CHAPTER 3

Storing, Interpreting, and Visualizing Data

Think of your AI solution as a high-performance jet engine; it doesn't matter how sleek the design is if you're pumping low-grade fuel. And data isn't just important—it's the jet fuel that determines whether your AI soars or sputters.

Let's say you're building a hospital readmission predictor: storing patient records in Azure SQL Database keeps them query-ready, interpreting lab results with Azure ML helps you spot hidden risk factors, and Power BI dashboards turn those insights into ER staffing decisions. But if you get this trifecta wrong, you'll end up with a model that either hallucinates diagnoses or drowns in HIPAA violations.

This chapter is your guide to avoiding those disasters. You'll learn how to pair Azure's tools like a pro, using Azure Data Lake to tame messy Internet of Things (IoT) sensor data before funneling it into your PyTorch models, and understanding how the choice between Azure Cosmos DB and Blob Storage can make or break your chatbot's response time. I'll also show you how to tackle real-world headaches (like visualizing 10 TB of retail foot traffic data) and equip you with exam-ready skills, from encrypting training datasets to slashing latency with Azure Cache for Redis. By the end, you'll be managing your data as a strategic asset.

Data Storage and Management in Azure AI

The effectiveness of Azure AI services—from machine learning models such as those in Azure ML to cognitive services like Computer Vision and Text Analytics—depends on the availability, quality, and organization of data. Having a sufficient amount of high-quality, relevant data is critical for training accurate and reliable models, and secure, well-managed storage is a necessity to ensure that it is readily accessible for training and inference, supporting the development of robust solutions.

In this section, we'll examine the various storage options you can use when working with AI services in Microsoft Azure, along with best practices for managing your data effectively. It's easy to assume that when data is stored in a given location, it can simply remain there, but we also need to abide by proper practices in managing it.

Choosing Storage Options for AI Solutions

Which storage solution to use when architecting an Azure AI solution is an important decision, as this will directly impact the performance, scalability, cost, and overall success of your application. Azure provides a wide array of storage services that are designed to handle different data types, usage patterns, and application requirements. For instance, an AI-driven recommendation engine might rely on transactional structured data from a SQL database for user records, combined with unstructured image or text data stored in Blob Storage for content-based analysis. This section will guide you through the options so you can make the right storage choices for your AI projects on Azure.

Understanding data types and requirements

The first step in selecting a storage option for an AI solution is understanding the nature of your data and the specific requirements of your application. Data can be broadly classified into two types: structured and unstructured.

Structured data. This includes data that's organized in a predefined manner, typically in tables. For AI solutions that utilize structured data, such as customer information, Azure SQL Database is a suitable choice for relational data needs. On the other hand, for globally distributed, multimodel databases, Azure Cosmos DB is ideal. It ensures low-latency access to data worldwide, making it perfect for applications that require high availability and geo-replication, and supports the following data models, all of which are commonly used for semi-structured data:

Document model
: This stores data as JSON-like documents, making it suitable for applications that handle semi-structured data with nested structures, such as user profiles and content management systems. This model provides rich query capabilities and indexing on document fields, allowing for the efficient retrieval and manipulation of complex data structures. This helps improve processing times.

Key/value model
: This stores data as a simple collection of key-value pairs, in which each key is unique and associated with a value. This model is optimal for applications requiring high-speed lookups with straightforward data access patterns, such as caching and session management. Its simplicity ensures fast read and write operations, making it a go-to choice for performance-critical applications.

Graph model
: This represents data as vertices (nodes) and edges (relationships), making it ideal for applications that involve complex relationships and traversals, such as social networks and recommendation engines. This model allows for efficient querying of interconnected data, which enables advanced analytics and insights into how entities relate.

Column-family model
: This organizes data into rows and columns that are grouped into column families. It's well suited for applications that require high write throughput and can handle large volumes of sparse data, such as time-series data and log analysis. Each row can have a different set of columns, which provides flexibility in data storage and allows for efficient handling of varying data schemas.

By supporting these diverse models, Azure Cosmos DB allows developers to choose the most appropriate data structure for their specific application needs, thus ensuring optimal performance and scalability.

Unstructured data. This includes data such as images, videos, and documents that do not fit neatly into tables. Machine learning workloads often involve unstructured text (such as product reviews and chat transcripts), audio data (such as voice recordings), or large image and video files for computer vision tasks. Azure Blob Storage provides a cost-effective and scalable solution for storing large volumes of unstructured data. It also supports various access tiers—hot, cool, and archive—allowing organizations to optimize costs based on how frequently their data is accessed. The *hot tier* is ideal for data that's accessed frequently. It offers the lowest access latency and has the highest storage cost. On the other hand, the *cool tier* is suited to data that's infrequently accessed and stored for at least 30 days; it provides a balance between lower storage costs and higher access costs than those of the hot tier. Finally, the *archive tier* is intended for data that's rarely accessed and stored for at least 180 days; it offers the lowest storage cost and the highest access latency and retrieval costs. By categorizing data into these tiers, users can achieve cost efficiency while maintaining access to their data according to their specific needs. Figure 3-1 shows one such architecture.

The workflow begins with the storage of images in Azure Blob Storage, followed by the triggering of an Azure Function that processes the images using the Azure Computer Vision service. The analysis results and metadata are then stored in Azure Cosmos DB.

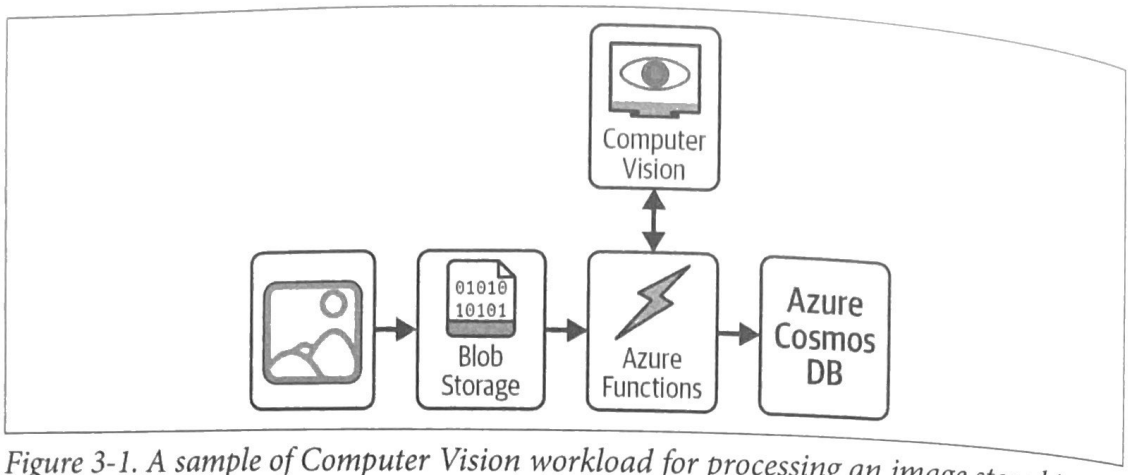

Figure 3-1. A sample of Computer Vision workload for processing an image stored in Blob Storage

This architecture leverages the strengths of each Azure service to provide a robust solution for computer vision workloads. Using Blob Storage for large file storage, Azure Functions for scalable processing, and Cosmos DB for fast data retrieval enables efficient handling of large volumes of image data and analysis tasks. An architecture like this is essential for processing and analyzing large datasets, as it provides optimized storage, quick access to data, and scalable compute resources—key to maintaining performance and responsiveness as workloads grow.

Azure Data Lake Storage (ADLS) offers a highly scalable and secure environment for big data analytics. ADLS integrates seamlessly with platforms like Azure Databricks and Azure HDInsight, thus supporting complex analytics workloads and ensuring optimized data processing and storage.

In many hybrid scenarios, you might work with structured data in Azure SQL for user management and unstructured image/text data in Azure Blob Storage for AI-driven tasks like object detection and sentiment analysis. Properly classifying the data types will make it easier to identify the most appropriate storage solution for each data subset, for optimal performance and cost management.

Performance and scalability

The storage solution you choose must scale with your application's needs and deliver the performance you require. Key aspects to consider when choosing a storage solution include throughput, latency, and the solution's ability to handle real-time or near-real-time data processing.

In terms of performance, for applications that require low-latency data access, Azure Cosmos DB guarantees single-digit millisecond latencies, making it suitable for real-time analytics and high-transaction workloads. On the other hand, Azure Disk Storage (which includes Premium SSD and Ultra Disks options) offers high input/output operations per second (IOPS), a performance metric that measures how many read

and write operations a storage device can perform per second. It also delivers low latency, making it well suited for compute-intensive applications such as SAP HANA and Microsoft SQL Server.

You can meet scalability requirements in a similar way with solutions like Azure Blob Storage and ADLS, which can handle massive amounts of data and scale automatically to accommodate growing data volumes and user requests without compromising performance. These services are designed to provide high throughput and support parallel data processing, thus ensuring that data is managed efficiently even as volumes increase.

Security and compliance

When evaluating storage solutions, you must ensure that they have proper data security methods in place and that they comply with regulatory standards. In this context, key aspects to consider when choosing a storage solution include its encryption, access control, and adherence to industry regulations.

Azure storage services offer robust encryption capabilities for data at rest and in transit. Services like Azure Storage and Azure SQL Database support automatic server-side encryption by using platform-managed keys or customer-managed keys, ensuring that data is protected against unauthorized access.

Azure storage solutions are also designed to comply with various industry regulations and standards, such as GDPR, the HIPAA, and the Payment Card Industry Data Security Standard (PCI DSS). Microsoft Purview provides comprehensive data governance capabilities and thus enables organizations to properly manage data security, privacy, and compliance. It includes features like data classification, lineage tracking, and policy enforcement to ensure data handling practices meet regulatory requirements.

Ease of integration

You must also facilitate the integration of relevant services to optimize your AI solutions. Azure provides robust integration capabilities that will simplify your efforts to connect various data services and AI tools. For instance, Azure Blob Storage is highly compatible with numerous AI services, and it enables easy access to and processing of unstructured data like images, videos, and documents.

Many Azure AI services, including Azure Cognitive Services and Azure ML, can directly consume data from Blob Storage, which can help streamline workflows. Azure ML also works well with Azure Data Lake Storage. Both of these storage options can store training data, models, and the outputs of experiments, seamlessly integrating with the Azure ML workspace. This flexibility allows data scientists to choose the most appropriate storage solution for the nature of their data and their specific project needs.

Cost

Cost is a critical factor to take into account when selecting storage services. Key cost-related aspects include storage volume, access patterns, and your data retention requirements. Azure offers various pricing tiers to accommodate different use cases. For instance, as described earlier, Azure Blob Storage provides hot, cool, and archive tiers that are suited to users with different access and cost requirements.

Frequently accessing data and processing transactions can significantly impact your costs. For example, Azure Cosmos DB charges based on throughput and storage consumed. High transaction rates can increase costs, so you'll want to optimize your queries and manage your throughput appropriately. For big data scenarios, Azure Data Lake Storage offers a cost-effective solution for storing large volumes of data while supporting advanced analytics and machine learning workloads. Understanding these cost structures and aligning them with your data access patterns can help you manage your expenses and optimize your resource utilization.

Carefully considering integration capabilities and cost implications will help you select the most suitable Azure storage options for your AI solutions, ensuring efficiency, scalability, and cost-effectiveness.

Azure storage options

Because data is the cornerstone of any AI project, it's important to understand the different types of storage solutions and services that are available in Azure when working with AI. Let's take a closer look at each one and explore when to use them in developing AI services.

Azure Disk Storage. Azure Disk Storage provides high-performance, durable block storage for Azure VMs. It offers multiple types of disks—including Ultra Disk, Premium SSD v2, Premium SSD, Standard SSD, and Standard HDD—which cater to different performance and cost requirements. Ultra Disk storage is designed for high-end workloads requiring submillisecond latencies, Premium SSD v2 provides a balance of performance and cost for transactional workloads, and Standard SSD and HDD are cost-effective options for less demanding applications.

For AI workloads involving intensive training (e.g., deep learning on large image datasets), using Premium SSD or Ultra Disk can significantly reduce training times by providing higher IOPS and lower latency. For instance, GPU-enabled VM clusters often benefit from Ultra Disk in scenarios requiring sustained high throughput. While Standard SSD or HDD might suffice in development or staging environments, production training pipelines usually require premium tiers to avoid bottlenecks during parallel data read/write operations.

Benchmark tests on Azure have shown that Ultra Disk with a GPU-enabled VM can achieve subsecond loading times for large batches of image data, making this an ideal

choice for scenarios like computer vision or NLP model training, where quick access to data chunks is critical. It's also useful for mission-critical applications that demand consistent performance, such as databases (e.g., SAP HANA, SQL Server, Oracle), enterprise applications (e.g., Microsoft Dynamics 365), and big data analytics. This combination provides excellent support for high-performance workloads, including real-time transaction processing and large-scale data analytics.

Azure Disk Storage is often used in data-intensive AI and analytics workloads that require rapid, consistent access to large datasets. This includes training deep learning models, where data is read and written at high speeds. For example, Moody's Analytics utilizes Azure Disk Storage to increase storage capacity and performance in VM scale sets. Selfhelp Community Services leverages Premium SSDs for enhanced VM performance and standard SSDs for high availability in Kubernetes clusters. Teradata Corporation benefits from Azure Disk Storage's scalability to support its large-scale analytics workloads.

Azure Blob Storage. Azure Blob Storage is an object storage solution that's optimized for storing vast amounts of unstructured data, such as text and binary data. It supports various types of blobs, including block blobs, append blobs, and page blobs, making it versatile for different data storage needs. With built-in data tiering, it helps manage costs by automatically moving data between hot, cool, and archive tiers based on access patterns.

Blob Storage is particularly suitable for storing the large datasets that are used in training machine learning models, including images, videos, audio files, and logs. It integrates with Azure Data Lake Storage, making it a robust choice for big data analytics, data lakes, and data warehousing solutions. It's also ideal for real-time analytics workloads and data preprocessing tasks for machine learning, thanks to its scalability and high availability. Many AI-driven applications, such as those for image recognition (e.g., Azure Cognitive Services) and NLP (e.g., Azure Text Analytics), utilize Blob Storage to efficiently manage and access large training datasets.

Azure Data Lake Storage. Azure Data Lake Storage combines the capabilities of Azure Blob Storage with features that are specifically optimized for big data analytics, such as a hierarchical namespace and enhanced security. It's designed to handle large volumes of both structured and unstructured data, providing high throughput and low latency. This service is ideal for scenarios requiring extensive data analytics and processing capabilities, including large-scale AI model training, big data applications, and enterprise data warehousing. It also supports extract, transform, load (ETL) processes, data integration, and advanced analytics.

The hierarchical namespace in Azure Data Lake Storage is particularly valuable for machine learning workflows that involve organizing training data into nested folder structures (e.g., */datalake/processed/year=2025/month=03/*). This structure simplifies

data versioning, partitioning, and retrieval, especially when you're dealing with iterative model development. For instance, storing model artifacts in a dedicated hierarchy (e.g., */datalake/models/<model_version>/*) can streamline MLOps processes, making it easier to track model lineage and maintain older versions for rollback or auditing.

Azure Data Lake Storage is also well suited for AI and machine learning projects that require significant data processing capabilities. It's commonly used in conjunction with Azure Databricks, Azure Synapse Analytics, and HDInsight for processing big data and training ML models. AI projects involving predictive analytics and large-scale data mining also rely on Azure Data Lake Storage for its ability to manage and process vast amounts of data while minimizing overhead. Finally, companies that use Azure Synapse Analytics for integrated analytics across their AI models benefit from its robust storage capabilities.

Azure Files. Azure Files provides fully managed file shares in the cloud, which are accessible via the server message block (SMB) and network file system (NFS) protocols. It supports features such as snapshot-based backups, geo-redundancy, and integration with Microsoft Entra ID. Azure Files is commonly used for sharing datasets and tools among applications, migrating legacy workloads to the cloud, and supporting lift-and-shift scenarios. It's particularly useful for applications that require the storage of shared files that need to be accessible by multiple virtual machines.

Azure Files also facilitates the sharing of datasets and intermediate results among various AI services, such as data preprocessing pipelines, model training, and inference applications. An example of this might be an environmental engineering company using Azure File Sync to synchronize data between its on-premises servers and Azure, ensuring the continuous availability of training datasets for its AI models, even during cloud migrations.

Azure NetApp Files. Azure NetApp Files provides high-performance file storage that leverages NetApp's enterprise-grade technology. It offers high throughput, low latency, and support for SMB, NFS, and dual-protocol volumes. This service is well suited for high-performance computing applications, including AI workloads that demand rapid access to large datasets. It's particularly beneficial in industries like genomics, media, and entertainment, where data-intensive applications require efficient storage solutions to maintain workflow speed, reduce latency, and optimize resource usage, enabling complex computations and real-time data processing.

Azure NetApp Files is also ideal for AI models that require high-speed data access, such as real-time analytics, simulations, and other performance-sensitive applications. For instance, genomics workflows like DNA sequencing and analysis often rely on Azure NetApp Files for fast, reliable storage. Similarly, financial institutions use it to support real-time fraud detection and risk analysis models.

Azure File Sync. Azure File Sync allows you to synchronize files across Azure Files and on-premises Windows Servers, supporting hybrid storage environments. It includes features like cloud tiering, which automatically offloads infrequently accessed files to Azure to optimize storage usage and reduce costs. This service is commonly used in scenarios that require data synchronization between on-premises environments and Azure, such as AI model training and inference in hybrid setups. It also supports business continuity by ensuring data availability across different locations.

Azure File Sync is especially useful for AI projects that demand consistent data availability across hybrid environments to ensure that their models can access up-to-date data, whether they are running in the cloud or on-premises. Enterprises with hybrid cloud strategies also use Azure File Sync to maintain data consistency across AI development and production environments. For instance, a healthcare provider might use it to synchronize patient records between on-premises systems and cloud-based AI analytics platforms.

Azure Stack Edge. Azure Stack Edge is a cloud-managed physical device that provides compute, storage, and AI capabilities at the edge. It includes field-programmable gate array (FPGA) or GPU hardware to accelerate machine learning workloads and supports offline and low-latency applications. It's commonly used in scenarios requiring real-time processing, such as IoT, manufacturing, and remote locations where connectivity might be intermittent. It also enables preprocessing of data before sending it to Azure, reducing latency and bandwidth usage.

Azure Stack Edge is ideal for edge AI applications that require immediate data processing, such as predictive maintenance, autonomous systems, and smart city solutions. It allows AI models to run locally at the edge so they can provide insights and actions in real time. For instance, manufacturing plants use Azure Stack Edge for predictive maintenance by processing sensor data locally to predict equipment failures. Similarly, IoT applications in smart cities leverage it for real-time traffic management and environmental monitoring.

Azure Data Box. Azure Data Box services facilitate the transfer of large volumes of data to Azure, which is particularly useful when network transfer is impractical due to bandwidth limitations or sheer data size. It's frequently used to move large datasets into Azure Blob Storage or Azure Data Lake Storage for processing and analysis—an essential step for AI projects that require significant historical data for model training. Azure Data Box is also well suited for large-scale data migrations, such as moving entire data warehouses or legacy archives into the cloud for AI-driven analytics. Organizations undergoing digital transformation often rely on it to enable cloud-based AI training at scale. For example, a retail company might use Data Box to transfer years' worth of transactional data to Azure to use for developing advanced recommendation systems.

Data Management Best Practices

Effective data management is critical to maximize the success potential of your AI projects on Azure. Adhering to the data management best practices described in this section will help ensure that the data your organization uses is well organized, secure, and well suited to working with Azure AI services. We'll start with data governance and cataloging, then consider data quality and preparation, and finally data backup and disaster recovery.

Data governance and cataloging

You need to establish a robust data governance framework so that you can effectively manage data in your organization. Establishing such a framework involves implementing appropriate policies and standards for data usage, security, quality, and compliance. These policies ensure consistent data management, define maintenance standards, and set clear guidelines for appropriate data usage.

Like its predecessor, Azure Data Catalog (which will be officially retired at the end of 2025), Microsoft Purview plays a pivotal role in data governance by enabling organizations to register, enrich, discover, understand, and consume data sources. It helps maintain an organized inventory of data assets, making it easier for stakeholders to locate and utilize relevant data when needed.

In addition, Microsoft Purview extends these capabilities by offering a comprehensive set of solutions for data governance across on-premises, multicloud, and software-as-a-service (SaaS) environments. It provides a unified platform that supports automated data discovery, sensitive data classification, and end-to-end data lineage. It also enables the creation of a holistic, up-to-date map of an organization's data landscape, which is critical for ensuring proper governance and responsible data usage.

This section will cover the key features of Microsoft Purview, including automated metadata management from hybrid sources, data classification using built-in and custom classifiers, and Microsoft Information Protection sensitivity labels. These capabilities ensure consistent labeling of sensitive data across various platforms, such as SQL Server, Azure, Microsoft 365, and Power BI.

By integrating with data catalogs and systems using Apache Atlas APIs, Purview provides a unified map of data assets and their relationships. This integration simplifies data discovery and governance, enabling a single-pane-of-glass experience for managing data across your entire data estate.

Purview also supports secure data sharing both within and between organizations. It offers a centralized platform for managing data-sharing relationships and revoking access to data as needed. Additionally, its data policy features support scalable,

fine-grained access controls, helping organizations maintain compliance with regulatory requirements.

Purview integrates seamlessly with solutions like Profisee Master Data Management (Profisee MDM) as well, enabling organizations to publish and sync metadata changes and governance details. This integration supports a robust master data model by aligning data governance practices with business priorities and enhancing the overall data strategy. Another example is CluedIn, a modern master data platform that unifies disparate data sources through AI-driven matching and data quality workflows. CluedIn can sync its enriched master data definitions and lineage with Purview for centralized governance.

Modern data governance solutions like Microsoft Purview provide several benefits, including the ability to handle the complexities of large data estates, support AI-enabled experiences, and promote a culture of data governance and protection. These solutions enable organizations to align their data governance practices with measurable business objectives, demonstrate business value, and ensure that data insights are at the core of their decision making.

AI-specific governance scenarios can also include tracking model lineage. For instance, Microsoft Purview can document which datasets were used to train a particular machine learning model, along with any transformations that were applied during data preprocessing. This ensures transparency and compliance, especially when sensitive datasets are involved (e.g., datasets containing personally identifiable information). Purview policies can also restrict access to training data based on labels like "confidential" or "PII," thus ensuring that only authorized data scientists can view or modify that data.

By leveraging these advanced data governance and cataloging tools, organizations can effectively manage their data, ensure compliance, and unlock the full potential of their data assets in support of AI and other advanced analytics initiatives.

Step-by-step guide to cataloging AI data with Microsoft Purview. Cataloging AI data using the Microsoft Purview portal involves several steps. Here's a comprehensive guide to help you through the process:

1. Create a Microsoft Purview account, if you don't have one (see Figure 3-2):
   a. Log in to the Azure portal using your Azure account.
   b. Search for "Microsoft Purview" and select Create to set up your account.
2. Provide the following account details:

   Subscription
   Choose your Azure subscription.

Resource group
: Select an existing resource group or create a new one.

Account name
: Enter a unique name for your Purview account, using the format ***`Your_Initials`*`-purview-acc`**. (Note that in all cases in these instructions, you should replace ***`Your_Initials`*** with your own initials.) Ensure that it does not contain spaces or special characters.

Region
: Choose the appropriate region. This should match your Microsoft Entra ID home region, because Purview accounts can't be moved to different regions after they're created.

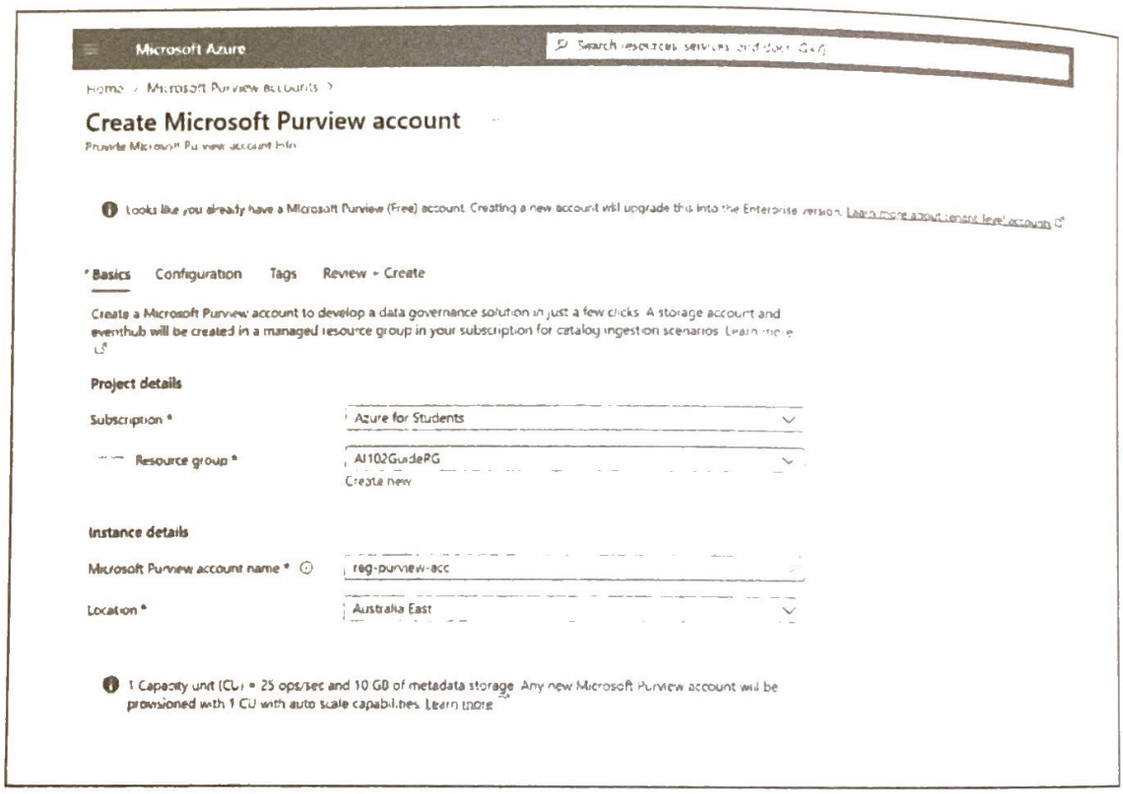

Figure 3-2. Creating a Microsoft Purview account

3. Set up and configure networking:
   a. Set up your network connectivity by deciding whether to allow access to all networks or use private endpoints for enhanced security. In this case, you'll allow access to all networks for simplicity.
   b. Add relevant tags, such as "Purview environment" with values like "production," "test," or "development." You can skip this step for now.
   c. Review your settings and click Create to set up the account.
4. Launch and access the Microsoft Purview governance portal:

a. Navigate to the classic Purview portal by clicking your new Purview account

b. Navigate to Overview, then click "Open the Microsoft Purview governance portal."

5. Create and register data sources:

   a. Navigate to Data map → Data Resources, click Register, and search for and select Azure Blob Storage.

   b. Name the data source **`Azure-Blob-`*`Your_Initials`***.

   c. Select the Azure subscription associated with your account.

6. Select the storage account name based on the storage account you have created:

   a. Make sure that the domain reflects the Purview account you created.

   b. Click Register after checking all the details.

7. Manage credentials and access:

   a. Navigate to the Management tab in the Purview governance portal.

   b. To grant Purview access to the key vault, navigate to Credentials and click New.

   c. Create a new key vault.

   d. Enter the name of the key vault as **`ai102-key-vault-`*`Your_Initials`***.

   e. Select the Azure subscription you are using.

   f. Select the key vault name you just entered.

   g. Click Create to create the new key vault.

   h. Navigate to the Credentials section and New.

   i. Name the credential **`ai102 -`*`Your_Initials`*`-credential`**.

   j. Make sure that the domain reflects the Purview account you created.

   k. Select "API key" as the authentication method.

   l. Select the newly created key vault as the Key Vault connection.

   m. Use an API key generator (such as the API Key Generator) to generate an API key.

   n. Use the API key as the secret name for the new credential.

   o. Verify that connections to the data sources are properly established and validated in Purview Studio.

8. Create and configure scans:

   a. Navigate back to the data source you've created and click "Scan rule sets" under "Source management."

b. Click New to create a new scan rule set.

c. Select Azure Blob Storage as the data source type.

d. Set the scan rule set name as **`ai102-rule-set-`*`Your_Initials`***.

e. Select your Purview account domain as the domain.

f. Click Continue.

g. Select all types of files to be scanned.

h. Select all the system rules that are available.

i. Click Create.

j. Navigate to Blob Storage under "Data sources," then click New Scan and run the scan to discover metadata across the registered data sources. This will populate the data map with discovered assets.

9. Discover and classify data:

   a. Go to the Data Map and click Classifications to view and manage data classifications.

   b. Purview comes with over 200 built-in classifiers that you can apply during the scanning process to automatically classify data based on predefined patterns. Select and apply one or more of these.

10. Analyze data lineage and impact:

    a. In the Purview portal, go to the Data Catalog section, select the container being used, and select Data Lineage.

    b. Select a specific data asset to view its lineage. The lineage view shows how data flows from its origin through various transformations to its final destination.

    c. Use the interactive map to visually explore data relationships and dependencies.

    d. In the Data Lineage view, use the impact analysis tool to assess how changes to data sources or processes affect downstream systems and reports.

    e. Document and manage dependencies to ensure that data changes do not disrupt business processes.

11. Monitor and govern data usage:

    a. In the Purview portal, navigate to the Insights section to access monitoring dashboards.

    b. Set up dashboards to track data usage metrics, compliance status, and governance health.

And with that, you've established your first Microsoft Purview solution! This foundational experience is essential for managing data stewardship effectively and ensuring responsible, well-governed data use in your AI solutions.

Data quality and preparation

Ensuring the quality of your data and preparing it effectively are essential parts of optimizing your AI services on Azure. Data quality encompasses several dimensions, including accuracy, completeness, uniqueness, consistency, timeliness, and validity. Maintaining high standards across each of these dimensions ensures that your data accurately reflects reality, is complete (meaning there are no missing values), is unique (meaning there is no duplication), is consistent across different data sources, is timely and up to date, and conforms to defined rules and standards.

To help you implement effective data quality and preparation practices, Microsoft Purview provides built-in tools for monitoring data quality. You can track quality metrics during both the ingestion phase and the processing phase through scheduled or on-demand scans, ensuring that you have reliable and trustworthy data to drive decision making. The tools allow you to monitor key indicators such as the number of ingested rows, rows containing null values, and schema validation failures, helping you maintain data integrity.

Using Azure Data Factory for data preparation. Azure Data Factory (ADF) is a powerful tool for use in data cleaning and transformation processes. It integrates with Microsoft Power Query Online, enabling users to visually clean and transform data without writing any code. This is particularly useful for data engineers and citizen data integrators who need to explore and prepare datasets quickly and efficiently.

ADF supports various data formats and authentication types, making it adaptable to different data sources. It translates Power Query M functions into Apache Spark code behind the scenes, facilitating large-scale data preparation and ensuring that the data is ready for downstream analytics and machine learning tasks.

Using Azure Databricks for data preparation. Azure Databricks provides a collaborative environment where data scientists and engineers can preprocess data before training AI models. It supports various data preparation tasks, such as handling missing values, removing duplicates, and standardizing data formats. It also enables exploratory data analysis (EDA), which can help you understand the characteristics and distributions of your data and identify and address data quality issues.

With Azure Databricks, organizations can transform raw data into structured formats that are suitable for AI model training. This helps ensure that the training data is accurate, consistent, and free from biases or errors, which in turn leads to better model performance and more reliable outcomes.

By leveraging tools like Azure Data Factory and Azure Databricks, organizations can enhance their data quality and preparation processes and thus ensure that their AI services on Azure will be built on a foundation of high-quality, well-prepared data. This approach not only improves the accuracy and reliability of AI models but also supports more informed and successful decision making.

Data backup and disaster recovery

Azure provides comprehensive solutions to help you ensure that your AI applications and data are protected against data loss and that you can quickly recover them in the event of a disaster. This is essential, because you must implement robust data backup and disaster recovery strategies to maintain the availability and integrity of your AI services.

Data backup with Azure Backup. Azure Backup is a scalable and secure solution that's designed to protect your data by creating snapshots at specified intervals. It supports a variety of workloads, including Azure VMs, SQL databases, and on-premises data. These regularly scheduled backups ensure that point-in-time snapshots will be available for data recovery, which is critical for maintaining the consistency and integrity of AI services. Azure Backup also allows for long-term retention policies, enabling data to be retained and recovered even after extended periods, which is essential for compliance with regulatory requirements.

Automating backup processes with Azure Policy, Microsoft PowerShell, or the Azure CLI helps maintain consistency and alignment with organizational standards. This automation reduces the risk of human error and guarantees that backups run regularly, without manual intervention. Azure Backup also supports application-consistent backups, which helps preserve application integrity during the backup process.

Disaster recovery with Azure Site Recovery. Azure Site Recovery (ASR) replicates workloads that run on physical and virtual machines (both in the cloud and on-premises) from a primary site to a secondary location. It allows you to define replication policies so that you can manage recovery point objectives (RPOs) and recovery time objectives (RTOs), helping you meet your business continuity requirements. The service supports seamless failover and failback operations, minimizing downtime during disaster recovery scenarios.

Geo-redundant storage (GRS) is another key feature: it replicates data across multiple geographic regions, so data remains available even if one region is compromised. This enhances resilience and disaster recovery capabilities by providing multiple recovery points.

Implementing and managing backup and recovery strategies. Azure provides a centralized management interface for backup and disaster recovery operations through the Azure portal. This interface allows you to define, monitor, and manage policies for enterprise workloads across hybrid and cloud environments. Setting up monitoring and alerting mechanisms helps you track the health and performance of your backup and disaster recovery setup, ensuring that you'll be able to detect and resolve on a timely basis any issues that may arise during replication or failover processes.

Compliance and security are integral to Azure's backup and disaster recovery solutions. Built-in security features include multifactor authentication, RBAC, and encryption, all of which protect your backup environment from unauthorized access and ransomware attacks. Additionally, Azure's solutions comply with a wide range of security and privacy regulations, providing you with peace of mind because your data is protected.

Testing and validation. You must perform regular testing to validate the effectiveness of your backup and disaster recovery strategies. Conducting simulations of disaster scenarios will help ensure that your team is prepared and that the processes work as expected. Be sure to provide all relevant stakeholders with detailed documentation and runbooks outlining the backup and recovery procedures, so they are aware of the steps they need to take during a disaster and will be able to execute them efficiently. Having efficient backup and recovery procedures in place is crucial to minimizing downtime, preventing data loss, and ensuring business continuity. Making sure stakeholders have clear guidance will enable them to respond quickly and effectively.

By following this guidance and leveraging Azure's comprehensive tools, organizations can safeguard their AI services against data loss and ensure swift recovery in the event of a disaster. This approach not only protects critical data but also maintains the continuity and reliability of AI applications.

Data Interpretation for AI Solutions

In this section, we'll discuss how you can leverage Azure AI services to assist you with data analysis, where data analysis can fit within the AI solution workflow, and how to choose the right models for training and selection within Azure AI.

Leveraging Azure AI for Data Analysis

Data analysis is a crucial component of modern AI workloads. You must be able to process and analyze data effectively to extract actionable insights and make informed, data-driven decisions. This involves creating event-driven architectures and building unified solutions that ensure seamless integration and standardization across different services and platforms.

With the rise of LLMs and foundation models such as GPT-based architectures, Azure now offers the Azure OpenAI Service. This enables you to perform advanced NLP tasks like summarization, content generation, and semantic search directly against data stored in services such as Cosmos DB or Azure Blob Storage. Integration patterns often involve using Azure Data Factory or Logic Apps to orchestrate data movement into a format that's suitable for prompting these models, then storing the resulting inferences (structured or unstructured) for further analytics.

Creating event-driven architectures

Event-driven architectures are designed to respond to events or changes in data in real time. By utilizing services such as Azure Functions, organizations can create highly responsive systems that process data as it arrives. Azure Functions supports serverless computing, enabling small pieces of code to be executed in response to events without the need for the organization to manage infrastructure.

For instance, data changes in Azure Cosmos DB can trigger an Azure Function to perform real-time analysis. This setup is particularly useful for scenarios such as monitoring social media feeds, analyzing sensor data, and processing transaction logs. When a new event is recorded in Cosmos DB, an Azure Function can analyze the data, detect anomalies, and generate insights or alerts. This real-time processing capability helps organizations react promptly to changes and make informed decisions based on up-to-date information.

Building a unified solution

Creating a unified solution involves integrating various Azure services to work seamlessly together so that data flows smoothly among different components of the architecture. It also facilitates standardization, making it easier to manage and scale AI workloads.

Azure Synapse Analytics plays a pivotal role in building unified solutions. It combines big data and data warehousing capabilities, enabling organizations to perform comprehensive analysis of both structured and unstructured data. Synapse Analytics can integrate with Azure Data Lake Storage for scalable storage and Azure Databricks for advanced data processing and machine learning tasks. These integrations ensure that data is readily available for analysis and insights can be generated efficiently. Efficient data access and processing are essential for minimizing latency, optimizing resource usage, and accelerating insight generation, which in turn allows businesses to make data-driven decisions more effectively.

Azure ML is another critical component of a unified AI solution. It provides a platform for building, training, and deploying machine learning models. By integrating Azure ML with services like Synapse Analytics and Databricks, organizations can streamline their machine learning workflows, from data preparation to model

deployment. This ensures that models are trained on high-quality data and can be deployed quickly to provide real-time predictions and insights.

To further enhance a unified solution, organizations can use Azure Logic Apps to automate workflows and orchestrate processes across different services. Logic Apps enable the automation of complex workflows by connecting Azure services with external systems. For example, they can automate the process of extracting data from Cosmos DB, triggering Azure Functions for real-time analysis, and storing the results in Synapse Analytics for further processing. This orchestration allows all components of the solution to work together seamlessly in a cohesive and efficient system.

Microsoft Fabric extends these unified analytics capabilities into a single SaaS platform. By leveraging OneLake as a common data lakehouse, Fabric brings together data engineering, real-time analytics, warehousing, and business intelligence in one unified workspace. Its embedded AI features (such as Copilot in Fabric) and native governance (via Purview) streamline insight generation while maintaining compliance across the entire data estate.

Model Training and Selection in Azure AI

Model selection is the process of identifying the most effective model from a set that have been trained on the same data using different configurations or algorithms. This is a crucial step in building an effective AI solution, and Azure provides several tools to help with the process.

Automated machine learning in Azure Machine Learning

Automated Machine Learning (AutoML) on Azure offers a robust solution for automating the process of model selection, enhancing the efficiency and productivity of data scientists and developers. It simplifies many stages of the machine learning model development, from data preprocessing to hyperparameter tuning and deployment.

AutoML automates the iterative and time-consuming tasks involved in building ML models. It evaluates various models and their configurations to identify the best-performing option for a given dataset. This process includes feature engineering, model training, and hyperparameter tuning, all of which are crucial for creating accurate and reliable models.

Using Azure Machine Learning Studio, you can set up AutoML experiments without writing any code. The user-friendly interface allows you to select a data source, define the problem type (such as classification or regression), and choose the target metric for model evaluation. AutoML then runs multiple experiments, trying out different algorithms and hyperparameter combinations to find the optimal solution.

Various advanced features and customization options are available. For instance, you can use AutoML to configure custom featurization settings, manage how missing data is handled, and select specific algorithms to include or exclude from the search space. These options allow you to ensure that the automated process aligns with the specific requirements of your dataset and problem domain. AutoML also integrates with Azure's extensive compute infrastructure, so you can start with local resources and then scale up to Azure Virtual Machines or Azure Databricks clusters as needed. This flexibility ensures that you can handle large datasets and complex models efficiently.

AutoML isn't limited to model selection—it also includes robust experiment tracking and model evaluation tools. You can monitor the progress of your experiments in real time through the Azure Machine Learning Studio dashboard, which provides detailed insights into each model's performance. Metrics such as accuracy, precision, and recall are available to help you make informed decisions about the best model to deploy.

Once you've selected a model, Azure provides you with seamless deployment options. You can deploy models as web services directly from Azure Machine Learning Studio, making it easy to integrate them into applications. Azure also supports MLOps capabilities, which facilitate continuous integration and delivery of machine learning models and thus ensure that your models remain up to date and performant in production environments.

The automation provided by Azure AutoML is particularly beneficial for organizations with limited data science expertise. It democratizes machine learning by enabling domain experts to build and deploy models even if they don't have deep knowledge of ML algorithms and techniques. This accelerates the time to market for these businesses' ML solutions and allows them to focus on solving their core problems rather than on the technicalities of model development.

Evaluation metrics and tools

Evaluation metrics and tools are critical in efforts to assess the performance of AI models, allowing data scientists and developers to make informed decisions about model selection and improvements. Azure Machine Learning offers a robust suite of tools and metrics that you can use to make sure your models meet desired performance standards.

Evaluation metrics. Accuracy is a fundamental metric for classification problems because it represents the proportion of true results (both true positives and true negatives) relative to the total number of cases examined. However, in scenarios where class distribution is imbalanced, other metrics, like precision and recall, become crucial. Precision measures the ratio of true positive results to the total predicted positives, and it highlights the accuracy of positive predictions. Recall (aka sensitivity) measures the ratio of true positives to actual positives, and it indicates the model's ability to identify all relevant instances. The F1 score, which is the harmonic mean of precision and recall, provides a balance between the two, especially when dealing with imbalanced datasets.

The *confusion matrix* is another essential tool that breaks down prediction results into true positives, true negatives, false positives, and false negatives, thus helping to identify systematic errors in model predictions. The *receiver operating characteristic* (ROC) curve and its associated *area under the curve* (AUC) summarize a model's performance across all classification thresholds, thus providing insights into the trade-offs between true positive and false positive rates. Similarly, the *precision-recall (PR) curve* plots precision against recall at various thresholds, offering a detailed view of the balance between these metrics.

Tools in Azure Machine Learning. Azure Machine Learning Studio is a comprehensive platform that supports model development, evaluation, and deployment. It provides built-in tools for visualizing and assessing a wide range of evaluation metrics, and its AutoML feature simplifies model building by automatically selecting the best algorithms and hyperparameters for your data, evaluating multiple models using various metrics, and presenting the results in an intuitive interface.

Prompt flow in Azure ML allows you to customize evaluation flows. You can develop new evaluation methods or modify existing ones, and you can log and aggregate metrics to provide an overall performance assessment. This feature is particularly useful for creating tailored evaluation processes that meet your specific project requirements.

Azure ML also supports foundation models, which are available in the model catalog and can be fine-tuned for specific tasks. You can adapt and evaluate these large-scale, pretrained models using Azure ML's built-in tools, which streamlines the process of customizing them for your use case. Additionally, Azure ML integrates with Azure Monitor to enable real-time monitoring and alerting based on evaluation metrics, which will help you promptly address any performance degradation.

By leveraging these metrics and tools, Azure ML helps you thoroughly and efficiently evaluate AI models. This will help you choose models that are optimized for performance and can be reliably deployed in production environments.

CI/CD for reproducibility

You must implement CI/CD pipelines to ensure reproducibility in your model training and deployment processes. This will guarantee consistency across your deployments and enhance the efficiency and reliability of your AI models in production. To understand how these pipelines function, let's take a closer look at the concepts of continuous integration, continuous deployment, and reproducibility in the context of machine learning:

Continuous integration (CI)
: CI involves the automation of code integration and testing. In the context of machine learning, this means automating the processes of data preprocessing, model training, and validation. By using CI tools like Azure DevOps and GitHub Actions, developers can ensure that each change in the codebase triggers an automated pipeline that runs data sanity checks, trains the model, and performs unit tests. For example, a typical CI pipeline in Azure DevOps might include tasks for setting up the Python environment, installing necessary dependencies, and running training scripts on specified compute resources.

Continuous deployment (CD)
: CD extends CI by automating the deployment of models to production environments. This includes packaging the trained model, creating Docker images, and deploying those images to environments such as Azure Kubernetes Service and Azure Container Instances. The CD pipeline ensures that once a model passes all tests, it's automatically deployed, which reduces manual intervention and speeds up time to market. Azure ML also provides capabilities for creating reproducible pipelines, managing model versions, and deploying models as endpoints for real-time scoring or batch inference.

Reproducibility
: Reproducibility is a key benefit of implementing CI/CD pipelines. By automating the entire ML lifecycle, from data preparation and model training to deployment, CI/CD pipelines ensure that each step is consistent and repeatable. Azure ML supports reproducibility by allowing the definition of reusable pipelines, tracking

all experiments, and capturing metadata and environment configurations. This comprehensive tracking ensures that any model can be retrained and redeployed under the same conditions, which is essential for maintaining model integrity and compliance.

Data Visualization Techniques and Real-Time Analytics

When you're working with AI solutions, it's important to understand the tools and services Azure provides for visualizing data used in AI workloads. Data visualization plays a key role in interpreting both intermediate processing results and final model outputs. You'll use it for a variety of purposes, including analyzing model performance and presenting those insights to stakeholders, helping drive informed decisions and gain buy-in from senior management.

In this section, we'll discuss different data visualization tools and explore how to integrate AI with data platforms.

Introduction to Azure Data Visualization Tools

Let's start by examining the various data visualization tools in Azure that are relevant to AI solutions, and which tools you should use in which situations.

Azure Machine Learning Studio

Azure Machine Learning Studio offers a comprehensive set of capabilities to build, train, and deploy machine learning models. It provides features for visualizing data and assessing the performance of models after iterative training. You can also use its charting and graphing tools to explore data and identify patterns before passing the data into AI models. These tools are also useful for evaluating model performance, including the creation of ROC curves and confusion matrices.

Power BI

Power BI is a powerful business analytics service by Microsoft that allows for the visualization of data (see Figure 3-3). It enables you to share insights throughout your organization and embed those insights into apps or websites. You can use Power BI to integrate your machine learning models with Azure and create interactive reports. These capabilities help you present the results of predictive analytics and monitor solutions in production.

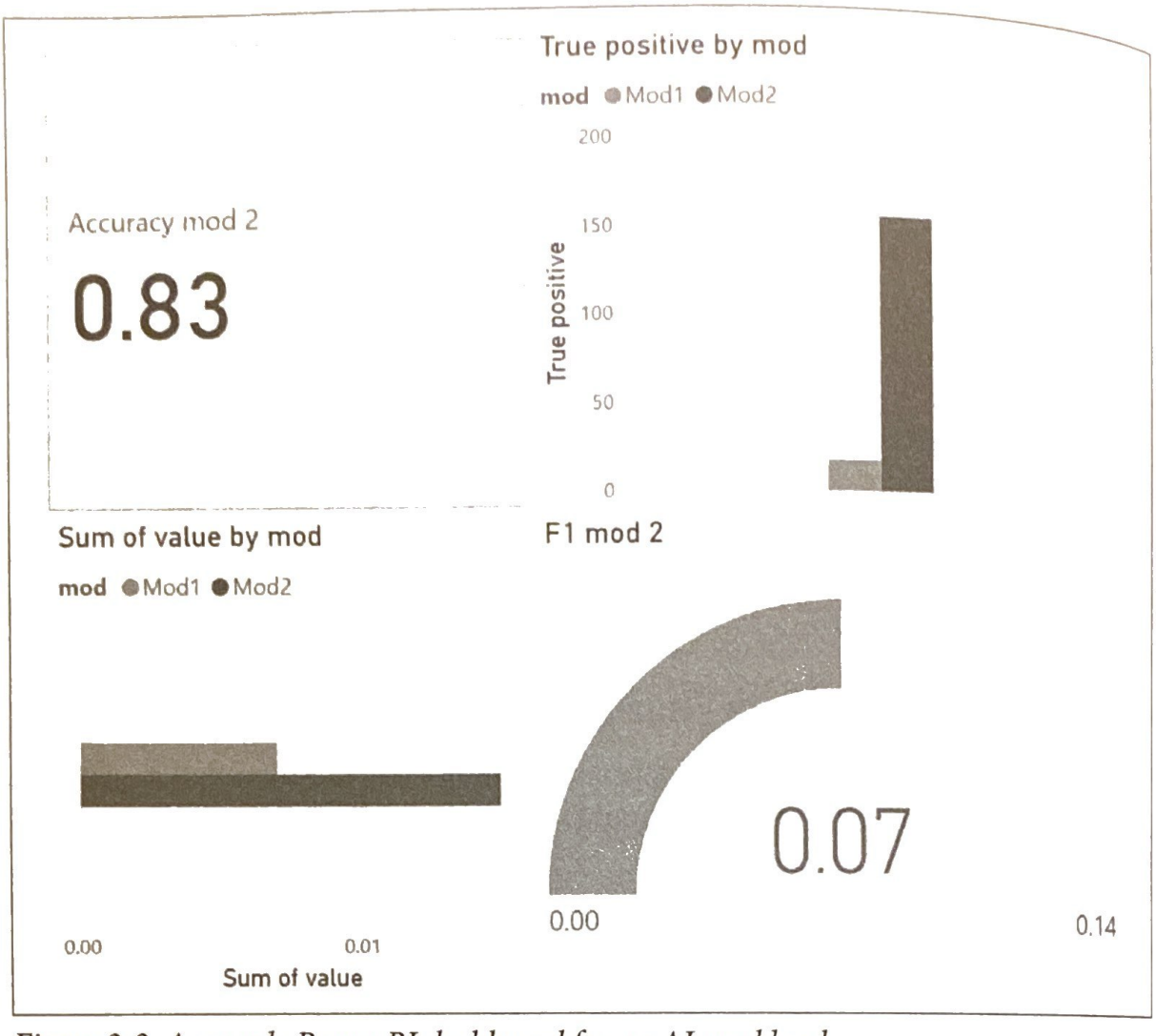

Figure 3-3. A sample Power BI dashboard for an AI workload

Azure Data Explorer

Azure Data Explorer is a fast and highly scalable data exploration service designed for log and telemetry data. You can use it to perform real-time analytics on large volumes of data, and it includes visualization options that allow you to create interactive data reports and dashboards directly from the Azure Data Explorer web interface. This is very useful for analyzing data that is fed into AI models or for visualizing the output of AI services.

Additional integrations

In addition to the tools discussed previously, Azure supports integration with various custom visualization tools and libraries. For example, you can use Python libraries like Matplotlib, Seaborn, and Plotly within Azure Notebooks. Azure can also integrate with external visualization platforms such as Datadog and Splunk, which are useful for visualizing logs, creating custom dashboards and reports, monitoring system behavior, and supporting auditing and troubleshooting.

The ability to surface results through dashboards, alerts, or custom visualizations is crucial for users to interpret data-driven insights and take action, regardless of whether your solution processes data in real time, near-real time, or in batches. We'll turn our attention to the services Azure provides to support different data processing speed requirements, and in particular real-time analytics, in the next section.

Real-Time Analytics and Decision Making

One important decision you'll need to make when designing AI solutions in Azure is how quickly your system needs to process data. In this section, we'll look at how to determine the optimal workload for your needs based on your specific data processing speed requirements. Let's start by considering the different categories.

Real-time analytics

Real-time analytics involves processing data and making decisions within milliseconds to a few seconds after data generation. This is crucial in scenarios where immediate responses are essential, such as fraud detection and monitoring IoT devices for anomalies. Azure offers the following key services to support real-time analytics:

Azure Stream Analytics
: This service is designed for real-time analytics on fast-moving data streams from applications, devices, sensors, and more. It can process data on the fly and trigger actions or alerts based on that data. With built-in temporal operators, such as windowed aggregates and temporal joins, it enables complex event processing with minimal latency.

Azure Event Hubs
: This big data streaming platform can collect, process, and store millions of events per second. It serves as the backbone for data ingestion in many real-time analytics architectures, and it integrates easily with services like Azure Stream Analytics and Azure Functions for further processing.

Azure Functions
: This serverless computing service allows you to run small pieces of code (functions) triggered by various events. It enables real-time processing without requiring you to manage infrastructure, making it ideal for tasks that demand immediate computation and response.

Azure Logic Apps
: This service helps you automate workflows and integrate systems and services using a visual designer. It allows for quick responses to events by automating data flows and service interactions, further enhancing real-time processing capabilities.

Near-real-time analytics

Near-real-time analytics processes data with a short delay—typically from a few seconds to a few minutes. It's suitable for scenarios where immediate action is not critical but timely responses are still necessary, such as social media monitoring or stock market trend analysis. Azure Synapse Analytics, combined with Apache Spark pools, enables near-real-time data processing and analytics by integrating streaming data with structured data from operational databases.

Microsoft Fabric's database mirroring feature lets you continuously replicate external data sources into OneLake in near real time, eliminating the need for complex ETL processes. You can enable mirroring for Azure SQL Database, Azure Cosmos DB, Snowflake, and many other sources directly from the Fabric portal, which provisions system-managed change feed schemas and tables in the source and lands analytics-ready Delta Parquet in OneLake for use across all Fabric experiences. Open mirroring (currently in preview) extends this capability by allowing any application to write change data directly into a mirrored Fabric database using public APIs and Delta Lake formats. In this case, storage replication costs are free up to capacity, and query compute is billed at regular Fabric rates. This setup allows organizations to analyze incoming data as it becomes available and to visualize it through dashboards or reports with minimal lag.

Non-real-time analytics

Non-real-time analytics focuses on deep analysis without the need for immediate decision making. It's often used for batch processing, historical data analysis, and other scenarios where insights are derived from data over extended periods. Services such as Azure Data Factory can orchestrate data pipelines for batch processes, ensuring that data is appropriately collected, transformed, and analyzed at scale.

In each case—whether processing is real-time, near-real-time, or non-real-time—the ability to visualize trends, anomalies, or aggregated results is critical for translating raw data into actionable insights.

Decision framework

Choosing the right Azure services for processing analytics workloads involves a detailed evaluation of several key factors, including latency requirements, data volume and velocity, complexity of analysis, and cost. Each of these elements plays a critical role in determining the most appropriate Azure solutions for building AI-driven applications that meet your specific organizational needs.

Considering latency requirements is paramount when selecting real-time analytics solutions. For applications requiring immediate responses, such as fraud detection and IoT device monitoring, low-latency processing is essential. Azure Stream Analytics is tailored to such scenarios because it offers subsecond latencies with built-in

support for temporal queries and windowed operations. Similarly, Azure Event Hubs provides a robust platform for ingesting large volumes of streaming data with low latency, making it suitable for real-time analytics setups.

The volume and velocity of data are also crucial considerations. High-velocity data streams, which are often seen in applications like social media monitoring and real-time financial analytics, require scalable solutions that are capable of handling large volumes of data efficiently. Azure Event Hubs excels in this area, supporting the ingestion and processing of millions of events per second. In addition, Azure Synapse Analytics, combined with Apache Spark pools, provides a scalable environment for processing both streaming and batch data, enabling comprehensive analytics on large datasets.

The complexity of the required analysis will also influence your choice of tools and services. You might be able to handle simple real-time data processing tasks with Azure Stream Analytics and Azure Functions, but more complex scenarios involving advanced analytics and machine learning might benefit from Azure Synapse Analytics. Synapse integrates deeply with machine learning frameworks and provides powerful data transformation capabilities, enabling sophisticated analysis and model training directly within the data processing pipeline.

Cost is also a critical factor, particularly for organizations with budget constraints. Azure's pay-as-you-go pricing model offers flexibility and scalability, allowing organizations to start small and scale up their infrastructure as needed without significant up-front investment . Services like Azure Logic Apps and Azure Functions provide cost-effective solutions for automating workflows and processing data on demand, ensuring that resources are utilized efficiently and costs are kept under control. Azure Synapse Analytics also supports elastic scaling, helping organizations optimize compute and storage resources based on actual usage and reduce unnecessary expenses.

By carefully evaluating these factors, organizations can choose the most suitable Azure services to meet their specific analytical and operational requirements. This comprehensive approach ensures that AI solutions are not only effective and responsive but also cost-efficient and scalable, providing a solid foundation for real-time decision making and analytics.

Implementing AI in Data Analysis

To create successful AI solutions, you need to be able to implement AI effectively in data analysis workflows. You also need to understand how to integrate such AI services into data platforms so that you can build scalable systems and solutions that comply with industry regulations. In this section, we'll discuss how to do this. We'll also explore a case study of data analysis in action.

Integrating AI with Azure's Data Platforms

Integrating AI with Azure's data platforms involves designing a unified data architecture that ensures seamless data access and movement across various Azure services. Following best practices for data interoperability, security, and compliance is essential for maintaining data integrity and protecting sensitive information. In addition, efficient integration ensures that AI models can process data without delays, enabling real-time insights and optimized decision making while fostering innovation across business operations.

Unified data architecture

To achieve effective AI integration, you must create a unified data architecture. Azure provides a range of services that support this, permitting seamless data flow between different platforms. For instance, Azure Synapse Analytics combines big data and data warehousing capabilities, allowing users to analyze large datasets using both SQL and Spark. This integration also facilitates data preparation and analysis, which is crucial for training AI models. Additionally, you can use Azure Data Factory to orchestrate data movement and transformation across various data sources and thus ensure that data is consistently prepared and available for AI workflows.

Interoperability and integration

Azure's data platforms are designed to be highly interoperable and support a wide range of data sources and formats. You can achieve this interoperability by using services like Azure Data Lake Storage (which provides scalable storage for both structured and unstructured data) and Azure Event Hubs (which allows for real-time data ingestion from various sources). The integration of Azure Databricks further enhances data processing capabilities by providing a collaborative environment in which data scientists can prepare and preprocess data for AI models.

Security and compliance

It's vital that integrated AI and data solutions adhere to security and compliance standards. Azure provides robust security features across its services, and Microsoft Purview offers comprehensive data governance that enables organizations to effectively manage data security, privacy, and compliance. This includes capabilities for data classification, lineage tracking, and access control. Additionally, Microsoft Defender and Microsoft Sentinel provide advanced threat protection and security monitoring to help you safeguard data across your AI and data platforms.

Scalability and flexibility are also fundamental best practices to incorporate when designing a unified data architecture for AI integration. Azure Synapse Analytics is ideal for scalable analytics, while Azure Cosmos DB supports globally distributed databases that handle high data throughput with low latency. These services provide

the necessary resources without compromising performance, ensuring that the data architecture can grow with the organization's needs.

Implementing robust data governance practices is essential to maintaining data quality, privacy, and compliance. Microsoft Purview plays a key role in this by offering tools for data classification, lineage tracking, and policy enforcement. By establishing clear governance policies and monitoring data usage, organizations can prevent unauthorized access and ensure that their data is used responsibly.

Automation and orchestration are also critical for efficient data management. Azure Data Factory automates data pipelines, reducing the need for manual intervention and ensuring that data is consistently transformed and made available for AI models. This not only enhances processing efficiency but also ensures that data workflows are repeatable and reliable.

Finally, leveraging integrated analytics platforms like Azure Synapse Analytics enables organizations to perform both real-time and batch analytics, providing comprehensive insights that drive AI model development and deployment. By integrating analytics directly into their data platforms, organizations can streamline their data processing workflows and ensure that insights are immediately actionable, which in turn supports better decision making and faster innovation.

Case Study of Data Analysis in Action

Let's take a look at a case study of an AI solution in action (see Figure 3-4). This example illustrates how storage, data processing, AI, and data visualization components come together in a complete solution.

This solution is used to process, enrich, and make unstructured data searchable. It involves several steps that integrate various Azure services to transform raw data into a queryable format.

The workflow begins with the ingestion of unstructured data from Blob Storage. This data typically consists of documents and images that need to be processed or "cracked" in the subsequent step. *Document cracking* extracts and converts data from various formats into a form that can be understood and utilized by AI services.

Once the data is prepared, it enters the enrichment phase, in which built-in AI skills are applied. Text analytics is used to extract key phrases, detect language, and identify sentiments within the text. Translator services may be used to convert text into different languages as needed. Computer vision capabilities also play a crucial role in analyzing images, recognizing visual content, and performing OCR to convert image text into searchable data.

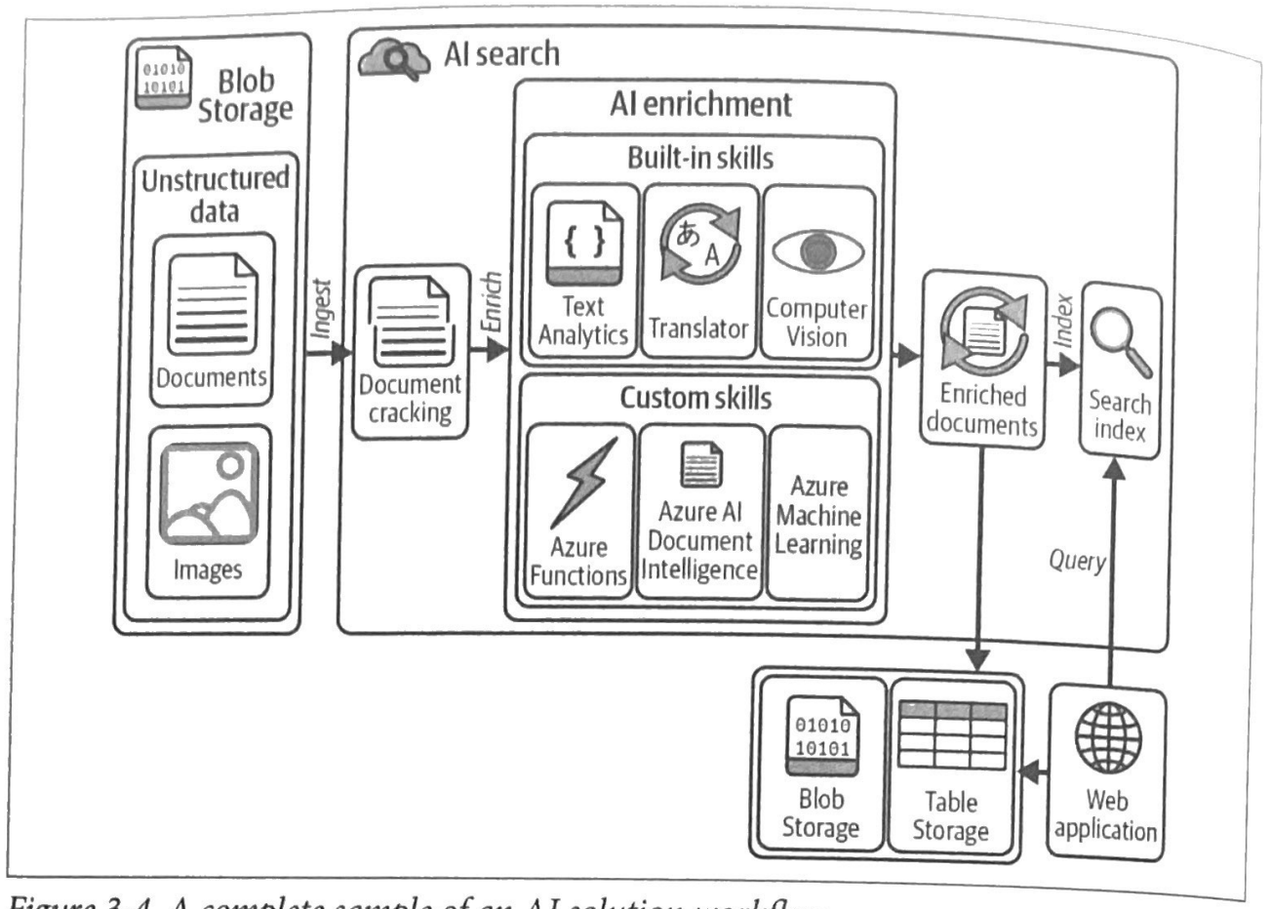

Figure 3-4. A complete sample of an AI solution workflow

In scenarios where the built-in skills don't suffice, custom skills may be required to meet specific business needs. Azure Functions can be used to execute custom code in response to events, while services like Azure AI Document Intelligence can extract text, key-value pairs, and tables from scanned documents. Azure ML can also be used to build and deploy custom machine learning models.

The enriched documents that emerge from this AI-driven process are then used to create a search index—a structured format of the data that enables efficient querying. Having an optimized search index is essential for minimizing response times and ensuring that users can access relevant information quickly, enhancing overall system performance and usability.

In addition to indexing, the enriched data can be projected into various formats and stored in different solutions, such as Blob Storage and Table Storage. This forms the *knowledge store*, a repository of searchable, enriched data.

At the end of the workflow, a web application interfaces with the index to facilitate the query process. Users interact with this application to search for information, and it in turn queries the index and, if necessary, retrieves additional data from the knowledge store to present the results.

Practical: Building an AI-Powered Analytics Dashboard

In this exercise, we will be using the following services to build an AI-powered analytics dashboard:

- Azure Blob Storage
- Azure Language
- Azure Data Factory
- Azure SQL Database
- Power BI

The following sections will walk you through the steps involved in doing this.

Setting Up Azure Blob Storage

You'll begin by setting up Azure Blob Storage to store customer feedback data.

1. Go to the Azure portal, where you'll create a storage account.
2. Select "Create a resource" → Storage → "Storage account."
3. Fill in the required details (subscription, resource group, and storage account name) and review the default settings.
4. Click "Review + create," then Create.

For enhanced security, consider enabling private endpoints so that your storage account is accessible only within your VNet. You can also configure network isolation by selecting "Selected networks" on the Networking tab and specifying the VNets or IP ranges that are permitted to access the account. This will disable or minimize access via the public internet, reducing exposure and helping you meet compliance requirements.

To keep your data secure, compliant, and cost-effective, you should also implement a formal data lifecycle policy. Begin by classifying your data according to its sensitivity and business value. Define retention periods for each category, and move aged data to archive storage or delete it when no longer needed. You can use the Azure Blob Storage lifecycle management rules to automatically tier cold data into the cool and archive tiers and expire blobs after a set number of days. Automate retention and deletion processes to reduce manual effort, and audit all actions through Azure Monitor logs for compliance. Finally, be sure to review and update your policies periodically to align them with changing regulations and business requirements, to ensure that your data remains both accessible and properly managed throughout its lifecycle.

Uploading the Customer Feedback Data

Now that you've set up Blob Storage, you need to upload the customer feedback data:

1. Select your new storage account in the Azure portal, then select "Data storage" → Containers in the lefthand menu.
2. Create a new container, and call it something like *ai102-customer-feedback*).
3. Upload the *customer-feedback.csv* file that's provided with the resources for this chapter in the book's GitHub repository.

Creating an Azure AI Services Language Service

Now that you've uploaded the customer feedback data, you need to create an Azure Language service resource to analyze the data and process the text:

1. In the Azure portal, search for "Language service."
2. Select it from the search results and click Create.
3. Select the default features; you can ignore the custom features for now.
4. Fill in the details (name, subscription, resource group, etc.), as you did in Chapter 2.
5. Click "Review + create," then Create.
6. Take note of the service's endpoint and key, for later use.

As with Blob Storage, you can configure private endpoints or networking settings for your Language service to ensure that all requests come from an approved VNet or IP range. This helps you maintain a secure environment by minimizing public internet exposure.

Creating and Configuring an Azure SQL Database

Now, you'll create an Azure SQL Database, which will automatically generate a table for your data. Here are the steps to follow:

1. In the Azure portal, create a new SQL database by going to "SQL databases" and clicking Create.
2. Specify the server, database name, and compute and storage settings.
3. After you create the database, locate and select it in the "SQL databases" section of the portal, set up the data source, and ensure that it can connect to your Azure Data Factory.

4. Next, you'll use the query editor in the Azure portal or Azure Data Studio to run a SQL script that creates a procedure to increment a unique ID column each time a new row is added to your data table:

```
CREATE SEQUENCE dbo.MySequence
    AS BIGINT
    START WITH 1
    INCREMENT BY 1;

CREATE TABLE CustomerFeedbackAnalysis (
    UniqueId NVARCHAR(20) NOT NULL
        CONSTRAINT DF_CustomerFeedbackAnalysis_UniqueId
        DEFAULT (
            RIGHT(
                REPLICATE('0', 20) +
                CAST(
                    NEXT VALUE FOR dbo.MySequence AS VARCHAR(20)
                ),
                20
            )
        ),
    FeedbackId NVARCHAR(50),
    FeedbackText NVARCHAR(MAX),
    Sentiment NVARCHAR(100),
    PositiveScore FLOAT
);
```

Setting Up Azure Data Factory

Now that you have your Azure SQL Database set up, you can set up an Azure Data Factory (ADF) for data analytics usage. Follow these steps:

1. In the Azure portal, select "Create a resource" → "Data factories" → "Create data factory."
2. Enter the required details:
   a. Subscription: select your Azure subscription.
   b. Resource group: select an existing resource group, or create a new one.
   c. Region: choose the region where you want the Data Factory to be deployed.
   d. Name: enter a unique name for your Data Factory.
   e. Version: select V2 for the latest features.
   f. Networking: set it to connect via the public endpoint.
3. Click "Review + create," then Create.

Creating a Pipeline for Data Movement and Transformation

Next, you'll create a pipeline for moving and transforming data with the help of ADF. Follow these steps:

1. Open your Data Factory Studio by selecting Launch Studio from the Overview page of the resource.
2. Create a new pipeline by selecting New → Pipeline.
3. Select Manage in the side pane, click "Linked services" under Connections, and then click New.
4. Select Azure Blob Storage. Provide an appropriate name for the service, select "Account key" as the authentication type, and select "From Azure subscription" as the connection string. Then, select the Azure subscription you are using and the storage account name you created.
5. Click Create.
6. Next, you'll create another linked service. Select New → Azure SQL Database.
7. Provide an appropriate name for the database and configure it the same way as the Blob Storage, up to "Azure subscription." Specify the server name, select the database name you just created, and set the authentication type as "SQL authentication."
8. In the side pane, select the Blob Storage-linked dataset you created and specify the file path where the output JSON file should be stored.
9. Select the Azure SQL table-linked dataset you created and ensure that the appropriate table has been selected.
10. Under "Move and transform," drag and drop the "Data flow" step onto the pipeline.
11. Click on this item and give it a new name that's appropriate for the data flow.
12. Click Add Source, then click "Output stream" and enter an appropriate output stream name and description. Choose Dataset as the source type, then select the JSON dataset that has been identified.
13. Click the small "+" icon at the bottom right of the source and select Flatten.
14. You will see that a new "flatten" step has been created in the data flow. Click on it and enter an appropriate name. Then, set "Unroll by" to "documents.sentences" and set the "Unroll root" to "{}." For the mapping, follow this structure:
    a. documents.id → FeedbackId
    b. documents.sentences.text → FeedbackText
    c. documents.sentiment → Sentiment

d. documents.sentences.confidenceScores.positive → PositiveScore

e. documents.sentences.confidenceScores.neutral → NeutralScore

f. documents.sentences.confidenceScores.negative → NegativeScore

15. Click the "+" icon at the bottom right of the "flatten" step to add a sink.

16. Provide an appropriate name for the sink. Make the sink type Dataset and set the Azure SQL table as the dataset.

17. Click Publish All.

18. Now you'll create a local script to call the Text Analytics API and pass the feedback data for sentiment analysis. Begin by importing the required libraries and declaring the constants you need for your script (replace the placeholders here with your own values):

```
import requests
import json
import pandas as pd
from azure.storage.blob import BlobServiceClient

endpoint = 'YOUR_TEXT_ANALYTICS_ENDPOINT'
key = 'YOUR_TEXT_ANALYTICS_KEY'
headers = {"Ocp-Apim-Subscription-Key": key}
sentiment_url = f"{endpoint}/text/analytics/v3.0/sentiment"

storage_account_name = 'YOUR_STORAGE_ACCOUNT_NAME'
storage_account_key = 'YOUR_STORAGE_ACCOUNT_KEY'
input_container_name = 'input'
output_container_name = 'output'
input_blob_name = 'customer-feedback.csv'
output_blob_name = 'sentiment-analysis-results.json'
```

19. Then, initialize a Blob Service client as follows:

```
blob_service_client = BlobServiceClient(
    account_url=f"https://{storage_account_name}.blob.core.windows.net",
    credential=storage_account_key
)

input_blob_client = blob_service_client.get_blob_client(
    container=input_container_name, blob=input_blob_name
)
downloaded_blob = input_blob_client.download_blob().readall()
with open(input_blob_name, 'wb') as f:
    f.write(downloaded_blob)
print(f"Downloaded blob '{input_blob_name}' successfully.")

df = pd.read_csv(input_blob_name, encoding='utf-8')
df = df[df['feedback'].str.strip().astype(bool)]
documents = {
    "documents": [
        {"id": str(i), "language": "en", "text": row["feedback"]}
```

```
            for i, row in df.iterrows()
        ]
    }
    print("JSON payload to be sent:")
    print(json.dumps(documents, indent=2))
    response = requests.post(sentiment_url, headers=headers, json=documents)
    if response.status_code != 200:
        print("Error response content:")
        print(response.text)
    response.raise_for_status()
    sentiments = response.json()
    print("Sentiment analysis response:")
    print(json.dumps(sentiments, indent=2))
```

Here, you process customer feedback data that's stored in a comma-separated values (CSV) file within Azure Blob Storage, using Azure's Text Analytics API for sentiment analysis. The script connects to Azure Blob Storage, downloads the customer feedback CSV file, and reads the feedback data into a Pandas DataFrame. Then, it prepares the feedback text for analysis by formatting it into a JSON structure that's compatible with the Text Analytics API. Finally, it sends a POST request to the API endpoint, retrieves the sentiment analysis results, and prints the results to the console:

```
results_filename = 'sentiment-analysis-results.json'
with open(results_filename, 'w') as f:
    json.dump(sentiments, f)
print(f"Sentiment analysis results saved to {results_filename}.")

output_blob_client = blob_service_client.get_blob_client(
    container=output_container_name, blob=output_blob_name
)
with open(results_filename, 'rb') as data:
    output_blob_client.upload_blob(data, overwrite=True)
print(
    "Sentiment analysis results have been uploaded to "
    "the output container."
)
```

After obtaining the sentiment scores, the script saves these results to a JSON file and uploads this file back to an output container in Azure Blob Storage.

This process automates the workflow of extracting feedback data, analyzing it for sentiment, and storing the analysis results, thus facilitating easy integration and further use in data processing or visualization pipelines.

Creating a SQL Database to Store the Data

Now, with the data pipeline established, you can create an Azure SQL Database to store the data:

1. Execute the script locally.
2. If it executes successfully, try running the Data Factory pipeline by clicking Debug.
3. If the pipeline runs successfully, query the SQL server and try to return the rows that were created, as follows:

```
select *
from CustomerFeedbackAnalysis
```

Visualizing the Data with Power BI

To visualize the data with Power BI, follow these steps:

1. Open Power BI Desktop.
2. Select Home → Get Data → Azure → Azure SQL Database.
3. Enter your database credentials and connect to the database.
4. Select the CustomerFeedbackAnalysis table.
5. Use the data fields to create various visualizations. For example:
   a. A line chart showing sentiment trends over time
   b. A pie chart displaying the proportion of positive, negative, and neutral feedback
6. Arrange these visualizations on a dashboard as per your preference.
7. Once your dashboard is ready, publish it to the Power BI service for sharing and further analysis.

When building dashboards in Power BI, follow these guidelines to ensure clarity and impact:

- Choose the right chart type for your data—such as bar charts for comparisons, line charts for trends, and scatter plots for relationships—so your visuals can communicate insights without confusion.
- Organize your layout in reading order, placing the most important metrics in the top-left corner and drilling into more detail as you move down and to the right.
- Limit the number of visual types on a single page and use consistent color palettes and fonts to reduce the reader's cognitive load.
- Label axes and legends clearly, include data labels only where they add value, and provide tool tips for contextual details.

- Design with your audience in mind by grouping related visuals and ensuring accessibility through high-contrast colors and meaningful alt text for screen readers.

And with that, you've gained hands-on experience in integrating multiple Azure services for storage, analysis, and visualization, while creating an end-to-end AI-powered analytics solution!

Chapter Review

In this chapter, you learned about various storage options that are provided by Azure and which ones to pick for different scenarios. You also learned about the tools you can use to perform data analysis and visualization when working with AI services. This will provide you with a strong foundation to draw on when tackling the material in the upcoming chapters.

To be successful on the exam, you'll need to have a firm grasp of the following concepts covered in this chapter:

- Which storage option to use for specific Azure AI service workloads
- Which tools to use to perform data analysis when working with AI services
- How to use data visualization tools to explore data relevant to Azure AI services

Keep in mind that we haven't covered all of the storage options you can use for different AI workloads. We'll delve into more of them as we develop solutions over the next chapters, where we will be exploring the AI workloads that are relevant to the AI-102 exam.

Now, go ahead and apply what you've learned by taking the following quiz, which is designed to evaluate your understanding of the material in this chapter.

Chapter Quiz

1. Which of the following is a key consideration when choosing storage options for AI solutions in Azure?

   A. The color scheme of the Azure portal

   B. The latency and throughput requirements

   C. The physical locations of the data centers

   D. The programming languages supported by the storage service