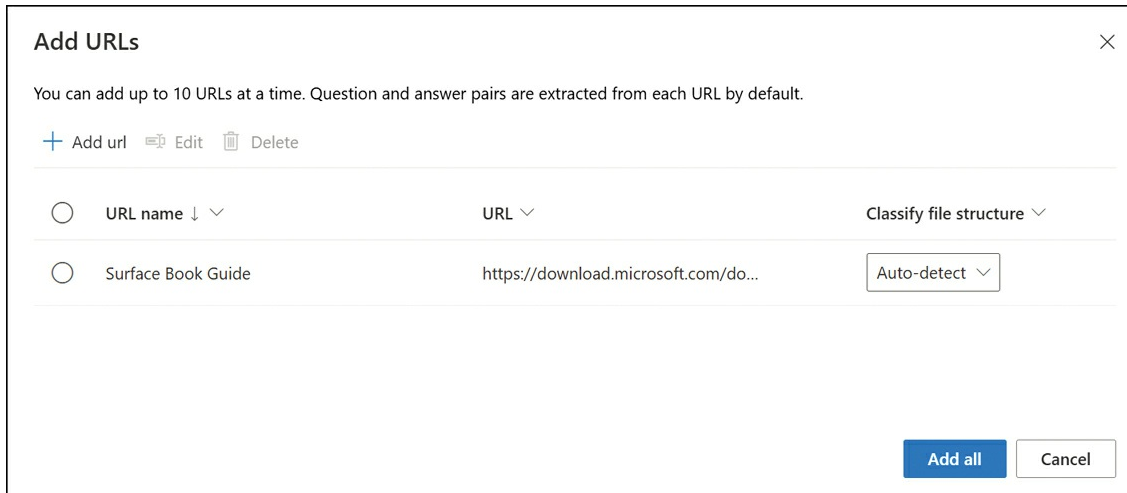


FIGURE 4-11 Add source from URLs widget

After you add the source, Language Studio will process the document, extracting potential question-answer pairs. You'll then have the opportunity to review and refine the extracted content or add entirely custom Q&A pairs, metadata, or synonyms to tailor the knowledge base to your application's needs, as shown in Figure 4-12.

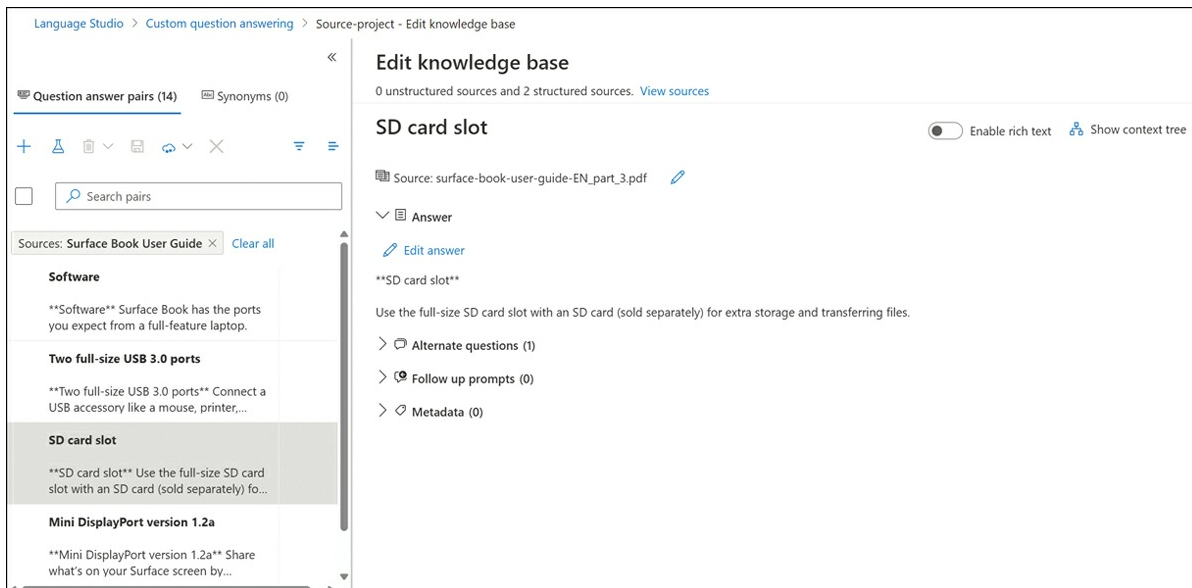


FIGURE 4-12 Edit knowledge base

After you create the project, load the knowledge base, and edit it, you can use REST APIs to get answers to user questions. The API URL is:

```
https://{endpoint}/language/:query-knowledgebases?projectName={YOUR_PROJECT_NAME}
&api-version=2021-10-01&deploymentName={DEPLOYMENT_NAME}
```

where in this case `YOUR_PROJECT_NAME` is the name of the project you created. `DEPLOYMENT_NAME` can either be `test` or `production` depending on whether you deployed your project from Language Studio > Question Answering > Deploy Project.

The request body should be a JSON formatted as follows:

```
{
  "question": "How can i change camera options?"
}
```

The return JSON will look like this:

```
{
  "answers": [
    {
      "questions": [
        "Change camera options"
      ],
      "answer": "***Change camera options**\n\nUse the Camera app to switch between the front and back camera or manually set camera options.",
      "confidenceScore": 0.99,
    }
  ]
}
```

```

    "id": 8,
    "source": "surface-book-user-guide-EN_part_13.pdf",
    "metadata": {},
    "dialog": {
      "isContextOnly": false,
      "prompts": [
        {
          "displayOrder": 0,
          "gnaId": 9,
          "displayText": "Change between the front and back camera"
        },
        {
          "displayOrder": 1,
          "gnaId": 10,
          "displayText": "Set the timer"
        },
        {
          "displayOrder": 2,
          "gnaId": 11,
          "displayText": "Change camera settings"
        },
        {
          "displayOrder": 3,
          "gnaId": 12,
          "displayText": "Turn off the camera shutter sound"
        }
      ]
    }
  }
}

```

Conversational Language Understanding

Also included with the Azure AI Language service is Conversational Language Understanding (CLU). This cloud-based API feature is an intelligent tool you can use in conversational applications to predict the overall intention of an incoming request. CLU doesn't perform any actions, it only provides the intelligence to understand the input text; the conversational application that contains it decides the actions that should be taken.

CLU enables you to create your own model for identifying the intent behind incoming utterances by training a model through standard training procedures: Select sample data, label that data, train the model, evaluate the model, and then, once satisfied with the metrics obtained, make it available for consumption. CLU also makes it possible to walk through a more streamlined deployment using previously trained LLMs and then go directly from the data selection step to the model deployment step, as shown in [Figure 4-13](#).

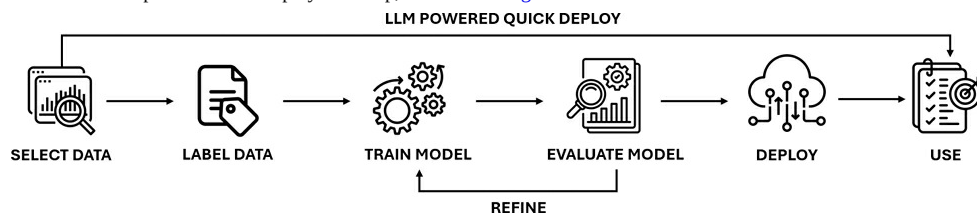


FIGURE 4-13 LLM powered quick deploy vs. custom ML deploy

Some use cases of this service could be:

- **Conversational bot** Using CLU to extract the main information from the incoming received request and decide what actions to implement. A practical example could be using CLU within a take-out food ordering bot.
- **Human assistant bot** Using CLU to sort customer requests to the appropriate support engineer.
- **Command and control applications** Using CLU to extract intent needed to perform certain actions such as starting or stopping music.
- **Guardrails over more canonical generative AI chatbots** Using CLU for added security. This feature is less prone to prompt hacking techniques.

Although you can train your own model based on personal data (from Language Studio, select Conversational Language Understanding under the Understand Questions and Conversational Language, then follow the wizard's creation steps), importing a pretrained Microsoft model is a good way to get to know the capabilities of CLU. To try the following example, download the EmailAppDemo from github.com/Azure-Samples/cognitive-services-sample-data-files/blob/master/language-service/CLU/EmailAppDemo.json. Once you have downloaded it, you have to load it to your Azure Language Resource. To do so, you have to make a POST call to the following URL:

```

https://{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/
:import?api-version=2023-04-01

```

Be sure to include your Resource API key in the request header as usual and pass the JSON file downloaded from the provided URL in the request body. Upon a successful request, the API response will contain an operation-location header with a URL you can use to check the status of the import job.

```

https://{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/
:import/jobs/{JOB-ID}?api-version=2023-04-01

```

Through a GET API call to this URL you can get the import operation status. If this succeeded you can now train and then use the CLU service.

Typically, after you create a project, you should build schema and label utterance. For this example, the project you imported already had schema and labeled utterances built in. To start a training job, make a POST API call to the following URL:

```

https://{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/
:train?api-version=2023-04-01

```

The body of the request should be a JSON formatted as follows:

```

{
  "modelLabel": "{MODEL-NAME}",
  "trainingMode": "{TRAINING-MODE}",
  "trainingConfigVersion": "2022-09-01",
  "evaluationOptions": {
    "kind": "percentage",
    "testingSplitPercentage": 20,
    "trainingSplitPercentage": 80
  }
}

```

Supported modes for {TRAINING-MODE} are Standard training, which is faster but only available for English, and Advanced training, which is supported for other languages and multilingual projects but involves longer training times. For this example you can set {MODEL-NAME} to Model1 and {TRAINING-MODE} to standard.

Note TestingSplitPercentage and TrainingSplitPercentage

In the JSON payload, the `testingSplitPercentage` and `trainingSplitPercentage` parameters are required only if the `kind` parameter is set to `percentage`. The other possible value for `kind` is `manual`.

As for the importing job, upon a successful request, the API response will contain an operation-location header with a URL you can use to check the status of the training job, through a GET API call. Once the status in the response of this API call is “succeeded” you can proceed to the model deployment.

To deploy a model you should make a PUT call to

```

https://{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}
/deployments/{DEPLOYMENT-NAME}?api-version=2023-04-01

```

Pass your Resource ApiKey in the header, and pass a JSON in the body of the request as follows:

```

{
  "trainedModelLabel": "{MODEL-NAME}",
}

```

Upon a successful request, the API response will contain an operation-location header with a URL you can use to check the status of the deployment job, through a GET API call. Once the status in the response of this API call is “succeeded” you can proceed querying the model.

To query the model, make a POST call to:

```

https://{ENDPOINT}/language/:analyze-conversations?api-version=2023-04-01

```

Format the body of the request like the following JSON:

```

{
  "kind": "Conversation",
  "analysisInput": {
    "conversationItem": {
      "id": "1",
      "participantId": "1",
      "text": "Read Matt's email"
    }
  },
  "parameters": {
    "projectName": "{PROJECT-NAME}",
    "deploymentName": "{DEPLOYMENT-NAME}",
    "stringIndexType": "TextElement_V8"
  }
}

```

The response will be formatted as follows:

```

{
  "kind": "ConversationResult",
  "result": {
    "query": "Read Matt's email",
    "prediction": {
      "topIntent": "ReadAloud",
      "projectKind": "Conversation",
      "intents": [
        {
          "category": "ReadAloud",
          "confidenceScore": 0.8933616
        },
        {
          "category": "SearchMessages",
          "confidenceScore": 0.7128862
        }
      ],
      "entities": [
        {
          "category": "SenderName",
          "text": "Matt",
          "offset": 5,
          "length": 4,
          "confidenceScore": 1
        }
      ]
    }
  }
}

```

When you send a query to your Conversations (CLU) project via the Runtime API, the response provides a structured prediction that includes both

intents and entities recognized in the input text.

Each intent and entity is accompanied by a confidence score between 0.0 and 1.0, indicating the model's certainty in its prediction. The top-scoring intent, the one the model is most confident about, is returned separately in its own field for easier access.

Only entities that are successfully recognized in the input text will appear in the response. For each predicted entity, the API provides:

- The text of the entity that was extracted
- Its offset (the position where the entity begins in the input string)
- The length of the entity (how many characters it spans)

This structure allows you to understand not only what the model identified in the user's input, but also how confidently it made each prediction and where each piece of information was located in the original utterance.

Use cases for the Azure AI Language service

I have implemented solutions for call centers for much of my career using a variety of tools. Call centers require feedback as they need to measure and improve customer satisfaction and evaluate call center agent performance.

Many call centers use feedback tools such as post-call surveys and listening to calls manually, which can be both inaccurate and time consuming. By using the Azure AI Language service in combination with the Azure AI Speech service (more on this in the next section), you can help staff and managers to gain statistics and insights into telephone calls in several ways:

- Speech to Text transcribes call recordings to text.
- Voice Recognition can be used to break the text into who is speaking in turn.
- Sentiment Analysis can analyze the sentiment of each interaction. You can then track:
 - How the customer sentiment changes over the lifetime of the call.
 - If the agent can convert negative exchanges into positive conversations.
- Key Phrase Extraction can extract the main talking points in the conversation. This can be used to categorize the call.
- Named Entity Recognition can extract entities such as people's names, company names, locations, dates, and personal information. This data can enhance the data held for the customer.

Another use case for Language services is in handling compliance. You can scan the emails and call recordings made by your sales team to automate compliance checking by scanning for mentions of key phrases or named entities that represent your products and services.

Identify the capabilities of the Azure AI Speech service

Speech recognition has long been a goal for AI. Now, thanks to the Azure AI Speech service, you have access to a set of easy-to-consume APIs you can use to incorporate speech recognition in your applications. These APIs are the same ones that Microsoft uses in its own products, such as the live captions and subtitles in PowerPoint.

Modern speech recognition technology relies on machine-learned statistical models that leverage cloud computing combined with vast quantities of sample audio. The Microsoft speech recognition technology handles dialects and accents of different speakers; it can deal with jargon, and it performs well in noisy conditions.

The Speech service contains an acoustic model and a language model for many languages. The acoustic model in the Speech service is a deep neural network trained on thousands of hours of audio using advanced algorithms. The number of languages varies by operation and is increasing all the time.

To deploy the service, specify Speech when creating a new resource in the Azure portal. You must select your region, select the resource group, provide a unique name, and select the pricing tier: Free F0 or Standard S0.

You can create Speech resources using the portal, or you can use the CLI, as follows:

```
az cognitiveservices account create --name <unique name> --resource-group <resource group name> --kind SpeechServices --sku F0 --location <region>
```

The Speech service has several speech-processing operations.

Speech to Text

The *Speech to Text* API detects and transcribes spoken input into text. It can transcribe audio into text in real time or from a recording. It converts fragments of sound into text using the acoustic model, and then uses the language model to create words and phrases. Speech to Text converts audio from a range of sources, including microphones, audio files, and Azure Blob storage.

The Speech to Text API can be used synchronously (real time) or asynchronously (batch). Actually, the Speech service includes two separate APIs: one for short audio (up to 60 seconds) that you can transcribe in real time, and the other for batch transcription. The batch Speech to Text API can translate large volumes of speech audio recordings stored in Azure Blob Storage.

More than 100 languages and variants are supported.

Text to Speech

Converting written text to spoken audio, text-to-speech capabilities are useful when you cannot look at a screen, such as when you are controlling other equipment or are using a mobile device while driving.

The *Text to Speech* API converts text into synthesized speech. You can choose from the provided neural voices or create a custom neural voice. You can also create your own custom voice for use in speech synthesis. There are over 400 voices available, and the Text to Speech feature supports over 140 languages and variants. OpenAI voices are also available through this feature of the Speech service in North Central US and Sweden Central.

Note Custom Neural Voice

Creating a custom neural voice requires an approval process by Microsoft to ensure responsible and ethical use of synthetic speech technology.

Speech Translation

You could achieve translation of speech yourself using a mixture of the Speech to Text, Translator, and Text to Speech features, but the *Speech Translation* feature simplifies this process for you. This API first detects and transcribes speech into text, then it processes the text to make translation easier. The text is then fed to the text translation service to convert to the target language. Finally, the translated text is synthesized into audio.

Speech Translation converts audio into text and translates into another language in real time or in a batch. Speech Translation can translate audio into more than 100 languages. Speech Translation performs both speech-to-text and speech-to-speech translations.

Speaker Recognition

The *Speaker Recognition* feature identifies speakers from their voice characteristics in an audio clip, answering the question: “Who is speaking?”

To be identified, speakers must be enrolled using sample audio recordings of their voice. The Speaker Recognition API extracts the characteristics of the voice to form a profile. This profile is used to verify that the speaker is the same person (speaker verification) or to identify the speakers in a conversation (speaker diarization). Speaker diarization can be used to assist in creating transcripts of conversations.

Use cases for the Azure AI Speech service

Mobile phone apps are a common use case for speech-related services. Many apps now use speech to text, text to speech, and speaker recognition. Apps such as Microsoft Translator enable you to translate speech across two languages on your mobile device.

Personal digital assistants, such as Cortana, use the Speech service to receive instructions and questions as audio through the microphone and respond with audio. You will find speech services in home automation, as well.

My banking app uses my voice to identify me. I had to enroll my voice in the app by saying the statement “My identity is secure because my voice is my passport, verify me” three times. I can now access my banking details by repeating the same phrase.

Telephony data that is generated through landlines and mobile phones is typically low quality, which creates challenges when converting speech into text. Recent improvements in the Speech service have significantly increased the accuracy of transcribing telephone call recordings.

Identify the capabilities of the Azure AI Translator service

The *Translator* service translates text from one source language (from) into multiple other (to) languages. The Translator service allows you to specify multiple “to” languages, so you can simultaneously translate into multiple languages.

The Translator service uses *neural machine translation* (NMT), which uses neural networks and deep learning to translate whole sentences.

The Translator service can translate text in real time or in batches across 90 languages, variants, and dialects.

Create a Translator resource

Translator is a nonregional service, and you should create your resource in the Global region, unless your application requires a specific region.

To deploy the Translator service, specify Translator when creating a new resource in the Azure portal. You must select your region, select the resource group, provide a unique name, and select the pricing tier: Free F0 or tiers with volume discounts.

You can create a Translator resource using the CLI as follows:

```
az cognitiveervices account create --name <unique name> --resource-group <resource
group name> --kind Translator --sku F0 --location Global
```

Use the Translator service

Translator is a modern JSON-based web API with several features:

- **Language** Returns the list of languages supported by the service
- **Detect** Identifies the language of the source text
- **Translate** Converts text from one language into multiple languages
- **Transliteration** Converts text from one language into another language and changes the script/character set
- **BreakSentence** Identifies the position of the sentences in the text
- **Dictionary Lookup** Provides alternative translations of a word or phrase
- **Dictionary Example** Provides examples of how a word or phrase is used

Formulate the request URL for the translate operation as follows:

```
https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&to=<language
code>
```

The Translator service supports translation into language variants such as French and Canadian French. The Translator service allows you to specify a cultural variant when translating into a language. You append the cultural code to the language code—for example, *fr-CA* for Canadian French—to the URL request.

You must include a valid subscription key and region (unless your resource is global, in which case you only have to specify your subscription key) in the request headers, just as shown in the previous API examples in this chapter. These headers are:

```
Ocp-Apim-Subscription-Key: <your-translator-key>
Ocp-Apim-Subscription-Region: <your-region>
```

The subscription key must belong to a resource of type Translator. If you do not yet have a Translator resource, you can create one in the Azure portal by following these steps:

1. Navigate to the resource group where you want to deploy the Translator service.
2. Select Create, then search for "Translator" in the Azure Marketplace.
3. Choose the Translator service from the list of results.
4. Specify the required details:
 - **Resource name**
 - **Region (this must match the region you use in the header)**
 - **Pricing tier (F0 for Free or S1 for Standard)**

After the resource is created, you can find both the subscription key and endpoint URL in the Keys and Endpoint section of the Translator resource.

Following is the JSON returned for a paragraph at the beginning of this section translated from English (en) to Spanish (es):

```
[[{"detectedLanguage": {"language": "en", "score": 1.0},
  "translations": [{"to": "es", "text": "El servicio Traductor puede traducir simult\u00e1neamente de un idioma a varios otros \u219d idiomas. El servicio Traductor traduce texto de un idioma a otro idioma. El servicio Traductor le permite \u219d especificar varios idiomas, para que pueda traducir simult\u00e1neamente a varios idiomas."}]]]
```

There are limits in the Translator API:

- **Translate operation** Limited to 50,000 characters, including spaces
- **Transliterate operation** Limited to 5,000 characters
- **Detect operation** Limited to 50,000 characters
- **Dictionary Lookup operation** Limited to 1,000 characters
- **Dictionary Examples operation** Limited to 2,000 characters
- **BreakSentence operation** Limited to 50,000 characters

Use cases for the Azure AI Translator service

There are use cases for translation support, such as translating speech and text in multi-country call centers.

Language translation apps for mobile devices make life easier for travelers. Similarly, you could create a multi-language website using Translator or build an app that supports multiple languages.

Microsoft PowerPoint provides real-time translation and subtitles for presentations. The Live Presentations feature allows audience members to see the presentation on their mobile devices with subtitles generated from your speech translated into their languages.

Chapter summary

In this chapter, you learned some of the general concepts related to natural language processing. You learned about the general features of natural language processing and about the specific services in Azure AI Services related to language and speech. Here are the key concepts from this chapter:

- Natural language processing is about extracting information and insights from both speech and text.
- NLP uses statistical analysis and other processes to extract meaning from text.
- NLP uses a language model to understand the text.
- Key phrase extraction finds the main topics in the text.
- Named entity recognition identifies known entities (people, places, and things) in the text.
- Sentiment analysis classifies the emotion in the text as positive, negative, neutral, and mixed.
- Speech recognition detects and interprets audio and converts it to text.
- Speech synthesis generates spoken audio from text.
- Translation converts text from one language into another language.
- The Language service is an Azure AI Service that performs a series of natural language processing operations on text, including detection of language, sentiment analysis, key phrase extraction, and named entity recognition.
- An intent is the action that the user wants to perform.
- An utterance is an example phrase that a user says. You need to supply sample utterances to train your model.
- Entities are the subjects or context for the intent.
- There are prebuilt entities for common objects, such as dates, places, and personally identifiable information (PII). You can add your own custom entities for your business domain.
- Speech to Text transcribes audio into text in real time or from audio files.

- Speech to Text can transcribe large quantities of audio recordings stored in Azure Blob Storage.
- Text to Speech synthesizes text into spoken audio.
- You can create a custom voice for conversion of text into synthesized speech, but this feature requires prior approval from Microsoft. You must submit a request and be granted access before you can use Custom Neural Voice capabilities.
- Speech Translation converts audio into text and translates into another language in real time.
- Speaker Recognition identifies enrolled people from their voices in an audio clip.
- Translator translates text in real time or in batches.
- Translator supports translation into many languages and variants.

Thought experiment

To apply what you have learned in this chapter and demonstrate your knowledge of its topics, try this thought experiment. You can find the answers in the section that follows.

You work for Litware, Inc., a company with several brands and that supplies business-to-business services across the world. Litware is interested in analyzing the large amount of text involved in their business using AI.

Litware wants to evaluate how cognitive services, such as Azure AI Services, can improve their internal document categorization.

Litware wants to create a single support desk to handle their worldwide customer base. This central desk will provide consistent responses to customers no matter their location or language.

Litware needs to understand how customers will respond to this move to a single support desk. Customers will receive a questionnaire asking them about this move. The questionnaire includes a series of questions, a space for the customer to write their thoughts, and the option for customers to request more information or be contacted by a Litware representative.

As part of this planned move, Litware monitors social media for mentions about these proposed changes and records telephone calls into the existing support desks.

Answer the following questions:

1. Which workload should you use to evaluate how the customer feels about the move to a central support desk?
2. Which workload should you use to discover the topics mentioned by customers in the questionnaire?
3. Named entity recognition extracts the intent and action from the request in the questionnaire. Is this correct?
4. Which workload should you use to monitor social media for negative mentions of Litware's brands?
5. Which workload should you use to transcribe telephone calls into the support desk?
6. Which Azure AI Service should you use to mine customer perceptions of Litware's products and services?
7. Which service should you use to translate the large volumes of telephone calls?
8. Do you have to use a standard voice, or can you create a custom voice for text to speech?

Thought experiment answers

This section contains the solutions to the thought experiment. Each answer explains why the answer choice is correct.

1. The emotion expressed in text is an example of the sentiment analysis workload. Sentiment analysis evaluates a piece of text and determines if the text has a positive or negative feeling/emotion.
2. The topics, or main talking points, contained in text are an example of the key phrase extraction workload. Key phrase extraction evaluates the text and identifies the key talking points in the text.
3. No, the intent and actions expressed in text are an example of conversational language understanding and not named entity recognition.
4. Analyzing social media for a brand is an example of when you would use sentiment analysis to determine the positive and negative mentions of a brand.
5. Transcription of a recording into text is an example of speech recognition. Speech recognition can convert audio into text.
6. Mining customer opinions can be performed by the Language service using the Sentiment Analysis operation. Sentiment Analysis explores customers' perceptions of products or services.
7. The Speech to Text API can be used synchronously (real time) or asynchronously (batch). Batch Speech to Text can translate large volumes of speech audio recordings stored in Azure Blob Storage.
8. You can choose from neural voices, standard voices, or a custom voice. The Text to Speech API can create custom voices.

Describe features of generative AI workloads on Azure

Organizations need to be contactable by their customers over multiple communications channels: email, web chat, telephone, chat, and a range of social media platforms, such as Teams, Reddit, Instagram, and Facebook. In return, organizations need to respond consistently in the same manner no matter which channel a customer decides to use.

Plus, customers are becoming more demanding. They require 24/7 responses, responses that are tailored to their behaviors and characteristics, and responses to more and more complex requests.

While *conversational AI* is the use of AI-powered agents, known as *bots*, to respond to human questions and requests in an intelligent way, generative AI extends beyond conversational bots and enables organizations to automatically generate high-quality content such as emails, reports, summaries, images, or even code. Organizations use generative AI solutions, such as conversational agents, to provide the first line of response to customers, handling common inquiries and content generation across multiple channels.

These AI agents not only initiate intelligent, natural-language interactions—whether over the web, SMS, or social platforms—but can also be integrated with agentic capabilities. They can take actions, call APIs, retrieve data, and autonomously complete tasks on behalf of the user, enabling more dynamic and personalized customer experiences.

This chapter explains the generative AI capabilities provided by Microsoft Azure. I will outline the concepts involved, describe use cases, and then explain how to use the Azure services involved.

Skills covered in this chapter:

- Skill 5.1: Identify features of generative AI solutions
- Skill 5.2: Identify Azure services and capabilities for generative AI

Skill 5.1: Identify features of generative AI solutions

When people interact with a computer application using text and speech, the person should not be expected to imitate the application’s internal instructions; this only causes irritation and ultimately leads the person to abandon using the app. A computer application must be able to handle language provided in a natural way and be able to react accordingly to the meaning that the human has expressed. Input and output can be multimodal; it can be text, speech, images, or other modalities. *Conversational AI* bridges the gap between human and computer language; it allows humans and computer applications to work together in a natural way.

Generative AI allows computers to recognize human inputs, interpret different languages or formats (audio, images, videos, and more), understand what is being said, and respond in a manner that mimics a human conversation. These models can generate coherent text, create realistic images, compose music, write code, and engage in meaningful conversations.

Generative AI is not limited to chatbots, although conversational agents are one of the most recognizable applications. Other use cases include document summarization, image generation, translation, code completion, and automated report writing.

Figure 5-1 shows how a chatbot works. A human asks the question, “How can I book an exam?”

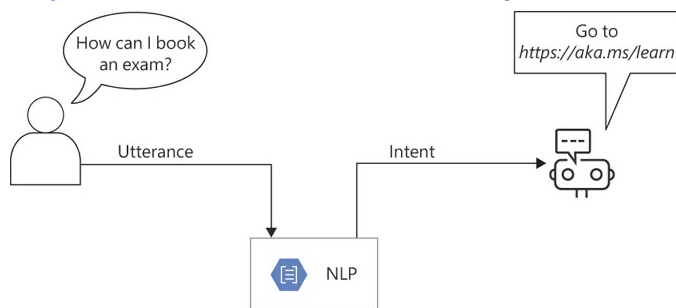


FIGURE 5-1 Chatbot diagram

This question is processed using a language model. Depending on the scenario, it can also be integrated with external APIs or data sources to personalize or complete tasks autonomously.

A good definition of generative AI is a class of AI models that can produce new content (text, images, audio, or code) based on patterns learned from existing data. This dialogue can occur over many different communication channels, such as web, social media, email, or voice.

This skill covers how to:

- Identify features of generative AI models
- Identify common scenarios for generative AI
- Identify Responsible AI considerations for generative AI

Identify features of generative AI models

One of the most common applications of generative AI is through web-based conversational experiences, but the capabilities go far beyond static chat.

A typical use of generative AI is in the development of intelligent virtual agents that simulate human conversation and perform tasks across different modalities. These systems often include a text input box for typed questions and, in some cases, a microphone icon for voice input. [Figure 5-2](#) shows a generative AI-powered chat interface.

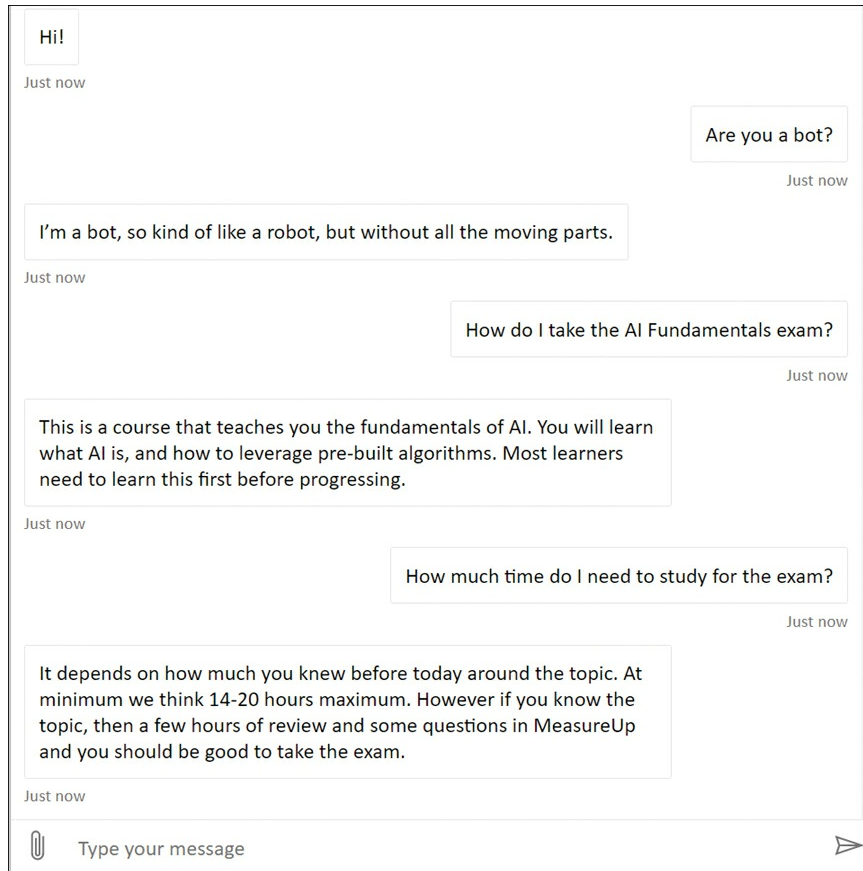


FIGURE 5-2 Chatbot example

Unlike traditional bots that follow scripts or predefined intents, generative AI models use large-scale neural networks to generate responses in real time. These responses are not limited to fixed answers but are tailored to the user's prompt, history, and sometimes external data sources.

Common features of generative AI models include:

- Natural language understanding and generation
- Multiturn conversation handling
- Creating images from text prompts (such as for design, advertising, education, and more)
- Generating diagrams, summaries, or flowcharts from descriptions
- Captioning or interpreting images using vision-language models
- Tool integration (calling APIs, retrieving files, running functions, and more)
- Transcribing and summarizing audio files
- Generating voiceovers or synthetic speech from text

Those features can combine two or more input/output types. For example, a generative AI system could accept a screenshot of an error message and respond with troubleshooting steps in natural language. Others may receive a voice query and return an image or a generated document as output.

A different daily example of a generative AI application is a new feature from search engines: AI-generated summaries for user search queries ([Figure 5-3](#)).

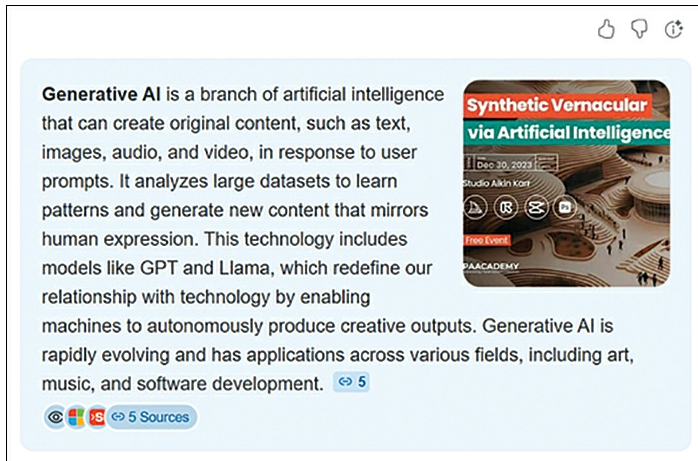


FIGURE 5-3 AI-generated summary by Bing

Note What is Prompt Engineering?

Prompt engineering is the practice of designing and refining input prompts to guide generative AI systems toward producing accurate, relevant, or creative outputs. Because generative AI models respond based on the input they receive, a well-crafted prompt can significantly influence the quality and usefulness of the response. Prompt engineering may involve specifying format, tone, context, or constraints, and is especially important when using models for complex tasks.

Identify common scenarios for generative AI

Generative AI powers a wide range of modern applications across industries, from productivity and customer service to education, healthcare, and creative design. These scenarios go beyond simple question-and-answer bots, leveraging large language models and multimodal capabilities to solve complex tasks, assist users, and create content dynamically.

Interactive voice response (IVR) on telephone systems is another area where you might find conversational AI being used.

A virtual assistant responds to a command by taking action—for example, creating a booking for you in your calendar. An early representative of a personal digital assistant was Cortana (Figure 5-5), although it refers to itself as a personal productivity assistant.

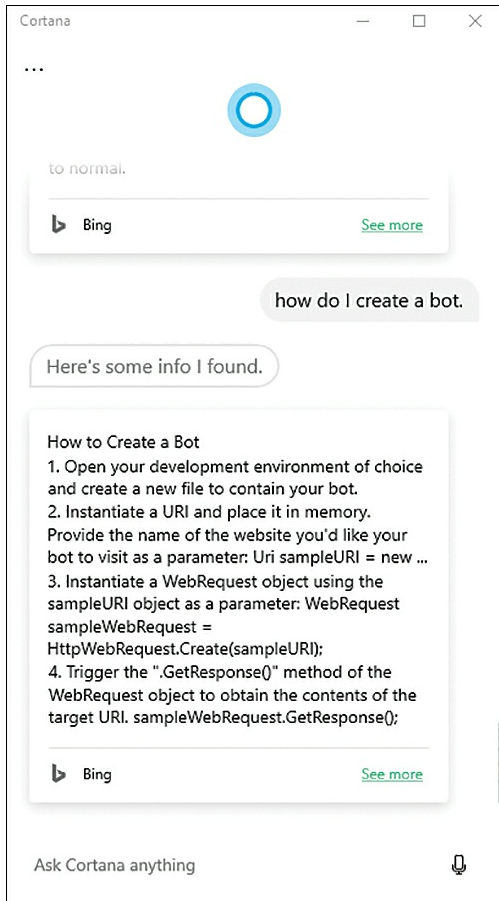


FIGURE 5-4 Cortana assistant answering a question

Cortana relied on speech recognition and simple command execution. Though helpful at the time, it lacked the generative capabilities and contextual understanding of today's AI systems. Microsoft retired Cortana in 2023, and newer tools have since replaced it with far more advanced capabilities.

Modern conversational AI solutions have vastly expanded these capabilities. For example:

- Microsoft Copilot, integrated into Windows 11 and Microsoft 365, uses generative AI to create content, summarize information, and control system settings via natural language, surpassing Cortana's static commands (Figure 5-5).

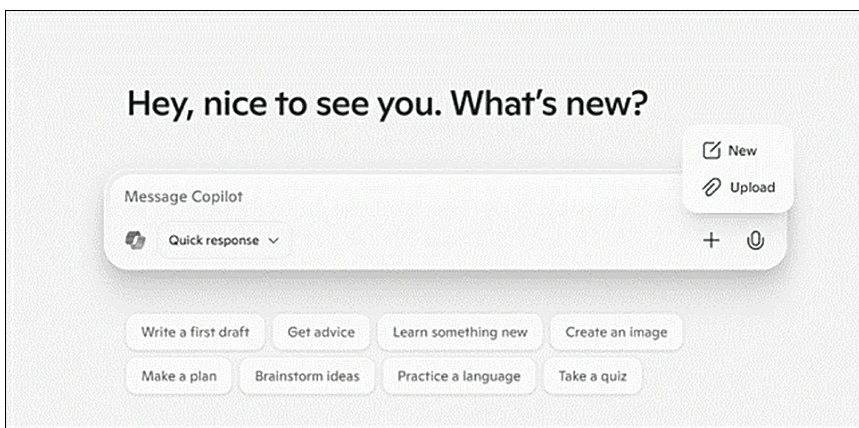


FIGURE 5-5 Microsoft Copilot homepage

- OpenAI's ChatGPT, with plug-ins and voice mode, can handle complex multidomain tasks, from booking appointments to writing emails, illustrating true conversational flexibility.
- Google Assistant and Amazon Alexa offer deep integration with devices and third-party services, enabling proactive and personalized assistance in everyday life.
- Adobe Firefly allows designers to create logos and images from text prompts.

- Tools like GitHub Copilot or Cursor AI suggest code, complete functions, and even explain errors in real time.

This progression shows how early AI bots laid the groundwork, while today's generative AI applications provide richer, more interactive, and action-driven experiences that better meet user expectations and business goals.



Exam Tip

You need to be able to identify the use cases for generative AI solutions or where applications that leverage generative AI can be used to meet the scenario described.

Identify Responsible AI considerations for generative AI

When building solutions with generative AI, it is important that you remember the guiding principles for Responsible AI. You can find a thorough discussion of all six in [Chapter 1](#), “[Describe artificial intelligence workloads and considerations](#),” but the main principles to keep in mind are:

- **Transparency** Users should be clearly informed when they are interacting with an AI system. This includes disclosing the AI's capabilities and limitations, and providing options to escalate interactions to human agents when necessary.
- **Fairness** Generative AI models must be designed to treat all users equitably, avoiding biases related to race, gender, age, or other characteristics. This involves training models on diverse datasets and continuously monitoring outputs to detect and mitigate any discriminatory patterns.
- **Reliability and Safety** Ensuring that generative AI systems produce accurate and safe outputs is critical. This includes implementing measures to prevent the generation of harmful, misleading, or inappropriate content. Techniques such as content filtering, prompt engineering, and grounding responses in factual data can enhance the reliability of AI outputs.
- **Privacy and Data Protection** Generative AI systems often process vast amounts of data, some of which may be sensitive. It's vital to implement robust data protection measures, ensuring compliance with privacy regulations and safeguarding user information. This includes data anonymization, secure storage, and obtaining user consent for data usage.
- **Accountability and Human Oversight** Developers and organizations deploying generative AI must be accountable for their systems' behavior. Establishing clear governance structures, documenting decision-making processes, and enabling human oversight are essential components of responsible AI deployment. In scenarios where AI systems make significant decisions, mechanisms should be in place for human review and intervention.

Skill 5.2: Identify Azure services and capabilities for generative AI

As generative AI continues to evolve, Microsoft Azure provides a suite of powerful, enterprise-ready tools to support the development, deployment, and scaling of intelligent agents. This section explores key Azure services that enable organizations to build advanced conversational and generative experiences with minimal complexity and maximum flexibility.

Azure AI Foundry shifts AI development into hyperdrive by grouping every model, dataset, and all of your cost and usage information all back to a single project, thus removing some of the complexity of having to manage dispersed AI materials from dispersed services and managing costs. Azure AI Foundry enables you to move easily from prototype to production without switching to another environment or learning another workflow.

The model catalog in Azure AI Foundry is helpful in model discovery; it offers one-click deployment, which enables you to compare performance (across Microsoft, partner, and open-source models) side by side. You can experiment with models from Hugging Face, Meta, and other top AI labs without dealing with the standard integration overhead.

Azure OpenAI brings the power of GPT, DALL·E, TTS, and Whisper behind the Microsoft enterprise security umbrella, offering something unique in the market. You are working with the same models that power ChatGPT, but your data never leaves your Azure tenant. The Azure OpenAI Service offers pay-per-use options and dedicated compute instances that guarantee dedicated capacity and private networking.

This skill covers how to:

- [Describe features and capabilities of Azure AI Foundry](#)
- [Describe features and capabilities of the Azure AI Foundry model catalog](#)
- [Describe features and capabilities of Azure OpenAI](#)

Describe features and capabilities of Azure AI Foundry

Azure AI Foundry is a single workspace for designing, extending, and governing generative AI applications and agents. Azure AI Foundry brings together the fragmented toolset of old solutions (AI Studio, other SDKs, manual deployment) with a single project-centric workspace that integrates model discovery, data connections, cost/timelines, and production-grade endpoints in a single view.

Azure AI Foundry is designed for developers to:

- Build generative AI applications on an enterprise-grade platform
- Explore, build, test, and deploy using cutting-edge AI tools and ML models, grounded in responsible AI practices
- Collaborate with a team for the full life-cycle of application development

Foundry projects

In Azure AI Foundry everything happens inside a *project*, a secure workspace that ties together AI models, datasets, vector indexes, and cost dashboards. You can work on your project directly in the Azure AI Foundry web page or use the SDK from your development tool of choice. After you create a project, you can reference it from your code.

Note Create an Azure AI Foundry Hub

To create an Azure AI Foundry hub, or have one created for you, you must have permissions to do so. You must be an Azure AI Developer, Contributor, or Owner on the hub in order to complete this tutorial. If your role is Contributor or Owner, you can directly proceed to follow the steps in the tutorial. If your role is Azure AI Developer, you need an existing hub to complete this tutorial.

Here are some of the main capabilities that Foundry provides with projects:

- **Model catalog** Think of a large library of AI models. That is what the model catalog is: a library of hundreds of proprietary and open-source models that you can browse through. When looking for options for your project, you can explore each model card, complete with performance benchmarks, licensing (commercial or open-source), and sample code to expedite your development. If you are trying to choose from different models, it also has a Compare feature, so that you can set leaderboard scores next to each other, so that you can choose the best one for your purpose.
- **Retrieval-Augmented Generation (RAG)** When large language models (LLMs) like GPT don't have access to your specific data, their answers can be incomplete or inaccurate. Retrieval-Augmented Generation (RAG) solves this by combining the model's language abilities with access to external data sources. This means the model can "look up" relevant information before generating a response. Foundry offers three generic flavors: a quick inportal vector store for small file sets, a complete Azure AI Search index (keyword + semantic + vector modes) that supports enterprise-size collections, and on-demand Bing grounding for fallback for any public-web queries. Choose what you need; the service manages chunking, retrieval, and prompt injection behind the scenes.
- **AI Foundry Agent Service** While most AI tools generate responses one message at a time, AI agents are designed to handle more complex tasks that require planning and multiple steps. An agent can decide what tools to use, what steps to follow, and how to respond based on an evolving conversation or task. The AI Foundry Agent Service in Azure builds on existing models to give them these planning and decision-making capabilities. It enables the use of tools (like search or file retrieval) during interactions and manages longer, multistep tasks, such as booking travel or troubleshooting a technical issue, from start to finish.
- **Fine-tuning** Pretrained AI models are general-purpose, they've learned from vast amounts of public data. But sometimes, you need the model to respond in a specific tone, follow your company guidelines, or understand industry-specific terminology. That's where fine-tuning comes in. Fine-tuning allows you to adjust the model's behavior by training it on your own example data. In Azure AI Foundry, for models that support customization, you can adjust models to suit your use case. The fine-tuning process is simple enough: Upload a JSONL file, select between serverless or managed compute, and you can rely on Foundry to keep track of versioning automatically. If you're interested in methods that utilize fewer resources to adapt models, LoRA (low-rank adaptation) adapters have been added to public preview and provide a lightweight alternative to full model fine-tuning.

There are two deployment options:

- **Serverless inference** Zero idle cost, pay per token
- **Managed compute** Reserved GPU, fixed SLA (service level agreement), VNet, and Private Link support. Both paths expose an HTTPS endpoint wired to Content Safety logs.

Note Deprecated Models

A model can be deprecated when newer checkpoints in the same family ship. Foundry will display the retirement date on the model card and, where available, allow you to toggle automatic updates so your deployments roll forward without a manual step.

All language models deployed via Standard or Managed Compute are fronted by Azure AI Content Safety for hate, sexual, self-harm, and violence categories. You can choose to disable filtering per-endpoint, but you will be fully responsible for harmful outputs.

Setting up a project is very easy, as you can see in [Figure 5-6](#). You simply go to ai.azure.com and select Create Project. The Create a Project dialog box that opens in Azure AI Foundry prompts you to enter a project name and select a hub, which is the collaboration environment for your team to exchange your project work. Your options include creating a new hub or selecting from existing ones.

Create a project

Projects are easy-to-manage containers for your work—and the key to collaboration, organization, and connecting data and other services.

Project name * ⓘ

Hub ⓘ [Create new hub](#)

Create **Cancel**

FIGURE 5-6 Create an Azure Foundry project

Once you create your Foundry project, you can see the project dashboard, which has everything you need to manage and personalize your solution (see Figure 5-7):

- **General** When you open your project, you land on Overview, where you grab keys, check on usage, and jump to the model catalog, playgrounds, or quick AI Services tools. It's the launch-pad for what you want to try next.
- **Build & Customize** This block is where you create things: launch Agents for multistep logic, clone already-made templates, run fine-tuning if your prompts are not enough, use Content Understanding to get the structure from docs or media, and connect the dots with Prompt Flow.
- **Assess & Improve** Turn on Tracing to see every call, use Evaluation to bulk-test and compare versions, and lock-down policies in Safety + Security to manage costs and compliance.
- **My Assets** A single shelf that shows every model, vector index, and web app you've deployed, so you can easily tidy up or track spend.

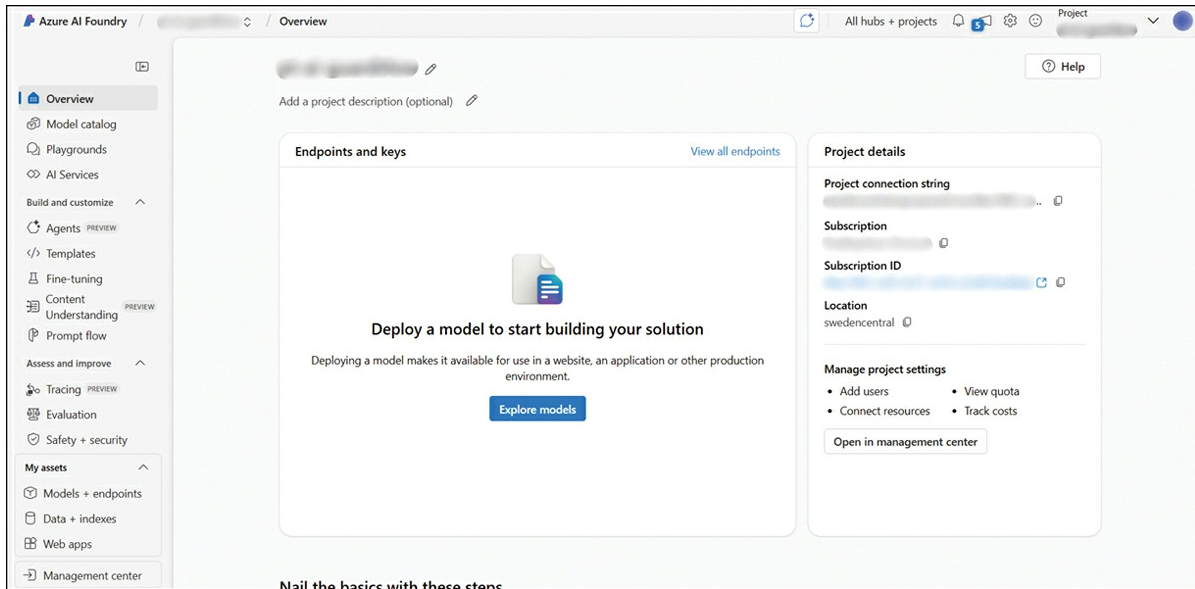


FIGURE 5-7 Overview page of a Foundry project

Note Project Type Selection

Be sure to choose the correct project type: If you create an OpenAI project, the environment exposes only the Azure OpenAI capabilities, models, and Assistants—not the full feature set of AI Foundry Agents Service.

To deploy a model, you can either click Explore Models or select Model Catalog (left) and then the appropriate model from the catalog.

After you select the model, give the deployment a name and select the deployment type. Below these settings, you will see the deployment details: the model version, region, AI resource it's attached to, and tokens-per-minute (TPM) (Figure 5-8). Each model has a limit on how many tokens it can work with per minute, and those numbers vary depending on the tier and version. In this example, the model allows up to 50K tokens per minute, effectively defining how much text you can send and receive in one minute. When you are ready, click Deploy, and the model is ready to go.

Deploy gpt-4.1

Deployment name 👁

Deployment type

Global Standard ▼

Global Standard: Pay per API call with the highest rate limits. Learn more about [Global deployment types](#).
Data might be processed globally, outside of the resource's Azure geography, but data storage remains in the AI resource's Azure geography. Learn more about [data residency](#).

▼ **Deployment details** 🔗 Customize

<p>Model version 2025-04-14</p> <p>Capacity 50K tokens per minute (TPM)</p> <p>Content safety DefaultV2</p>	<p>AI resource gpt-4.1-2025-04-14</p> <p>Resource location East US</p> <p>Version upgrade policy Once a new default version is available</p>
--	---

Deploy

Cancel

FIGURE 5-8 Deploy a model in Azure Foundry

Building agents

The Azure AI Foundry Agent Service transforms a generative AI model into an autonomous agent capable of executing complex tasks. Unlike simple chatbots, these agents can reason, plan, interact with tools, and operate within enterprise-grade security frameworks. The service enables end-to-end orchestration, allowing models to function as proactive digital workers rather than reactive assistants.

You can think of an agent as a small team in a box, managed by an orchestration layer. The model provides reasoning and language; the instructions state the goal and guardrails; the tools provide access to knowledge and action: query a vector index, call a REST API, write a row in a database. Whereas a chat assistant waits for a user, an agent moves forward to completion.

To create an agent, first you pick the model, for example GPT-4o. Then you customize the agent with a prompt or a fine-tune run, so it speaks your domain language. The next step is to attach the appropriate tools the agent will need to access information and perform actions. These tools are grouped into two main categories:

- Knowledge tools, such as Azure AI Search, File Search, Bing, and Microsoft Fabric, allow the agent to retrieve facts and context needed to answer questions or support decisions.
- Action tools, including Logic Apps, Azure Functions, OpenAPI integrations, and low-level function calls, enable the agent to perform real-world tasks like sending an email, creating a support ticket, or updating a database record.

You can also explicitly specify which tool the agent should use, or you can leave the selection to the model's discretion, guided by the instructions you provide in the prompt.

The orchestration layer manages retry, thread state, and the wiring of tool outputs. Trust modules provide content filtration, network isolation, and role-based access control (RBAC). Finally, Application Insights captures traces so you can replay every decision.

You have full visibility into every messaging, native agent-to-agent messaging, server-side tool orchestration, built-in content safety, and enterprise identity, which are all available with zero additional infrastructure code. The same GPT-4.x Mistral and Llama deployments that you saw in the catalog work here, both in Standard pay-as-you-go and in Provisioned throughput.

To physically create an agent, select Agents from your Foundry project's Overview page, then click New Agent to enter the Agent Playground (Figure 5-9).

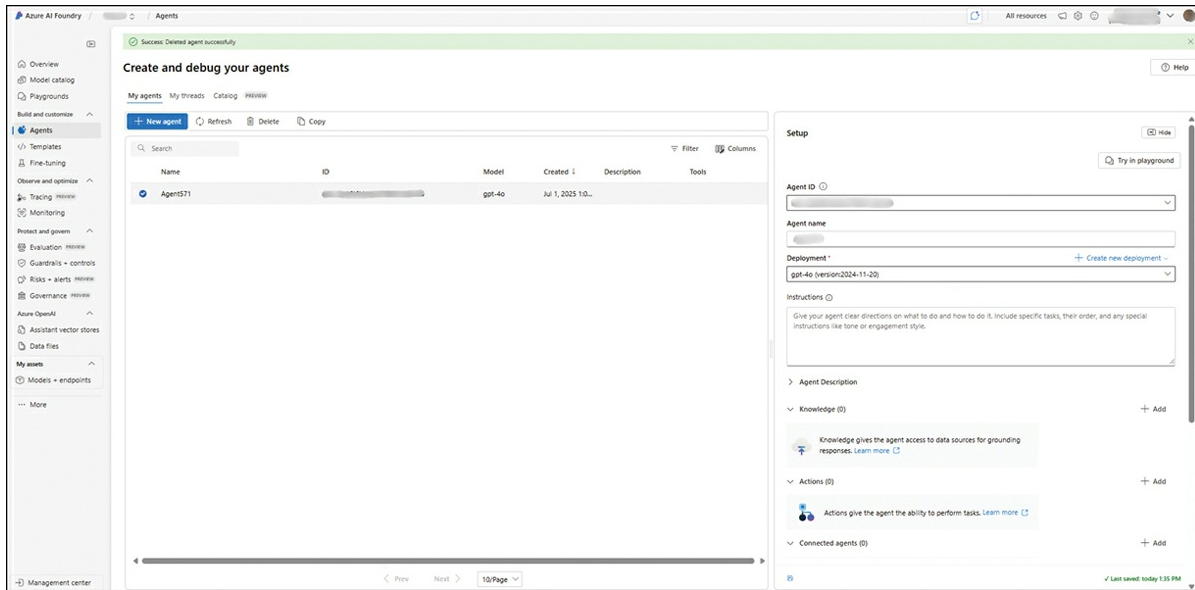


FIGURE 5-9 Creating and configuring an assistant in Azure Foundry

The service gives you an agent ID immediately; the agent’s name is up to you. More importantly, select the appropriate model deployment. Anything you enter in the Instructions field becomes the system prompt, so phrase it as if you’re addressing the model directly, for example: “You’re a support bot for our retail site...”

Below the Agent Description field, turn your attention to the three agents:

- **Knowledge** Exposes the model to context so the agent can ground its responses on that content. Click Add Knowledge to attach a ready-made data connector: anything from local files and Azure AI Search to Microsoft Fabric, SharePoint, Bing-grounded web search, or vertical feeds like Tripadvisor and Morningstar.
- **Actions** Determines what the agent can do while it chats. Select Add Action to open a gallery of built-ins that you can attach with one click (Figure 5-10): OpenAPI 3.0, Azure Logic Apps, and Code Interpreter, which gives the agent a sandboxed Python kernel for on-the-fly calculations, visualizations, and file manipulation.

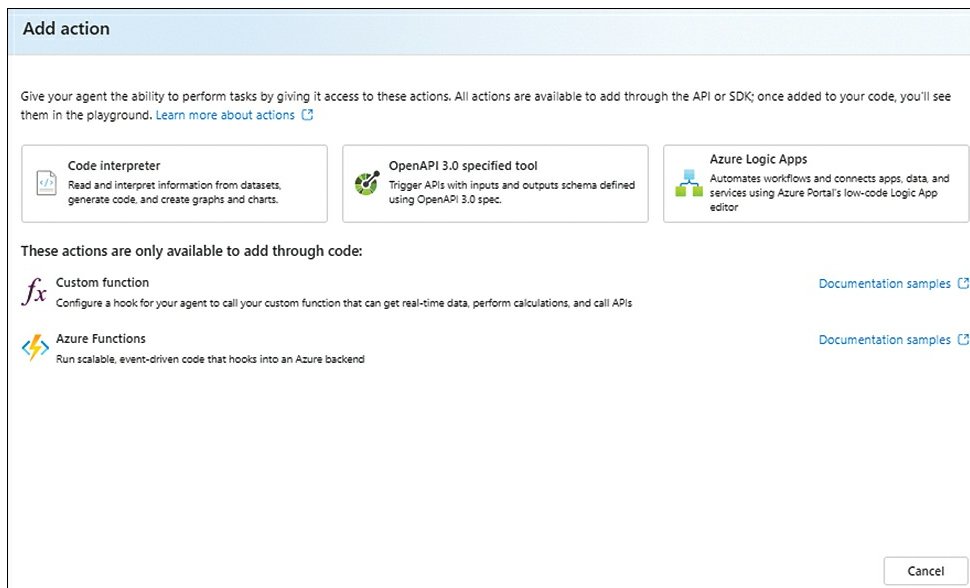


FIGURE 5-10 Add Action window in Azure Foundry

For bespoke integrations, choose Custom Function, paste a JSON schema describing the operation and its parameters, and save; the model decides when to invoke it and returns the arguments so your backend or a Logic Apps flow can run the real work. If you prefer a serverless backend, register the same signature through Azure Functions and the hook executes on the Functions runtime instead.

- **Connected Agents** Lets your primary agent pass the thread (and any accumulated context) to purpose-built sub-agents, making multi-agent workflows possible without writing custom orchestration code.

In the Model Settings section, you’ll find two sliders (Figure 5-11): Temperature raises or lowers the model’s creativity (higher = freer, lower = stricter),

while Top P trims the token pool to a nucleus of most likely words so you can fine-tune response diversity without losing coherence. Adjust them together to steer how adventurous or conservative your agent sounds.

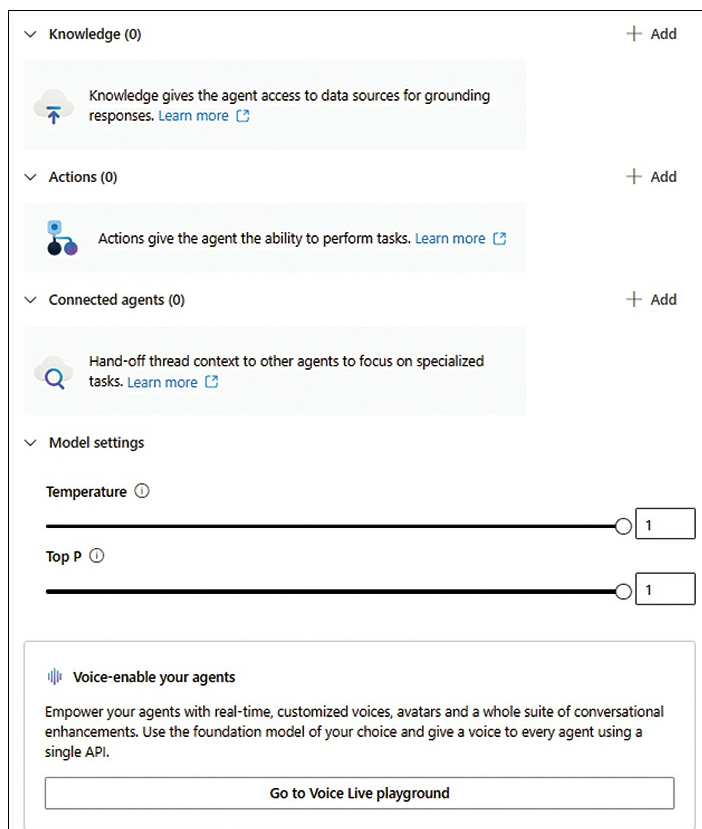


FIGURE 5-11 Model Settings in Azure Foundry

Retrieval and RAG Foundations

If you want a large language model to answer using your own documents instead of the public internet, follow the retrieval augmented generation (RAG) pattern, which is structured as follows: When a user submits a question, a search engine retrieves the most relevant passages to this query from a predefined document collection. These passages, together with the original question, form the prompt that is sent to the language model, the model then produces a grounded answer ensuring that the answer is based on your data rather than general internet content. Figure 5-12 illustrates the RAG process.

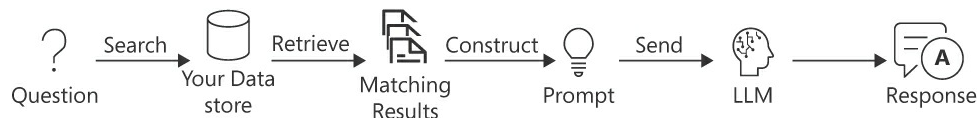


FIGURE 5-12 Simple RAG flow

To make that retrieval fast and cost-efficient, each passage must be stored in an *index*: a well-structured catalog that supports multiple search techniques. This hybrid index combines traditional keyword-based search (inverted index), semantic search, and vector search. It also includes metadata describing how the engine should find the content, which embedding model was used, and which search flags are enabled. In Azure, this catalog is packaged as an Index asset that lives in Azure AI Search, which means your content never leaves your tenant and is given the same compliance considerations as the rest of your workloads, as shown in Figure 5-13.

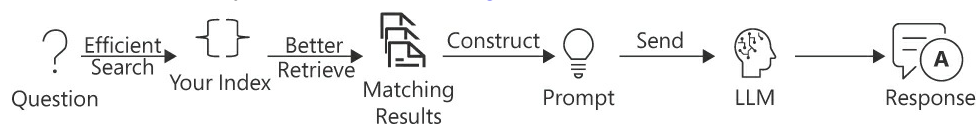


FIGURE 5-13 RAG flow with the index highlighted

Azure AI Search performs two key functions:

- **Indexing** Azure AI Search slices the user’s content, such as PDFs, web pages, or SQL rows, into meaningful chunks. It runs built-in skills like OCR, language detection, and entity recognition. It also generates embeddings and uploads everything into the index in a single operation, meaning no moving files around.
- **Querying** Azure AI Search scans the indexed data at the same endpoint and applies a combination of keyword filtering, vector similarity search, and deep semantic ranking. The keyword filter clause limits the possible results, the vector similarity clause finds the nearest matches, the semantic

ranking clause orders them by meaning and returns a few JSON snippets, just the amount of information the model needs to stay within its context window, and maintains token costs at a reasonable level.

You can control the quality of search results directly within the index by configuring several settings, such as scoring profile, synonym maps, geo-filters, and ranker's weights.

To add search capabilities to your agent, navigate to the Tools section, turn on File Search, select Add Vector Store, then select Create a New Vector Store. Assign a name to the vector store, drag in the files you want the agent to reference, then click Upload and save. The agent now has its own private index and can ground every answer in the provided documents. [Figure 5-14](#) shows the single interface you need to connect the vector store.

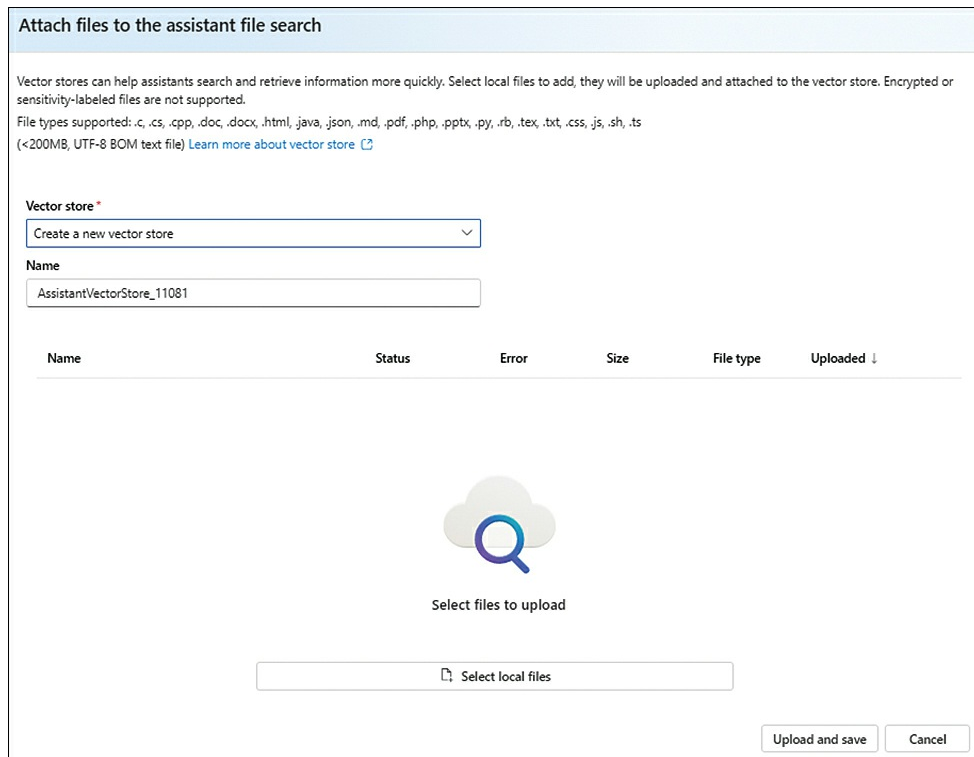


FIGURE 5-14 Add Vector Store in Azure Foundry

Because Search is a first-class Azure resource, you can configure it to operate behind private endpoints, inside VNets, and next to your existing monitoring and backup policies. This means that the net is a single integrated service that transforms raw documents into evidence-based answers without ever moving your proprietary data outside your own cloud boundary.

Fine-tuning models

Fine-tuning occurs at the terminal intersections of prompt engineering and RAG. If the base model still stumbles over edge cases, can't format your output reliably, or is forcing you to load dozens of examples into the prompt, it's time to teach it with data.

The first step is to collect high-quality prompt-and-response pairs—ideally, several hundred. Make sure to consistently label them, partition the pairs into training and validation, and keep the delineation between the end of the prompt and the start of the answer clear.

Whatever model you choose, the workflow is the same:

1. Upload data to the project.
2. Train the new variant.
3. Evaluate it against the base line. Foundry lets you plot loss curves or run the benchmark suite again.
4. Deploy to a standard endpoint, a managed compute VM, or both.
5. Use the endpoint in your application like you would call the original model.

Selecting the base model is a balancing act. Smaller checkpoints generally respond faster and cost less; larger checkpoints solve harder problems, but result in a much larger wallet and GPU quota hit. The leaderboards and playgrounds are designed to help you figure out what is "good enough" before you pay for training.

Fine-tuning is region locked. At the time of this writing, GPT-4.1 variants train in North Central US and Sweden Central only. A Global training preview requires fewer tokens to train (is cheaper), but assumes you do not need a data residency guarantee for a region such as Italy North or Australia East.

Use a tight feedback loop: Watch the loss, save checkpoints every few epochs, and if the validation score plateaus, stop early. Most importantly, start small, validate the gain, then move to larger solutions: Quality data always beats quantity.

Describe features and capabilities of the Azure AI Foundry model catalog

In Azure AI Foundry, the model catalog aggregates over a thousand models for language, vision, audio, and code, categorizing them by domain. Each model's card shows the model name, type (chat completion, speech-to-text), and an icon indicating performance analytics. You can deploy most models in basically two clicks on your GPUs or as pay-per-token serverless APIs.

The catalog combines three worlds into one list:

- **Azure OpenAI models** This collection includes models such as GPT-4, DALL·E, Text-To-Speech, and Codex all supported with the same SLA (service-level agreement) you would receive from the Azure OpenAI Service. (The next section goes into more detail on OpenAI.)
- **Curated by Azure AI** This group features partner models developed by Meta, NVIDIA, Mistral AI, and others, vetted and optimized by Microsoft. A green check mark indicates official support within Azure.
- **Open Models from Hugging Face** This category offers access to hundreds of open-source checkpoints for a wide range of NLP or computer vision tasks. Here you will find models like Mistral, Llama, and DeepSeek.

On the Find the Right Models to Build Your Custom AI Solution page (Figure 5-15), you can search by keyword, and filter by provider, modality (language, vision, audio), task ("summarization"), or language supported.

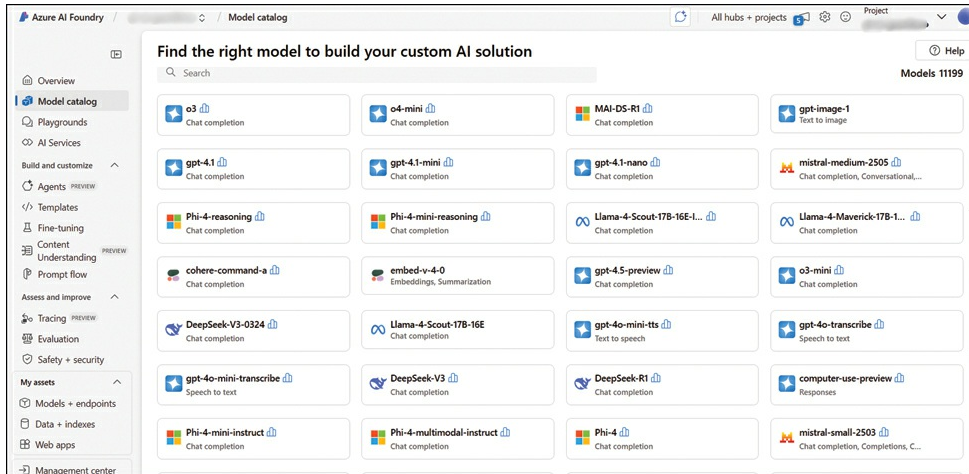


FIGURE 5-15 The model catalog in Azure AI Foundry

Try opening a model to dive into the details: Select the GPT-4.1 model located in the OpenAI suite. As shown in Figure 5-16, opening the tile reveals a model card with quick facts, an extended description, versioning information, accepted input types, size, any known limitations, license agreement, and code samples to simply copy (Python, C#, curl).

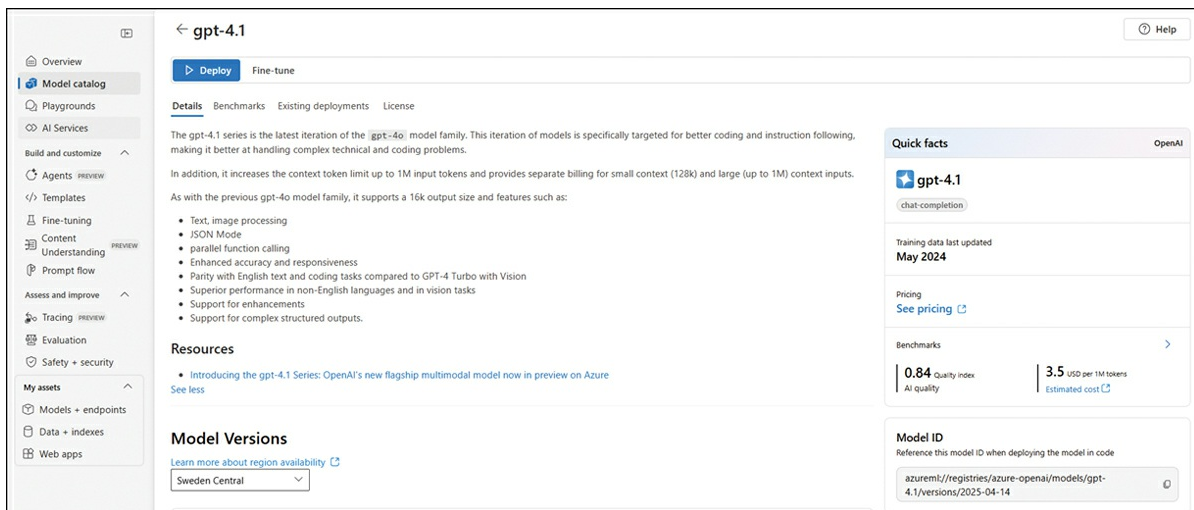


FIGURE 5-16 GPT4.1 general details inside a Foundry project

The integrated leaderboard features in Foundry display BLEU, Rouge, F1, or accuracy scores on public datasets (Figure 5-17). You can select two or more models and select Compare to see the metrics side by side. You can also test and run a custom evaluation on your own data and see how it performs in your own scenario.

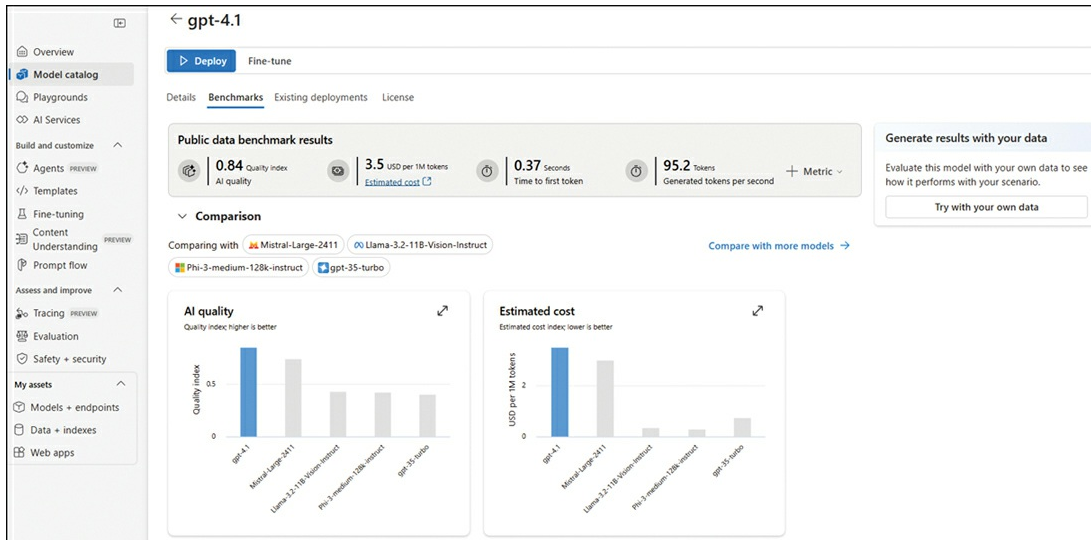


FIGURE 5-17 GPT4.1 benchmarks inside a Foundry project

Finally, Foundry offers rankings of models for quality, cost, and throughput in real-time leaderboards. Open Browse Leaderboards, and you will see the latest top three trade-off charts that depict quality versus throughput or throughput versus cost, and scenario boards such as Question Answering + Reasoning. There you can even rerun the same tests on your own dataset by selecting Try With Your Own Data.

Language models that are deployed as serverless APIs come with default text filters (hate, self-harm, sexual, violent). You can edit these filters per endpoint.

Serverless endpoints behave as the Public Network Access flag of your Foundry hub. When the flag is off, attach a private endpoint for zero-trust traffic. Managed compute takes on the VNet configuration settings that you want to set upon your deployment.

Describe features and capabilities of Azure OpenAI Service

Azure OpenAI is a cloud-based service that allows you to interact with OpenAI's premier AI models through the Azure ecosystem.

Azure OpenAI provides the same models as OpenAI's own API (for example, GPT series for text, o-series for reasoning, Codex for code, DALL·E for images, Whisper for speech), but running on Azure. Thus, you get the benefits of reliability and security (your data remains within your Azure tenant), and added bonuses such as content filtering and responsible AI guardrails that Microsoft has provided.

To create a resource for OpenAI in the Azure portal, select Azure OpenAI and click Create, as shown in Figure 5-18.

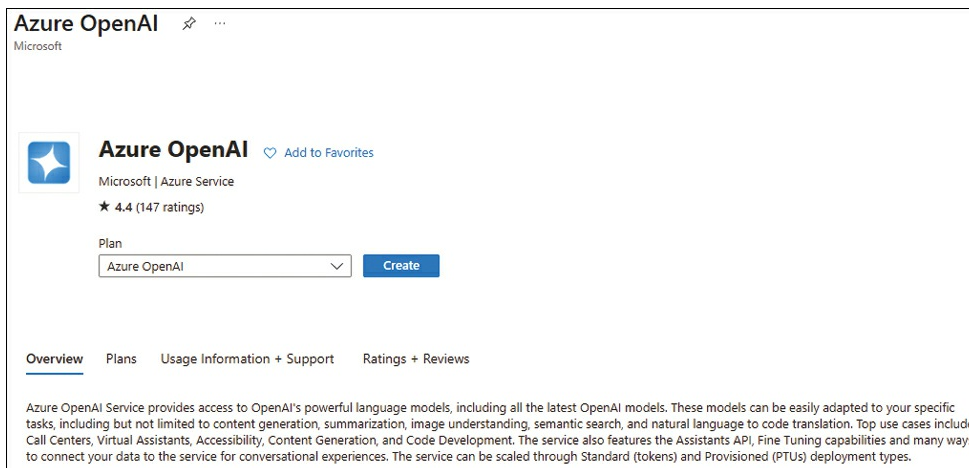


FIGURE 5-18 Overview of Azure OpenAI

To create the resource with the CLI, use the following:

```
az cognitiveservices account create --name <resource-name> --resource-group <group> \
--location <location> --kind OpenAI --sku <pricing-tier> --subscription \
<subscriptionID>
```

Whether you insert the information in the CLI command or select it from the Azure portal (Figure 5-19), you'll need to provide a resource group and name, region, and pricing tier (if you use the pricing tier S0, you will be creating the standard tier) when you create the resource.

Create Azure OpenAI

1 Basics 2 Network 3 Tags 4 Review + submit

Azure OpenAI Service provides access to OpenAI's powerful language models, including all the latest OpenAI models. These models can be easily adapted to your specific tasks, including but not limited to content generation, summarization, image understanding, semantic search, and natural language to code translation. Top use cases include Call Centers, Virtual Assistants, Accessibility, Content Generation, and Code Development. The service also features the Assistants API, Fine Tuning capabilities and many ways to connect your data to the service for conversational experiences. The service can be scaled through Standard (tokens) and Provisioned (PTUs) deployment types.

[Learn more](#)

Project Details

Subscription *
 Resource group *
[Create new](#)

Instance Details

Region
 Name *
 Pricing tier *

[View full pricing details](#)

Content review policy

To detect and mitigate harmful use of the Azure OpenAI Service, Microsoft logs the content you send to the Completions and image generations APIs as well as the content it sends back. If content is flagged by the service's filters, it may be reviewed by a Microsoft full-time employee.

[Learn more about how Microsoft processes, uses, and stores your data](#)

[Apply for modified content filters and abuse monitoring](#)

FIGURE 5-19 Create an Azure OpenAI resource

In the Azure portal, select Keys and Endpoint (on the left of Figure 5-20) to save your access keys necessary to use every Azure OpenAI model.

Azure OpenAI | Keys and Endpoint

Search Regenerate Key1 Regenerate Key2

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Resource visualizer
- Resource Management
- Keys and Endpoint** ☆
- Encryption
- Pricing tier
- Networking
- Identity
- Cost analysis
- Properties

Info These keys are used to access your Azure AI services API. Do not share your keys. Store them securely— for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

KEY 1

KEY 2

Location/Region

Endpoint

FIGURE 5-20 Get keys and endpoint in Azure portal

Finally, you'll need to instantiate your deployment inside the Foundry portal, the SDK, or with CLI. Here is the command to instantiate a deployment with CLI:

```
az cognitiveservices account deployment create --name <name> --resource-group
<my-group> --deployment-name <deployment-name> --model-name "gpt-4.1" --model-version <model-
version> --model-format "OpenAI" --sku-capacity "1" --sku-name "Standard"
```

`--sku-name` chooses the pricing tier of the deployment (Free F0 for limited testing or Standard S0/S1 or higher for paid, higher-throughput use), while `--sku-capacity` tells Azure how many units of that tier you would like to allocate. Usually, you'll set 1 for a single endpoint for your operational purpose, but you can set higher numbers to scale up.

Models Available in Azure OpenAI Service

Azure OpenAI offers a range of pretrained AI models within a few common model classes. The service currently has the following classes:

- **Large Language Models** These models serve as general-purpose language processors, ideal when the job is dealing with text such as writing emails, articles, summarizing reports, or answering questions in chat. The o-series (GPT-o3 and GPT-o4-mini family) is the top tier for reasoning and the best choice for any nuance or long-form tasks. The GPT-4x family provides quicker, cheaper results for everyday tasks.
- **Codex** A class by itself, this model can be considered the programming-specialized counterpart to GPT models. You describe a function in plain English, and the model spits out working Python, JavaScript, or C#. It can also fix or comment on existing code. There is nothing better for building AI pair-programmers or autogenerating boilerplate in your development tools.
- **Embedding Models** Models like text-embedding-small and text-embedding-large take sentences and convert them to high-dimensional vector representations, allowing for semantic comparison based on meaning rather than keywords. You will use them for semantic search, recommendations, or chunking relevant text into a RAG pipeline.
- **Image Models** These models, like DALL·E 2, generate original images from textual prompts. Give the model a prompt like “retro-style poster of a cat astronaut,” and it generates original artwork on the fly. They are perfect for concept art, marketing mockups or rapid illustration. You operate through the Images playground or the REST API, and there's built-in content moderation.
- **Speech Models** The Whisper model converts noisy, accented audio into clean text, or even translates it to any language. It is great for call-center transcripts or voice notes. Newer GPT-4o audio endpoints add real-time text-to-speech and speech-to-speech (through the Realtime beta API), so you can build fully conversational apps.

These main model families in Azure OpenAI Service can be adapted to a whole handful of tasks, including, but not limited to, content generation, text summarization, image creation and understanding, semantic search, and even translating natural language to code. All of the models have pretrained versions that you can use immediately. There's no need to derive these models from scratch, you only have to pick which one is right for your job.

Use cases and tasks for each model

Now that you have a better understanding of the types of tasks you can accomplish with the Azure OpenAI models, here are two examples showing how this group of models can be used in a complete system:

- **Smart help-desk copilot** You deploy a GPT-4.1-based mini chat deployment into your support site. It reads the customer's question, searches a knowledge base with embedding vectors to pull the right docs, and begins to craft a reader-focused answer to the user's question. When a user requests a relevant code sample, the same process is repeated then invokes Codex to output a working Python code snippet. If a marketing team member requires an illustration for the answer, you submit a one-shot request to DALL·E 2 to generate a branded image of the explanation.
- **Multimedia meeting minute maker** You record a project briefing, submit the audio to Whisper, and receive a true transcript. You ask GPT-4o to summarize the important decisions and action items in bullet point format, while a search through embeddings links each point back to the transcripts line for traceability. Do you need a cover slide? Submit the prompt “minimalist timeline with rocket icons” to DALL·E 2, and get a generated image to embed in the deck.

All of these capabilities are available without having to write any custom AI algorithms. You can leverage the existing intelligence of the relevant model via simple API calls or using the Azure AI Studio. You can use the Azure OpenAI to easily integrate and operationalize all of these capabilities in your applications.

Azure AI Foundry vs. Azure OpenAI

Use Azure OpenAI (inside Azure AI Foundry) if you need only a quick GPT endpoint, maybe behind a private VNet. Consider the full Azure AI Foundry, when you want a governed workspace with:

- A model catalog beyond GPT
- Integrated RAG tooling
- Agent orchestration
- Fine-tuning pipelines
- Cost/safety dashboards

In a lot of enterprise scenarios Azure AI Foundry will be the recommended front door and the raw Azure OpenAI resource will be used for the compute backend.

Chapter summary

In this chapter, you learned some of the general concepts related to generative AI. You learned about the features of conversational AI, and you learned about the new Azure AI Foundry platform and the services that surround it. Here are the key concepts from this chapter:

- Generative AI refers to AI systems capable of creating new content, such as text, images, audio, and code, based on patterns learned from large datasets.
- Generative AI models can perform tasks like summarization, content creation, translation, and image generation, often using natural language prompts.
- Conversational AI is a form of generative AI that enables natural dialog between humans and AI agents across various communication channels.
- Conversational AI systems understand what is being said and respond in human-like ways using technologies like natural language processing (NLP),

speech-to-text, and text-to-speech.

- Chatbots are applications that guide users to solutions using a mix of free-form language understanding and predefined responses. They can handle both text and voice inputs.
- Chatbots are a major use case for conversational AI but are just one part of a larger ecosystem that includes virtual assistants, copilots, and autonomous agents.
- Virtual assistants go beyond answering questions; they can perform tasks or take actions in response to commands.
- Transparency is one of the principles of Responsible AI that applies to bots.
- Azure AI Foundry organizes every model, dataset, and cost transparency inside projects.
- The Foundry model catalog enables you to search, compare, and one-click publish Microsoft, partner, and open-source checkpoints.
- Retrieval-Augmented Generation (RAG) bases answers on your data by either indexing documents or linking to your existing Azure AI Search service.
- The Azure AI Foundry Agent Service takes a model and upgrades it to an autonomous agent that lets the model plan its tasks, calls external tools, enforces policy, and exposes a traceable HTTPS endpoint.
- Fine-tuning allows you to customize the foundation models for your domain data, and enables one-click side-by-side evaluations against the base model.
- Select serverless inference (zero idle cost) for bursty traffic, or go with managed compute (fixed SLA, private networking, reserved GPU capacity).
- Azure OpenAI Service provides several model families, such as LLM, Embedding, Image, Speech, and Codex.
- Use Foundry when you need a managed, multimodel environment; use a standalone Azure OpenAI resource when you only need a rapid, private GPT endpoint.

Need More Review? Hands-On Lab

For more hands-on experience with natural language processing, complete the lab found at microsoftlearning.github.io/mslearn-ai-fundamentals/Instructions/Exercises/03-text-analysis.html.

Thought experiment

To apply what you have learned in this chapter and demonstrate your knowledge of its topics, try this thought experiment. You can find the answers in the section that follows.

You work for Tailspin Toys, a manufacturer of toys crafted from wood. Tailspin Toys are concerned with customer experience and want to implement omnichannel customer support with an automated first-line solution.

Tailspin Toys plans to launch a new line of products that teaches children the basics of logic using moveable wooden tags to represent logic gates and wooden gears to demonstrate how data is processed. These educational toys will contain electrical contacts and sensors that will capture the positions of the wooden components. As part of this launch, an app for mobile phones is planned that can take pictures of the arrangement of the components. The app will confirm if the components are correctly arranged, or if they're not, the app will offer demonstration videos and read instructions to the customer. The app needs to use speech input. Users can request the bot to upload their pictures to a community board where customers can share their progress and experiences.

The existing support system relies on customer support agents accessing a variety of documents, databases, and applications, including the following:

- PDF files
- Microsoft Excel spreadsheets
- A SQL database
- Various Microsoft SharePoint lists
- Microsoft Word documents held on SharePoint

Answer the following questions:

1. What technologies enable a chatbot to understand and respond to user questions?
2. Name some common use cases for webchat bots.
3. What Responsible AI principle should Tailspin Toys apply to ensure users are aware they are interacting with a bot in their support app?
4. How does Retrieval-Augmented Generation (RAG) help ensure accurate answers from the bot in Tailspin Toys' support system?
5. What is the purpose of the Azure AI Foundry Agent Service?
6. What role does Azure AI Content Safety play in model deployment in Foundry?
7. Publishing support questions and answers to a website or app is an example of which workload?
8. If a customer asks the bot to share a photo of their toy setup to a community board, what workload type does this represent in Azure AI Foundry?
9. What capabilities does the model catalog in Azure AI Foundry offer to help Tailspin Toys choose the right AI models for their app?
10. Which sources could you use to build a modern FAQ knowledge base?
11. Which Azure resources are created when you enable a RAG-powered FAQ in Azure AI Foundry?
12. What file format do you use to upload custom persona or "chit-chat" data for fine-tuning?

Thought experiment answers

This section contains the solutions to the thought experiment. Each answer explains why the answer choice is correct.

1. Natural language processing (NLP) enables webchat bots to understand and respond by converting text into intents using language models, which the bot can then act on to formulate responses.
2. Common use cases include customer support, answering frequently asked questions, assisting with online ordering, and travel reservation and booking.
3. The Transparency principle of Responsible AI states that users must be informed when they are interacting with a bot. The bot should clearly identify itself, explain its capabilities and limitations, and provide a path to escalate the issue to a human agent when needed.
4. RAG allows the language model to pull relevant information from Tailspin's support documents before generating a response. This ensures the answers are grounded in up-to-date, company-specific knowledge rather than relying solely on pretrained data.
5. The AI Agent Service enables models to perform complex, multistep tasks by planning actions and using tools such as search or file retrieval. It manages tasks like booking travel or troubleshooting over the course of a full conversation or workflow.
6. All language models deployed using Standard or Managed Compute are fronted by Azure AI Content Safety, which monitors outputs for hate speech, sexual content, self-harm, and violence. Filtering can be disabled, but doing so transfers responsibility for harmful outputs to the user.
7. Publishing support questions and answers to a website or an app uses a Q&A workload where existing documents are ingested into a RAG pipeline so that an LLM, for example an Azure OpenAI model, can retrieve and provide the most relevant FAQ answers.
8. Publishing customer images to a community board is considered an action workload, where the agent invokes an Azure Function or Logic App to save the image in a specified location (for reference, a SharePoint library or blob storage), then the agent returns to the user confirming the upload was successful.
9. The model catalog provides access to hundreds of open-source and proprietary AI models, each with detailed model cards including performance benchmarks, licensing information, and sample code. Tailspin Toys can also compare models side by side using a built-in leaderboard feature to select the most suitable one.
10. A modern FAQ knowledgebase can integrate all your support sources: PDF and Word documents in SharePoint, Excel or CSV files, SQL database records through built-in-connection, and SharePoint lists, by capturing them through the Foundry RAG pipeline to build a comprehensive answer index.
11. When you activate RAG-powered FAQ in Azure AI Foundry, the system will automatically provision an Azure AI Search index for hybrid retrieval, and Azure OpenAI Service endpoint for natural language responses, along with any additional compute and storage, which is handled by Foundry.
12. To provide custom persona or "chit-chat" data for fine-tuning, you upload a JSONL file of prompt-completion pairs, with each row representing the training example that adjusts the tone and behavior of the base model.

AI-900 Microsoft Azure AI Fundamentals exam updates

The purpose of this chapter

This chapter will be updated over time, and a PDF will be posted online so you can access the latest information about exam changes, even after purchasing this book.

Why do you need a chapter that updates over time? There are three reasons:

- To add more technical content to the book before it is time to replace the current book edition with the next edition. This chapter will include additional technology content and possibly additional PDFs containing more content.
- To communicate details about the next exam version, tell you about publishing plans for that edition, and help you understand what that means to you.
- To accurately map the current exam objectives to existing chapter content. While exam objectives evolve and products are renamed, much of the content in this book will remain accurate and relevant. In addition to covering any content gaps that appear through additions to the objectives, this chapter notes how the new objectives map to the current text.

After the initial publication of this book, supplemental updates will be provided as digital downloads for minor exam updates. We (Microsoft Press) may announce a new edition if an exam has major changes or accumulates enough minor changes. We will do our best to provide any updates to you free of charge before we release a new edition. However, if the updates are significant enough between editions, we may release the updates as a low-priced standalone e-book.

If we do produce a free updated version of this chapter, you can access it on the book's product page, simply visit MicrosoftPressStore.com/ERAI9002e/downloads to view and download the updated material.

About possible exam updates

Microsoft reviews exam content periodically to ensure it aligns with the technology and job role associated with the exam. This includes, but is not limited to, incorporating functionality and features related to technology changes, changing skills needed for success within a job role, and revisions to product names. Microsoft updates the exam details page to notify candidates when changes occur. Once you register this book, you will be notified when updates are made, and the updated chapter is available.

Impact on you and your study plan

Microsoft's information helps you plan, but it also means that the exam might change before you pass the current exam. That impacts you, affecting how we deliver this book to you. This chapter gives us a way to communicate in detail about those changes as they occur, but you should watch the following sites for news:

- **Microsoft Learn** Check the main source for up-to-date information: microsoft.com/learn. Make sure to sign up for automatic notifications.
- **Microsoft Press** Find information about products, offers, discounts, and free downloads: microsoftpressstore.com. Make sure to register your purchased products.

As changes to the AI-900 exam occur, we will publish an updated version of this chapter, which will likely include the following:

- **The content that has been removed** If you plan to take the new exam version, you can ignore the removed content when studying for the exam.
- **New content planned per new exam topics** This will tell you what's coming.

Exam objective updates

You can find the current study guide for the AI-900 exam at learn.microsoft.com/en-us/credentials/certifications/resources/study-guides/ai-900.

It contains the most recent version of the exam objective domain.

Updated technical content

The current version of this chapter has no additional technical content.

Objective mapping

This *Exam Ref* is structured by the authors based on the topics and technologies covered on the exam and not based on the specific order of topics in the exam objectives. [Table 6-1](#) maps the current version of the exam objectives to chapter content, allowing you to locate where a specific exam objective item has coverage without consulting the index.

TABLE 6.1 Exam Objectives mapped to chapters

Exam Objective	Chapter
----------------	---------

Exam objective	Chapter
Describe Artificial Intelligence workloads and considerations	
Identify features of common AI workloads	1
<ul style="list-style-type: none"> ▪ Identify computer vision workloads ▪ Identify natural language processing workloads ▪ Identify document processing workloads ▪ Identify generative AI workloads 	
Identify guiding principles for Responsible AI	1
<ul style="list-style-type: none"> ▪ Describe the Fairness principle ▪ Describe the Reliability and Safety principle ▪ Describe the Privacy and Security principle ▪ Describe the Inclusiveness principle ▪ Describe the Transparency principle ▪ Describe the Accountability principle ▪ Understand the Microsoft AI for Good program 	
Describe fundamental principles of machine learning on Azure	
Identify common machine learning types	2
<ul style="list-style-type: none"> ▪ Understand machine learning model types ▪ Describe regression models ▪ Describe classification models ▪ Describe clustering models ▪ Describe deep learning techniques ▪ Describe features of the Transformer architecture 	
Describe core machine learning concepts	2
<ul style="list-style-type: none"> ▪ Understand the machine learning workflow ▪ Identify the features and labels in a dataset for machine learning ▪ Describe how training and validation datasets are used in machine learning ▪ Describe how machine learning algorithms are used for model training ▪ Select and interpret model evaluation metrics 	
Identify core tasks in creating a machine learning solution	2
<ul style="list-style-type: none"> ▪ Understand machine learning on Azure ▪ Understand Azure Machine Learning Studio ▪ Describe data ingestion and preparation ▪ Describe feature selection and engineering ▪ Describe model training and evaluation ▪ Describe model deployment and management 	
Describe capabilities of no-code machine learning with Azure Machine Learning	2
<ul style="list-style-type: none"> ▪ Describe Azure Automated Machine Learning ▪ Describe Azure Machine Learning Designer 	
Describe features of computer vision workloads on Azure	
Identify common types of computer vision solution	3
<ul style="list-style-type: none"> ▪ Describe AI Services ▪ Understand computer vision ▪ Describe image classification ▪ Describe object detection ▪ Describe optical character recognition 	

- Describe facial detection, recognition, and analysis

Identify Azure tools and services for computer vision tasks 3

- Understand the capabilities of the Vision service
- Understand the Custom Vision service
- Understand the Face service

Describe features of Natural Language Processing (NLP) workloads on Azure

Identify features of common NLP workload scenarios 4

- Describe natural language processing
- Describe language modeling
- Describe key phrase extraction
- Describe named entity recognition
- Describe sentiment analysis
- Describe speech recognition and synthesis
- Describe translation

Identify Azure tools and services for NLP workloads 4

- Identify the capabilities of the Azure AI Language service
- Identify the capabilities of the Azure AI Speech service
- Identify the capabilities of the Azure AI Translator service

Describe features of generative AI workloads on Azure

Identify features of generative AI solutions 5

- Identify features of generative AI models
- Identify common scenarios for generative AI
- Identify Responsible AI considerations for generative AI

Identify Azure services and capabilities for generative AI 5

- Describe features and capabilities of Azure AI Foundry
 - Describe features and capabilities of the Azure AI Foundry model catalog
 - Describe features and capabilities of Azure OpenAI Service
-

Index

A

- accountability (Responsible AI), 171
- Accountability principle, 11
 - Responsible AI, 10
- accuracy, improving, 12–13
- ACI (Azure Container Instance), 58
- Address entity (Language service), 137
- Adult feature (computer vision), 91
- Age attribute (facial detection), 111
- agents, building, AI Foundry, 175–178
- AI (artificial intelligence), 1, 10–11
 - AI for Good, 10
 - conversational, 4
 - Responsible, 7–8
 - Accountability principle, 10
 - Fairness principle, 8
 - Inclusiveness principle, 9
 - Privacy and Security principle, 9
 - Reliability and Safety principle, 8
 - Transparency principle, 9–10
 - workloads, 1–3. *see also* workloads
- AI -900, 191–192
- AI Agent Service, 80
- AI for Good, 10
- AI Foundry, 172
 - building agents, 175–178
 - model catalog, 181–183
 - versus OpenAI service, 187
 - projects, 172–175
- AI Model Inference, 80
- AI Search, 80, 179
- AI Services
 - containers, 85
 - deploying, 82–84
 - describing, 80–82
 - security, 84–85
- AKS (Azure Kubernetes Service) online endpoints, 58
- algorithms
 - hyperparameters, 54
 - ML (machine learning), 1
 - model training, 29
- Anomaly Detector, 81
- ApacheSpark, 33
- artificial intelligence (AI). *see* AI (artificial intelligence)
- aspect-based sentiment analysis, 131
- attention mechanisms, 23–24
- authoring, Azure Learning Studio, 39
- Automated ML, 39, 59–65
- Azure
 - computer vision system workloads, 2–3
 - Designer, 65–75
 - model creation, 65–72
 - model deployment, 72–75
 - Learning Studio, 38–41
 - marketplace, 34
 - ML (machine learning), 33

- workspace, 33–38
- Azure Container Instance (ACI), 58
- Azure Kubernetes Service (AKS) online endpoints, 58

B

- backpropagation, 21
- batch inferencing, 57
- BERT (Bidirectional Encoder Representation from Transformers), 24
- bias, ML (machine learning), 27–28
- Bidirectional Encoder Representation from Transformers (BERT), 24
- binning, 51
- Blur attribute (facial detection), 112
- Bot Service, 81
- Brands feature (computer vision), 92

C

- Captions feature (computer vision), 91–94
- Categories feature (computer vision), 92
- celebrity image recognition, 96–97
- classes, 20
- classification models, 19
 - evaluating, 31–32
- CLU (Conversational Language Understanding), 153–157
- clustering models
 - evaluating, 32
 - metrics, 32
 - unlabeled data, 20
- codex models, 185
- Color feature (computer vision), 92
- compact domains, 106
- compute instances, 39–41
- compute targets, 37–41
- computer clusters, creating, 52–54
- computer vision systems, 2–3, 11, 85, 89–90
 - capabilities, 90–101
 - captions, 93–94
 - content tags, 95–96
 - Custom Vision, 102–103
 - creating models, 103–109
 - modes, 103
 - versus Vision, 110
 - versus document intelligence, 102
 - domain-specific content, 96–97
- Face, 110–111
 - detection, 111–113
 - liveness, 113
 - recognition, 114
- facial detection, recognition, and analysis, 89–90
- identifying common types, 79–90
- image analysis, 91–92
- image classification, 86–87
- object detection, 87–88, 94–95
- OCR (optical character recognition), 88–89, 97–102
- thumbnail generation, 96
- use cases, 85–86
- Video Indexer, 114–116

- computing, Azure Learning Studio, 39
- confusion matrix, 31–32
- containers
 - AI Services, 85
 - deploying, 58

- Content Moderator, [81](#)
- Content Safety, [81](#)
- content tags, [95–96](#)
- conversational AI (artificial intelligence), [4](#), [11](#)
- Conversational Language Understanding (CLU), [153–157](#)
- core tasks, identifying, [32–37](#)
- conversational AI, [165](#)
- Custom Named Entity Recognition, [137](#)
- Custom Question Answering (Language service), [146–153](#)
- Custom Vision, [81](#), [102–103](#)
 - creating models, [103–109](#)
 - modes, [103](#)
 - versus Vision, [110](#)

D

- data assets, creating, [44](#)
- data ingestion, [42–49](#)
- data preparation, [49–50](#)
- data protection (Responsible AI), [171](#)
- Data Science virtual machine (DSVM), [33](#)
- datasets
 - identifying features and labels, [26–28](#)
 - ML (machine learning), feature and label identification, [26–28](#)
 - registering, Automated ML, [59–65](#)
 - splitting, [28](#)
 - training, [28–29](#)
 - validation, [28](#)
- DateTime entity (Language service), [137](#)
- decoders, [23–24](#)
- deep learning, [21–22](#)
- deployment
 - AI Services, [82–84](#)
 - models, [57–58](#)
- deprecation, models, [173](#)
- Design Ideas feature (PowerPoint), [1](#)
- Designer (Azure), [65–75](#)
 - model creation, [65–72](#)
 - model deployment, [72–75](#)
- Detect Language operation (Language service), [130–131](#)
- detection
 - Face service, [111–113](#)
 - objects, computer vision systems, [94–95](#)
- Document Intelligence, [5](#), [11](#), [81](#)
- document intelligence, [128](#)
 - versus computer vision systems, [102](#)
 - OCR (optical character recognition), [99–102](#)
- document processing workloads, [4–5](#)
- domains, image classification, [106](#)
- domain-specific content, computer vision systems, [96–97](#)
- DSVM (Data Science virtual machine), [33](#)

E

- Email entity (Language service), [137](#)
- embedded models, [186](#)
- embedders, [22–23](#)
- encoders, [22–24](#)
- endpoints
 - managed online, [58](#)
 - multi-service AI, [109](#)
- engineering, features, [27](#)
- evaluating, models, [51–57](#)

evaluation metrics, models, 29
Event entity (Language service), 137
Exposure attribute (facial detection), 112

F

Face service, 81, 92, 110–111
 detection, 111–113
 liveness, 113
 recognition, 114
 versus Video Indexer, 116
facial detection, recognition, and analysis, computer vision systems, 89–90
Facial hair attribute (facial detection), 112
fairness (Responsible AI), 171
Fairness principle, 11
 Responsible AI, 8
feature engineering, 51
Feature Hashing module, 51
features
 bias, 27–28
 datasets, identifying, 26–28
 engineering, 27
 selection, 28, 50
form recognition, 5
frequency text analysis, 123

G

GenAIOps, 6
generative AI, 11
 AI Foundry
 building agents, 175–178
 projects, 172–175
 features, 166–168
 OpenAI Service, 183–185
 models, 180–181, 185–186
 prompt engineering, 168
 RAG (Retrieval-Augmented Generation), 179–181
 Responsible AI, 170–172
 social media posts, 6
Generative Pretrained Transformer (GPT), 24
generative workloads, 5–7
Get Thumbnail operation, 96
Glasses attribute (facial detection), 112
GPT (Generative Pretrained Transformer), 24
GPUs (graphic processing units), 22
graphic processing units (GPUs), 22

H

Hair attribute (facial detection), 112
Head pose attribute (facial detection), 112
human oversight (Responsible AI), 171
hyperparameters, 54

I

image analysis, computer vision systems, 95–96, 91–92
image classification, computer vision systems, 86–87
image classification mode, Custom Vision, 103
image models, 186
ImageType feature (computer vision), 92
Immersive Reader, 81, 128

- Inclusiveness principle, [11](#)
 - Responsible AI, [9](#)
- indexing (AI Search), [179](#)
- inferencing, [57–58](#)
- ingestion, [42–49](#)
- interactive voice response (IVR), [169](#)
- IoT Edge, [58](#)
- IP entity (Language service), [137](#)
- IVR (interactive voice response), [169](#)

J

- Jupyter notebook, [33](#)

K

- key phrase extraction, [3](#), [135–136](#)
 - NLP (natural language processing), [125](#)
- K-means clustering, [20](#)

L

- labels, [17](#), [19](#)
 - datasets, identifying, [26–28](#)
- landmark image recognition, [96–97](#)
- language detection, [3](#)
 - text analysis, [123](#)
- language modeling, [123](#), [124–125](#)
- Language service, [81](#), [128](#), [129–145](#)
 - CLU (Conversational Language Understanding), [153–157](#)
 - Custom Question Answering, [146–153](#)
 - Detect Language operation, [130–131](#)
 - Key Phrase Extraction operation, [135–136](#)
 - NER (named entity recognition), [136–140](#)
 - Sentiment Analysis operation, [131–135](#)
 - Summarization feature, [140–145](#)
 - use cases, [157](#)
- language translation, [127](#)
- Language Understanding (LUIS), [81](#)
- large language models (LLMs), [22](#), [185](#)
- layout analysis, [4](#)
- Learning Studio, Azure ML, [38–41](#)
- learning types, [16–17](#)
 - deep, [21–22](#)
 - supervised, regression models, [17–19](#)
- liveness, Face service, [113](#)
- LLMs (large language models), [22](#), [185](#)
- Location entity (Language service), [137](#)
- long short term memory (LSTM), [23](#)
- loss function, [21](#)
- LSTM (long short term memory), [23](#)
- LUIS (Language Understanding), [81](#)

M

- machine learning (ML). *see* [ML \(machine learning\)](#)
- MAE (mean absolute error), [31](#)
- Makeup attribute (facial detection), [112](#)
- managed online endpoints, [58](#)
- management, models, [57–58](#)
- marketplace, Azure, [34](#)
- mean absolute error (MAE), [31](#)
- mean zero one error (MZOE), [31](#)

- metrics
 - clustering models, 32
 - regression models, 31
- Metrics Advisor, 81
- Microsoft Azure. *see* [Azure](#)
- Microsoft PowerPoint. *see* [PowerPoint](#)
- ML (machine learning), 1, 10, 15
 - algorithms, 1
 - Azure, 33
 - bias, 27–28
 - datasets
 - feature and label identification, 26–28
 - training, 28–29
 - validation, 28
 - deep learning, 21–22
 - identifying core tasks, 32–37
 - Learning Studio, 38–41
 - model evaluation metrics, 29
 - model training, algorithms, 29
 - model types, 15–16
 - classification, 19
 - clustering, 20–21
 - learning, 16–17
 - regression, 17–19
 - Transformer architecture, 22–24
 - no-code machine learning, 58–75
 - Automated ML, 59–65
 - workflow, 25
 - workspace, 34–38
- MLOps, 6
- model catalog, AI Foundry, 181–183
- model types, ML (machine learning), 15–16
 - classification, 19
 - clustering, 20–21
 - learning, 16–17
 - regression, 17–19
 - Transformer architecture, 22–24
- models
 - classification, evaluating, 31–32
 - clustering, evaluating, 32
 - codex, 185
 - creating, 65–72
 - Custom Vision, creating, 103–109
 - data ingestion, 42–49
 - data preparation, 49–50
 - deployment, 57–58, 72–75
 - deprecation, 173
 - embedded, 186
 - evaluation metrics, 29
 - fine-tuning, 180–181
 - image, 186
 - management, 57–58
 - model-building tasks, 56
 - OpenAI Service, 185–186
 - regression, 31
 - speech, 186
 - training, algorithms, 29
 - training and evaluation, 51–57
 - web services, 34–38
- modes, Custom Vision, 103
- multi-head attention, 24
- MZOE (mean zero one error), 31

N

- named entity recognition (NER), 3, 123
- named field extraction, 4
- natural language processing (NLP). *see* [NLP \(natural language processing\)](#)
- NER (named entity recognition), 3, 123–125, 136–140
- NLP (natural language processing), 3–4, 11, 121
 - describing, 122
 - key phrase extraction, 125
 - language modeling, 124–125
 - Language service, 129–145
 - CLU (Conversational Language Understanding), 153–157
 - Custom Question Answering, 146–153
 - Detect Language operation, 130–131
 - Key Phrase Extraction operation, 135–136
 - NER (named entity recognition), 136–140
 - Sentiment Analysis operation, 131–135
 - Summarization feature, 140–145
 - use cases, 157
 - language translation, 127
 - NER (named entity recognition), 125
 - sentiment text analysis, 126
 - speech recognition, 126
 - Speech service, 157–158, 158, 158–159, 159
 - speech synthesis, 126–127
 - text analytics, 123
 - text services, 128–129
 - Translator service, 159–161
 - use cases, 124
- no-code machine learning, 58–75
 - Automated ML, 59–65
- nodes, computer clusters, 52–54
- Noise attribute (facial detection), 112
- normalization, 28

O

- object detection, computer vision systems, 87–88, 94–95
- object detection mode, Custom Vision, 103
- Occlusion attribute (facial detection), 112
- OCR (optical character recognition), computer vision systems, 88–89, 97–102
- ONNX (Open Neural Network Exchange), 33
- Open Neural Network Exchange (ONNX), 33
- OpenAI Service, 80, 128, 183–185
 - versus AI Foundry, 187
 - models, 185–186
 - fine-tuning, 180–181
- Opinion Mining, 131
- optical character recognition (OCR). *see* [OCR \(optical character recognition\)](#)
- Organization entity (Language service), 137

P

- Part of Speech (PosTag) text analysis, 123
- patterns, 15–16
- People Detection feature (computer vision), 91
- Person entity (Language service), 137
- Personalizer, 81
- PersonType entity (Language service), 137
- Phone Number entity (Language service), 137
- phoneme, 126
- pipelines, 52
- PowerPoint, Design Ideas feature, 1

- predictive power, increasing, [51](#)
- privacy (Responsible AI), [171](#)
- Privacy and Security principle, [11](#)
 - Responsible AI, [9](#)
- Product entity (Language service), [137](#)
- projects, AI Foundry, [172–175](#)
- prompt engineering, generative AI, [168](#)
- Prompt Flow, [39](#)
- PyCharm notebook, [33](#)
- PyTorch, [33](#)

Q

- QnA Maker, [81](#)
- Quantity entity (Language service), [137](#)
- querying (AI Search), [179](#)
- question answering and summarization, NLPs (natural language processing), [3](#)
- Question Answering feature (Language service), [146–153](#)

R

- R² (coefficient of determination), [31](#)
- RAE (relative absolute error), [31](#)
- RAG (Retrieval-Augmented Generation), [6](#), [179–181](#)
- Read visual feature (computer vision), [91](#)
- real-time inferencing, [57](#)
- recognition, Face service, [114](#)
- recommender model, [20](#)
- recurrent neural networks (RNNs), [23](#)
- registering, datasets, Automated ML, [59–65](#)
- regression models, [17–19](#)
 - metrics, [31–32](#)
- reinforcement learning, [17](#)
- relative absolute error (RAE), [31](#)
- relative squared error (RSE), [31](#)
- reliability (Responsible AI), [171](#)
- Reliability and Safety principle, [11](#)
 - Responsible AI, [8](#)
- Responsible AI, [7–8](#), [11](#)
 - Accountability principle, [10](#)
 - Fairness principle, [8](#)
 - generative AI, [170–172](#)
 - Inclusiveness principle, [9](#)
 - Privacy and Security principle, [9](#)
 - Reliability and Safety principle, [8](#)
 - Transparency principle, [9–10](#)
- Rest APIs (representational state transfer application programming interfaces), [2](#)
- retrieval, RAG (Retrieval-Augmented Generation), [179–181](#)
- Retrieval-Augmented Generation (RAG), [6](#)
- RMSE (root mean squared error), [31](#)
- RNNs (recurrent neural networks), [23](#)
- root mean squared error (RMSE), [31](#)
- RSE (relative squared error), [31](#)
- R-squared (coefficient of determination), [31](#)

S

- safety (Responsible AI), [171](#)
- scikit-learn, [33](#)
- scoring, [54–56](#)
- SDKs (software development kits), [2](#)
- security, AI Services, [84–85](#)
- selection, features, [27](#)

- self-attention mechanism, 23
- semantic modeling, 123
- sentiment analysis, 3
- sentiment text analysis, 123, 126, 131–135
- Skill entity (Language service), 137
- Smart Crop feature (computer vision), 91
- Smile attribute (facial detection), 112
- social media posts, generative AI, 6
- Speaker Recognition, 159
- speech models, 186
- Speech Recognition service, 123, 126–128
- Speech Service, 81
- Speech service, 157–158
 - Speaker Recognition, 159
 - Speech to Text service, 158
 - Speech Translation service, 158–159
 - Text to Speech service, 158
 - Translator service, 159–161
 - use cases, 159
- speech synthesis, 126–127
- Speech to Text service, 128, 158
- Speech Translation service, 128, 158–159
- splitting, datasets, 28
- statistical text analysis, 123
- Summarization feature (NLP), 140–145
- supervised learning, 17, 61
 - regression models, 17–19

T

- table extraction, 5
- Tags feature, 95–96
- Tags feature (computer vision), 91
- TensorFlow, 33
- text analytics, NLPs (natural language processing), 123
- Text to Speech service, 128, 158
- thumbnail generation, 96
- tokenize text analysis, 123
- topic detection, 123
- training, models, 51–57
- Transformer architecture, 22–24
- Translator Service, 81
- Translator service, 128, 159–161
- transparency (Responsible AI), 171
- Transparency principle, 11
 - Responsible AI, 9–10
- types, computer vision systems, identifying, 79–90

U

- unlabeled data, clustering models, 20
- unsupervised learning, 17
- URL entity (Language service), 137
- use cases
 - CLU (Conversational Language Understanding), 154
 - computer vision systems, 85–86
 - Language service, 157
 - NLPs (natural language processing), 124
 - OpenAI Service models, 186
 - Speech service, 159

V

validation datasets, [28–29](#)
Video Indexer, [81, 114–116](#)
Vision, versus Custom Vision, [110](#)
vision systems. *see* [computer vision systems](#)

W

web services, models, [34–38](#)
workflow, ML (machine learning), [25](#)
workloads, [1–3](#)

- case study, [11–13](#)
- computer vision systems, [2–3](#)
- document processing, [4–5](#)
- generative, [5–7](#)
- NLP (natural language processing), [3–4](#)
 - key phrase extraction, [125](#)
 - language modeling, [124–125](#)
 - Language service, [129–145](#)
 - language translation, [127](#)
 - NER (named entity recognition), [125](#)
 - sentiment text analysis, [126](#)
 - speech recognition, [126](#)
 - speech synthesis, [126–127](#)
 - text services, [128–129](#)
- NLPs (natural language processing), [121](#)
 - describing, [122](#)
 - text analytics, [123](#)
 - use cases, [124](#)

workspace, Azure ML, [33–38](#)

Plug into learning at

MicrosoftPressStore.com

The Microsoft Press Store by Pearson offers:

- Free U.S. shipping
- Buy an eBook, get multiple formats – PDF and EPUB – to use on your computer, tablet, and mobile devices
- Print & eBook Best Value Packs
- eBook Deal of the Week – Save up to 60% on featured title
- Newsletter – Be the first to hear about new releases, announcements, special offers, and more
- Register your book – Find companion files, errata, and product updates, plus receive a special coupon* to save on your next purchase

Discounts are applied to the list price of a product. Some products are not eligible to receive additional discounts, so your discount code may not be applied to all items in your cart. Discount codes cannot be applied to products that are already discounted, such as eBook Deal of the Week, eBooks that are part of a book + eBook pack, and products with special discounts applied as part of a promotional offering. Only one coupon can be used per order.



Hear about it first.

Since 1984, Microsoft Press has helped IT professionals, developers, and home office users advance their technical skills and knowledge with books and learning resources.

Sign up today to deliver exclusive offers directly to your inbox.

- New products and announcements
- Free sample chapters
- Special promotions and discounts
- ... and more!

MicrosoftPressStore.com/newsletters



Exam Ref AI-900 Microsoft Azure AI Fundamentals Second Edition

List of URLs

Chapter 1: Describe artificial intelligence workloads and considerations

www.microsoft.com/ai/responsible-ai

www.microsoft.com/ai/ai-for-good

Chapter 3: Describe features of computer vision workloads on Azure

[https://{endpoint}/computervision/imageanalysis:analyze\[?features\]\[&model-version\]\[&language\]\[&api-version\]](https://{endpoint}/computervision/imageanalysis:analyze[?features][&model-version][&language][&api-version])

[https://{endpoint}/vision/v3.2/analyze\[?visualFeatures\]\[&details\]\[&language\]](https://{endpoint}/vision/v3.2/analyze[?visualFeatures][&details][&language])

<https://{endpoint}/computervision/imageanalysis:analyze?api-version=2024-02-01&features=caption>

<https://{endpoint}/computervision/imageanalysis:analyze?api-version=2024-02-01&features=densecaptions>

[https://{endpoint}/vision/v3.2/analyze?visualFeatures=Description\[?maxCandidates\]\[&language\]](https://{endpoint}/vision/v3.2/analyze?visualFeatures=Description[?maxCandidates][&language])

[https://{endpoint}/vision/v3.2/describe\[?maxCandidates\]\[&language\]](https://{endpoint}/vision/v3.2/describe[?maxCandidates][&language])

<https://{endpoint}/computervision/imageanalysis:analyze?api-version=2024-02-01&features=objects>

<https://{endpoint}/vision/v3.2/analyze?visualFeatures=Objects>

<https://{endpoint}/vision/v3.2/detect>

<https://{endpoint}/computervision/imageanalysis:analyze?api-version=2024-02-01&features=tags>

[https://{endpoint}/vision/v3.2/analyze?visualFeatures=Tags\[?language\]](https://{endpoint}/vision/v3.2/analyze?visualFeatures=Tags[?language])

[https://{endpoint}/vision/v3.2/tag\[?language\]](https://{endpoint}/vision/v3.2/tag[?language])

[https://{endpoint}/computervision/imageanalysis:analyze?api-version=2024-02-01&features=smartCrops\[&smartcrops-aspect-ratios\]](https://{endpoint}/computervision/imageanalysis:analyze?api-version=2024-02-01&features=smartCrops[&smartcrops-aspect-ratios])

[https://{endpoint}/vision/v3.2/generateThumbnail\[?width\]\[&height\]\[&smartCropping\]](https://{endpoint}/vision/v3.2/generateThumbnail[?width][&height][&smartCropping])

[https://{endpoint}/vision/v3.2/models/{model}/analyze\[?language\]](https://{endpoint}/vision/v3.2/models/{model}/analyze[?language])

<https://{endpoint}/computervision/imageanalysis:analyze?api-version=2024-02-01&features=read>

<https://{endpoint}/vision/v3.2/ocr>

[https://{endpoint}/documentintelligence/documentModels/prebuilt-read:analyze\[?output\]\[&api-version\]](https://{endpoint}/documentintelligence/documentModels/prebuilt-read:analyze[?output][&api-version])

[https://{endpoint}/documentintelligence/documentModels/prebuilt-read/analyzeResults/{resultId}\[?api-version\]](https://{endpoint}/documentintelligence/documentModels/prebuilt-read/analyzeResults/{resultId}[?api-version])

www.customvision.ai

[https://{endpoint}/face/v1.0/detect\[?returnFaceId\]\[&returnFaceLandmarks\]\[&returnFaceAttributes\]\[&returnRecognitionModel\]\[&recognitionModel\]\[&detectionModel\]\[&faceIdTimeToLive\]](https://{endpoint}/face/v1.0/detect[?returnFaceId][&returnFaceLandmarks][&returnFaceAttributes][&returnRecognitionModel][&recognitionModel][&detectionModel][&faceIdTimeToLive])

Chapter 4: Describe features of natural language processing (NLP) workloads on Azure

<https://{endpoint}/language:analyze-text?api-version=2023-11-15-preview>

www.contososteakhouse.com

<https://en.wikipedia.org/wiki/Seattle>

<https://{endpoint}/language/analyze-text/jobs?api-version=2023-04-01>

<https://<your-language-resource-endpoint>/language/analyze-text/jobs/<your-job-id>?api-version=2023-04-01>

https://{endpoint}/language:query-knowledgebases?projectName={YOUR_PROJECT_NAME}&api-version=2021-10-01&deploymentName={DEPLOYMENT_NAME}

<https://{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/import?api-version=2023-04-01>

<https://{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/import/jobs/{JOB-ID}?api-version=2023-04-01>

<https://{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/train?api-version=2023-04-01>

<https://{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/deployments/{DEPLOYMENT-NAME}?api-version=2023-04-01>

<https://{ENDPOINT}/language/:analyze-conversations?api-version=2023-04-01>

<https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&to=<languagecode>>