DP-100 – cases and exam questions

Module 1: Design a machine learning solution

Multiple choice questions

. Which of the following tasks is associated with designing a data ingestion solution for machine learning projects in Azure?

A. Implementing model training algorithms.
B. Choosing appropriate compute for machine learning.
C. Selecting features for model training.
D. **Identifying the data source and desired data format. (Correct answer)**

.

. What could be a reason to train a model with GPU compute instead of CPU compute?

A. The model is trained in a Jupyter notebook.
B. The model is trained using Azure Databricks.
C. **The model is trained using PyTorch. (Correct answer)**
D. The model is trained using PySpark.

.

. When aiming to design a model deployment solution in Azure for machine learning projects, which aspect should you primarily consider first?

A. Selecting the appropriate machine learning algorithms.
B. Converting notebooks to scripts for production.
C. **Choosing the appropriate compute resources. (Correct answer)**

.

. What is the main purpose of designing a machine learning operations (MLOps) solution?

A. Building and training machine learning models.
B. **Making your machine learning workloads robust and reproducible. (Correct answer)**
C. Deploying your machine learning models to production.
D. Evaluating your machine learning models with appropriate metrics.

After training a machine learning model to predict customer churn, you need to evaluate its performance to ensure it meets the business requirements. Which of the following metrics is most appropriate for evaluating the performance of a classification model?

A. Mean Absolute Error (MAE)
B. Root Mean Squared Error (RMSE)
C. Precision, Recall, and F1 Score
D. R-squared (R^2)

Answer: C. Precision, Recall, and F1 Score

Which of the following is a key benefit of using Azure Machine Learning environments?

A. Automatic scaling of compute resources
B. Version control for dependencies and packages

C. Built-in data visualization tools
D. Integrated development environment (IDE) for coding

Answer: B. Version control for dependencies and packages

In Azure Machine Learning, what is the primary purpose of an environment?

A. To manage user access and permissions
B. To define the compute resources for training models
C. To specify the software dependencies and configurations for experiments
D. To store and manage datasets

Answer: C. To specify the software dependencies and configurations for experiments

Open-ended questions

1. What is the benefit of storing data in a data storage solution on Azure, separated from compute?
   Possible answer: Separating compute from storage allows you to (1) scale the compute independently from the storage capacity, and (2) allow you to shut down compute to save costs while persisting your data separately.
2. What is the benefit of Spark compute?
   Possible answer: Spark allows for distributed processing which can reduce the time it needs to process data and train models.
3. What could be a reason for a team to choose Azure AI services to train a machine learning model?
   Possible answer: Using the prebuilt models provided with Azure AI services, a team can save time and effort to train complex models for tasks like computer vision and natural language processing.

Use case 1 - **Predicting Customer Churn for a Telecom Company**

Scenario:

You are a data scientist at a telecom company. The company is facing a high rate of customer churn and wants to implement a machine learning solution to predict which customers are likely to leave. This will help the company take proactive measures to retain these customers.

Objectives:

1. **Understand the Problem**: Discuss the business problem and the importance of predicting customer churn.
2. **Data Collection and Preparation**: Identify the types of data needed (e.g., customer demographics, usage patterns, customer service interactions) and discuss data cleaning and preprocessing steps.
3. **Feature Engineering**: Brainstorm potential features that could be useful for the model (e.g., average call duration, number of complaints, monthly charges).
4. **Model Selection**: Discuss different types of models that could be used (e.g., logistic regression, decision trees, random forests) and the criteria for selecting a model.
5. **Model Training and Evaluation**: Outline the steps for training the model, including splitting the data into training and test sets, and discuss evaluation metrics (e.g., accuracy, precision, recall, F1 score).
6. **Model Deployment**: Discuss how to deploy the model into a production environment and monitor its performance over time.
7. **Ethical Considerations**: Address any ethical concerns related to the use of customer data and the potential impact of the model's predictions.

Discussion Points:

- **Data Quality**: How to handle missing or inconsistent data.
- **Feature Importance**: How to determine which features are most important for predicting churn.
- **Model Interpretability**: The importance of having a model that is interpretable and explainable to stakeholders.
- **Bias and Fairness**: How to ensure the model does not unfairly target certain groups of customers.
- **Continuous Improvement**: Strategies for continuously improving the model as more data becomes available.

Tentative Answers:

1. **Understand the Problem**:
    - **Answer**: The business problem is high customer churn, which impacts revenue and customer acquisition costs. Predicting churn allows the company to take proactive measures to retain customers, such as offering promotions or improving customer service.
2. **Data Collection and Preparation**:
    - **Answer**: Data needed includes customer demographics (age, gender), usage patterns (call duration, data usage), and customer service interactions (number of complaints, service calls). Data cleaning involves handling missing values, removing duplicates, and normalizing data.

3. **Feature Engineering**:
    - **Answer**: Potential features include average call duration, number of complaints, monthly charges, contract type, tenure, and payment method. Feature engineering might also involve creating new features like the ratio of data usage to call duration.
4. **Model Selection**:
    - **Answer**: Suitable models could be logistic regression for its simplicity and interpretability, decision trees for their ability to handle non-linear relationships, and random forests for their robustness and accuracy. Model selection criteria include performance metrics, interpretability, and computational efficiency. You can find an overview here: https://learn.microsoft.com/en-us/azure/machine-learning/how-to-select-algorithms?view=azureml-api-1
5. **Model Training and Evaluation**:
    - **Answer**: If you have enough data, split the data into training (70%) and test (30%) sets. Train the model on the training set and evaluate it using metrics like accuracy, precision, recall, and F1 score. Cross-validation can be used to ensure the model's robustness.
6. **Model Deployment**:
    - **Answer**: Deploy the model using a cloud service like Azure Machine Learning. Monitor its performance using dashboards and set up alerts for any significant drops in accuracy. Regularly retrain the model with new data to maintain its performance.
7. **Ethical Considerations**:
    - **Answer**: Ensure customer data is anonymized and secure. Obtain consent for data usage. Be transparent about how the model's predictions are used and ensure it does not discriminate against any group of customers.
8. **Compute selection**
    - For designing a machine learning solution, you have several compute environment options, each with its own strengths.
    - ### 1. Azure Databricks (Spark)

        - **Use Case**: Ideal for big data processing and advanced analytics.
        - **Strengths**: Scalable, integrates well with Azure services, supports collaborative workspaces, and is optimized for Apache Spark.
        - **Example**: Use Databricks for preprocessing large datasets, feature engineering, and training machine learning models on distributed data.

    ### 2. Azure Machine Learning Compute Instances

        - **Use Case**: Suitable for development, training, and deployment of machine learning models.
        - **Strengths**: Easy to set up, integrates with Azure ML services, supports Jupyter notebooks, and provides GPU options for deep learning.
        - **Example**: Use compute instances for interactive development and experimentation with machine learning models.

    ### 4. Azure Kubernetes Service (AKS)

        - **Use Case**: Ideal for deploying and managing containerized applications.
        - **Strengths**: Scalable, supports microservices architecture, integrates with CI/CD pipelines, and provides robust orchestration.
        - **Example**: Use AKS to deploy machine learning models as microservices, ensuring scalability and easy management.

5. Azure Fabric

- **Use Case**: Suitable for microservices and container orchestration.
- **Strengths**: High availability, scalability, supports stateful and stateless services, and integrates with Azure services.
- **Example**: Use Azure Fabric to deploy and manage machine learning models in a microservices architecture.

6.Azure Machine Learning Compute Clusters are a powerful option for scaling your machine learning workloads. Here's how they can be used effectively:

Use Case:

- **Scenario**: Training large machine learning models or running extensive hyperparameter tuning tasks that require significant computational resources.

Strengths:

- **Scalability**: Automatically scales up or down based on the workload, ensuring efficient use of resources.
- **Cost-Effective**: Only pay for the compute resources you use, with the ability to scale down to zero when not in use.
- **Integration**: Seamlessly integrates with Azure Machine Learning services, making it easy to manage and monitor experiments.
- **Flexibility**: Supports a variety of VM sizes, including GPU-enabled VMs for deep learning tasks.

Example Workflow:

1. **Setup**: Create a compute cluster in the Azure Machine Learning workspace. Choose the appropriate VM size based on your workload requirements.
2. **Data Preparation**: Use the cluster to preprocess and clean your data, leveraging its scalability for large datasets.
3. **Model Training**: Train your machine learning models on the cluster. Utilize distributed training for large models to speed up the process.
4. **Hyperparameter Tuning**: Run hyperparameter tuning experiments in parallel, taking advantage of the cluster's ability to handle multiple jobs simultaneously.
5. **Model Evaluation**: Evaluate model performance using the cluster, ensuring that you have the computational power to handle large evaluation datasets.
6. **Deployment**: Once the model is trained and evaluated, you can deploy a batch deployment to a cluster.

Benefits:

- **Efficiency**: Reduces the time required for training and experimentation by leveraging multiple nodes.
- **Resource Management**: Automatically manages resources, scaling up during peak demand and scaling down during idle times.
- **Ease of Use**: Integrated with Azure Machine Learning, providing a unified interface for managing experiments, data, and compute resources.

9. **Managed Online Endpoints** in an Azure Machine Learning (Azure ML) workspace offers a streamlined way to deploy machine learning models for real-time inference.
    a. Pros of Using Managed Endpoints
        i. Fully Managed Infrastructure
            1. Azure handles provisioning, scaling, patching, and OS updates.

2. No need to manage Kubernetes clusters or container instances.

ii. Real-Time Inference
    1. Designed for low-latency, synchronous predictions.
    2. Ideal for applications like fraud detection, personalization, and recommendation systems.

iii. Auto-Scaling
    1. Automatically scales based on traffic load.
    2. Supports both CPU and GPU SKUs.

iv. Enterprise-Grade Security
    1. Supports private endpoints, managed identities, and network isolation.
    2. Integration with Azure Key Vault and Azure Private Link.

v. Monitoring & Logging
    1. Built-in integration with Azure Monitor and Log Analytics.
    2. Provides detailed metrics, logs, and cost breakdowns per deployment.

vi. MLOps Features
    1. Supports A/B testing, traffic splitting, and champion-challenger models.
    2. Enables safe rollouts and rollback strategies.

vii. Developer-Friendly
    1. Supports deployment via CLI, Python SDK, YAML, and Azure ML Studio.
    2. Compatible with MLflow and Triton for no-code deployment.

b. Cons of Using Managed Endpoints

i. Cost
    1. Always-on compute can be expensive compared to batch endpoints.
    2. Additional charges for networking, storage, and monitoring services.

ii. Limited Customization
    1. Less control over the underlying infrastructure compared to Kubernetes endpoints.
    2. Not suitable for highly customized deployment environments.

iii. Network Constraints
    1. Requires careful configuration for private networking and outbound rules.
    2. Public access must be explicitly enabled or restricted by IP.

iv. Quotas and Limits
    1. Max 50 managed endpoints per subscription.
    2. Max 20 deployments per endpoint.

v. Cold Start Latency
    1. Initial deployment or scaling may introduce slight delays before the endpoint becomes responsive.

c. When to Use Managed Endpoints

Use them when:

- You want **quick deployment** without managing infrastructure.
- Your application requires **real-time predictions**.
- You need **enterprise-grade security and monitoring**.
- You're building **production-grade ML services** with MLOps practices.

Avoid them if:

- You need **fine-grained infrastructure control**.
- Your use case is **batch-oriented** or **cost-sensitive**.
- You already have a **Kubernetes setup** and prefer full control.

Part 2: Explore and configure the Azure Machine Learning workspace

Multiple choice questions

Which developer tool should be used when interacting with the Azure Machine Learning workspace with the purpose of automation?

A. The Azure portal.
B. The Azure Machine Learning Studio.
C. The Azure Machine Learning Python SDK.
D. **The Azure Machine Learning extension for the Azure CLI. (Correct answer)**

Which URI should be used to connect to data stored in an Azure Data Lake (Gen2)?
A. http(s)
B. **abfs(s) (Correct answer)**
C. azureml
D. blob

Which compute can be used for experimentation in notebooks?
A. **Serverless Spark (Correct answer)**
B. Compute cluster
C. Containers
D. Kubernetes clusters

Which of the following best describes an MLTable in Azure Machine Learning?

A. A single file containing raw data
B. A directory containing multiple files
C. A YAML-based file that defines how data files should be loaded and transformed
D. A Python script used for data preprocessing

Answer: C

What is the primary advantage of using an MLTable over directly referencing files or folders in Azure Machine Learning?

A. MLTable allows for the execution of Python scripts
B. MLTable provides a detailed blueprint for data loading, including transformations and column type definitions
C. MLTable automatically trains machine learning models
D. MLTable is used to store model artifacts

Answer: B

Which of the following is a valid type of data asset in Azure Machine Learning SDK v2?

A. `uri_file`
B. `uri_folder`
C. `mltable`
D. **All of the above**

Answer: D

How do you create a data asset from a local folder using Azure ML SDK v2?

A. `DataAsset.create(path='path/to/folder')`
B. `DataAsset.from_path(path='path/to/folder')`
C. `Data(path='path/to/folder', type=AssetTypes.URI_FOLDER)`
D. `DataAsset.load(path='path/to/folder')`

Answer: C

Open-ended questions

1. Explain one benefit of using data assets in the Azure Machine Learning workspace.
    a. **Possible answers: Data assets allow you to (1) share and reuse data easily with other team members, (2) seamlessly access data during model training without worrying about connection strings or data paths, and (3) version the metadata of the data asset.**
2. Explain one approach to save compute costs.
    a. **Possible answers: Compute costs can be minimized by efficiently using compute by (1) choosing the appropriate size, and (2) by minimizing compute time, for example by using compute cluster that scale down when inactive, or scheduling the compute instance to stop at specified times. Using different modules in pipeline with different compute configurations**
3. Explain the difference between curated and custom environments in Azure Machine Learning.
    a. **Possible answer: A curated environment is predefined and immediately available in the workspace on creation. Curated environments are easy to use for common purposes. When you need to define your own environment (with specific packages), you can create a custom environment to use and reuse across workloads.**
4. What is a benefit of using components?
    a. Possible answer: Components allow you to create reusable scripts that can easily be shared across users within the same Azure Machine Learning workspace.

Module 2. Explore data, and run experiments

Part : Experiment with Azure ML

Multiple choice questions

What is the primary purpose of using Automated Machine Learning? Automate model deployment.

A. Automate data exploration.

B. **Automate model selection and tuning. (Correct answer)**
C. Automate data visualization.

. When using Automated Machine Learning to find the best classification model, what does the system optimize?
A. Training time
B. Model complexity
C. **Model accuracy metrics (Correct answer)**
D. Number of features

. How does MLflow contribute to tracking model training in notebooks? MLflow optimizes model hyperparameters.
A. MLflow facilitates data preprocessing.
B. **MLflow logs and monitors model metrics. (Correct answer)**
C. MLflow automates model deployment.

Which of the following methods is commonly used for hyperparameter tuning in machine learning models?

A) Gradient Descent
B) Grid Search
C) Principal Component Analysis (PCA)
D) K-Means Clustering

Answer: B) Grid Search

Which of the following techniques can be used to reduce the computational cost of hyperparameter tuning?

A) Random Search
B) Grid Search
C) Gradient Descent
D) Principal Component Analysis (PCA)

Answer: A) **Random Search**

In the context of hyperparameter tuning, what is the purpose of using cross-validation?

A) To increase the size of the training dataset
B) To ensure that the model is not overfitting to the training data
C) To reduce the number of hyperparameters
D) To convert categorical data into numerical data

Answer: B) To ensure that the model is not overfitting to the training data

Open-ended questions

1. What is a potential benefit of using Automated Machine Learning?
   Possible answer: Instead of manually having to test and evaluate various configurations to train a machine learning model, you can automate it and train multiple models in parallel. The "best" model can more quickly be found.

2. What is MLflow and why would it be used?
   Possible answer: MLflow is an open-source library for tracking and managing machine learning experiments and models. MLflow Tracking allows you to log parameters, metrics, and artifacts.
3. What are the different tasks that can be chosen in Automated Machine Learning
   Possible answer: Automated Machine Learning can be used for classification, regression, time-series forecasting, computer vision, and natural language processing.

Use case 2 – Automl

Customer Use Case: Customer Churn Prediction

Scenario: A telecommunications company wants to predict customer churn to proactively retain customers. They have historical data on customer behavior, including usage patterns, service complaints, and demographic information. The company decides to use Azure Machine Learning's AutoML capabilities to build and deploy a churn prediction model.

Tasks:

1. **Data Preparation:**
   - **Question:** What steps would you take to prepare the data for AutoML?
   - **Answer:**
     - Clean the data by handling missing values and outliers.
     - Feature engineering to create relevant features from raw data.
     - Split the data into training and validation sets.
     - Ensure the target variable (churn) is correctly labeled.
2. **Configuring AutoML:**
   - **Question:** How would you configure an AutoML experiment in Azure Machine Learning?
   - **Answer:**
     - Define the experiment settings, including the task type (classification), primary metric (e.g., AUC), and compute target.
     - Specify the training data, an ML table and target column.
     - Set up data preprocessing steps (e.g., normalization, encoding).
     - Configure the experiment to run multiple iterations with different algorithms and hyperparameters.
3. **Running AutoML Experiment:**
   - **Question:** What steps are involved in running the AutoML experiment and selecting the best model?
   - **Answer:**
     - Submit the AutoML experiment and monitor its progress.
     - Review the experiment results to compare model performance metrics.
     - Select the best-performing model based on the primary metric.
     - Analyze the model's feature importance and performance on the validation set.
4. **Model Deployment:**
   - **Question:** How can you deploy the best model to production using Azure Machine Learning?
   - **Answer:**
     - Register the best model in the Azure Machine Learning model registry.
     - Create an inference pipeline for the model.

        - Deploy the model as a web service
        - Test the deployed model with sample data to ensure it works correctly.
5. **Model Monitoring and Management:**
    - **Question:** What strategies would you use to monitor and manage the deployed model?
    - **Answer:**
        - Set up monitoring to track model performance metrics and usage.
        - Implement logging to capture predictions and errors.
        - Use Azure Machine Learning's model management features to update and retrain the model as needed.
        - Establish a feedback loop to incorporate new data and improve the model over time.

More on Featurization is a crucial step in the machine learning pipeline, especially when using AutoML in Azure Machine Learning. It involves transforming raw data into features that can be used to train machine learning models. Here's a more detailed look at the featurization process in AutoML for Azure ML:

Steps in Featurization

1. **Data Cleaning:**
    - **Handling Missing Values:** AutoML can automatically handle missing values by either imputing them with mean, median, or mode, or by using more sophisticated techniques like K-nearest neighbors (KNN) imputation.
    - **Removing Duplicates:** Duplicate records are identified and removed to ensure data quality.
2. **Data Transformation:**
    - **Normalization and Scaling:** Continuous features are normalized or scaled to ensure they have a similar range, which helps improve the performance of many machine learning algorithms.
    - **Encoding Categorical Variables:** Categorical features are converted into numerical values using techniques like one-hot encoding, label encoding, or target encoding.
3. **Feature Engineering:**
    - **Creating New Features:** AutoML can automatically create new features from existing ones. For example, it can generate interaction features (e.g., multiplying two features together) or polynomial features (e.g., squaring a feature).
    - **Date and Time Features:** Date and time features can be decomposed into more granular components like year, month, day, hour, etc., to capture temporal patterns.
4. **Text Featurization:**
    - **Text Vectorization:** Text data is converted into numerical features using techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings (e.g., Word2Vec, GloVe).
    - **N-grams:** AutoML can generate n-grams (combinations of n consecutive words) to capture more context from text data.
5. **Feature Selection:**
    - **Removing Low-Variance Features:** Features with little to no variance are removed as they do not contribute much to the model's predictive power.
    - **Correlation Analysis:** Highly correlated features are identified, and redundant features are removed to reduce multicollinearity.
    - **Feature Importance:** AutoML can use techniques like feature importance scores from tree-based models to select the most relevant features.

In Azure Machine Learning, AutoML handles featurization automatically, but you can also customize the process to some extent. Here are some key aspects:

- **Automatic Featurization:** By default, AutoML in Azure ML performs automatic featurization, which includes all the steps mentioned above. This is particularly useful for beginners or when you want to quickly prototype models.
- **Custom Featurization:** You can customize the featurization process by specifying certain transformations or by providing your own preprocessed data. This is useful when you have domain-specific knowledge that can improve the feature engineering process.
- **Featurization Settings:** Azure ML allows you to configure featurization settings through the `FeaturizationConfig` class. You can enable or disable specific transformations, set parameters for imputation, and more.

Module 3: Train and deploy models

Multiple choice questions

When tracking a machine learning model with MLflow, what method should be used to log the value of the regularization parameter?

A. mlflow.log_param() (Correct answer)
B. mlflow.log_metric()
C. mlflow.log_artifact()

When executing a sequence of multiple scripts in Azure Machine Learning, what type of job is run?

A. Single
B. Command
C. Sweep
D. Pipeline (Correct answer)

What is an important aspect of ensuring code is production-ready?

a. Refactoring code to functions. (Correct answer)
b. Adding comments to code for readability.
c. Converting scripts to notebooks.

What feature in the Responsible AI dashboard should be used to determine whether your findings related to fairness, error analysis, and causality are a result of your dataset's distribution?

A. Error analysis
B. Feature importance
C. Data analysis (Correct answer)

Where in the MLmodel file will you find which framework was used to train the model?

A. Artifact path
B. Flavor (Correct answer)
C. Signature

What is shown in the fairness assessment when different cohorts of the data perform differently when comparing selected performance metrics?

A. Disparity in model performance (Correct answer)
B. Disparity in selection rate
C. Disparity in data cohorts

When deploying a model for real-time predictions, what is the easiest way to accomplish this?

A. Deploy an MLflow model to a batch endpoint.
B. Deploy a custom model to a batch endpoint.
C. Deploy an MLflow model to a managed online endpoint. (Correct answer)

How can a managed online endpoint quickly be tested?

A. Through the Azure portal.
B. Through the Azure Machine Learning Studio. (Correct answer)
C. Through a notebook in the Azure Machine Learning workspace.

Open-ended questions

A. **What is the difference between a managed online endpoint and a batch endpoint**?
   a. Possible answer: A managed online endpoint is an HTTPS endpoint that returns real-time predictions for individual data points, with Azure Machine Learning managing the underlying infrastructure. A batch endpoint is an HTTPS endpoint that triggers a batch scoring job, allowing you to get batch predictions by using a compute cluster.

B. How do you troubleshoot a batch scoring job?
   a. Possible answer: To troubleshoot a batch scoring job, you can review its details, outputs, and logs. The batch scoring job runs as a pipeline job, so you can troubleshoot it by reviewing the details and outputs of the pipeline job itself.

C. What is the difference between logging a model as an artifact and logging it as a model?
   a. Possible answer: When you log a model with MLflow, you can log it as an artifact or as a model. When you log a model as an artifact, the model is treated as a file. When you log a model as a model, you're adding information to the registered model that enables you to use the model directly in pipelines or deployments.

D. What is the purpose of the MLmodel file?
   a. Possible answer: The purpose of the MLmodel file is to contain the model's metadata, which allows for model traceability. The MLmodel file is also used when deploying the model.

E. What is the difference between aggregate and individual feature importance?
   a. Possible answer: Aggregate feature importance indicates the overall feature importance for all test data. It shows the relative influence of each feature on the predicted label. On the other hand, individual feature importance shows the feature importance for an individual prediction. In classification, this shows the relative support for each possible class per feature.

Use case 3 – MLFlow

Customer Use Case: Predictive Maintenance for Manufacturing Equipment

Scenario: A manufacturing company wants to implement a predictive maintenance system to reduce downtime and maintenance costs. They have historical data on equipment failures, maintenance logs, and sensor readings from various machines. The company wants to build a machine learning model to predict equipment failures before they occur. They have chosen to use MLflow to manage the machine learning lifecycle.

Tasks:

1. **Model Training:**
   - **Question:** How would you use MLflow to track experiments and model training?
   - **Answer:**
     - Use MLflow to log parameters, metrics, and artifacts.
     - Track different model versions and their performance.
     - Use MLflow's experiment tracking to compare different models and select the best one.
2. **Model Deployment:**

- **Question:** How can MLflow help in deploying the best model to production?
- **Answer:**
  - Use MLflow's model registry to manage model versions.
  - Deploy the model using MLflow's deployment tools (e.g., MLflow Models).
  - Monitor the deployed model's performance and update it as needed.

3. **Model Monitoring:**
   - **Question:** What strategies would you use to monitor the model's performance in production?
   - **Answer:**
     - Set up automated monitoring to track model performance metrics.
     - Use MLflow to log predictions and compare them with actual outcomes.
     - Implement alerting mechanisms for model drift or performance degradation.
4. **Collaboration:**
   - **Question:** How can MLflow facilitate collaboration among data scientists and engineers in this project?
   - **Answer:**
     - Use MLflow's centralized tracking server to share experiment results.
     - Collaborate on model development and deployment using MLflow's version control.
     - Share insights and findings through MLflow's UI and reports.

Use case 4 – Best practice for deploying models with Azure ML

A customer asks you about deploying models in Azure ML. Discuss why to use cloud solutions like Azure ML for deployment.

Deploying machine learning models effectively is crucial for ensuring they perform well in production environments. Here are some best practices for deploying models, particularly using Azure Machine Learning:

Best Practices for Deploying Models

1. **Model Versioning and Registration:**
   - Always register your models in a central repository like Azure ML's model registry. This ensures version control and easy access for deployment and monitoring.
2. **Environment Consistency:**
   - Use containerization (e.g., Docker) or virtualization to ensure that the environment in which the model was trained is consistent with the deployment environment. This helps avoid discrepancies due to different software versions or configurations.
3. **Automated Deployment Pipelines:**
   - Implement CI/CD pipelines for model deployment. Tools like Azure DevOps or Github Actions can automate the process of testing, validating, and deploying models, reducing manual errors and speeding up the deployment process.
4. **Scalability:**
   - Design your deployment to scale based on demand. Azure ML's managed endpoints can automatically scale up or down based on the traffic, ensuring optimal performance and cost-efficiency.
5. **Monitoring and Logging:**
   - Continuously monitor the deployed model for performance metrics such as latency, throughput, and accuracy.
6. **Security:**
   - Implement robust security measures to protect your model and data. This includes securing endpoints, using authentication and authorization mechanisms, and encrypting data in transit and at rest.
7. **A/B Testing and Rollbacks:**
   - Use A/B testing to compare different versions of the model in production. This helps in understanding which version performs better. Also, have a rollback strategy in place to revert to a previous version if the new deployment causes issues. Safe rollout for online endpoints - Azure Machine Learning | Microsoft Learn
8. **Batch vs. Real-time Inference:**
   - Choose the appropriate inference method based on your use case.
9. **Documentation and Collaboration:**
   - Maintain thorough documentation of the deployment process, model specifications, and any dependencies. This facilitates collaboration among team members and ensures that the deployment process is reproducible.
10. **Regular Updates and Retraining:**
    - Regularly update and retrain your models to incorporate new data and improve performance. Automate this process as much as possible to keep the models up-to-date without manual intervention.

By following these best practices, you can ensure that your machine learning models are deployed efficiently, securely, and with high reliability.

Optimize language models for AI applications

Prepare for model optimization

Question 1:What is the primary goal of fine-tuning a pre-trained language model?

- A) To train the model from scratch
- B) To adapt the model to a specific task or domain
- C) To reduce the size of the model
- D) To increase the number of parameters in the model

Answer: B) To adapt the model to a specific task or domain

Question 2: Which technique is commonly used to prevent overfitting when fine-tuning language models?

- A) Increasing the learning rate
- B) Using dropout regularization
- C) Reducing the size of the training dataset
- D) Adding more layers to the model

Answer: B) Using dropout regularization

Question 3: What is the purpose of using a validation dataset during model optimization?

- A) To train the model
- B) To test the model's performance after training
- C) To guide hyperparameters and evaluate performance
- D) To increase the size of the training dataset

Answer: C) To guide hyperparameters and evaluate performance

Question 4: What is the role of learning rate scheduling in optimizing language models?

- A) To keep the learning rate constant throughout training
- B) To increase the learning rate as training progresses
- C) To adjust the learning rate dynamically to improve convergence

- D) To reduce the size of the model

Answer: C) To adjust the learning rate dynamically to improve convergence

Question 5: What is the primary purpose of testing a deployed model in the playground?

- A) To retrain the model with new data
- B) To evaluate the model's performance on real-world inputs
- C) To increase the model's accuracy
- D) To optimize the model's architecture

Answer: B) To evaluate the model's performance on real-world inputs

Optimize through prompt engineering and prompt flow

Question 1: What is the primary goal of prompt engineering in optimizing language models?

- A) To modify the model's architecture
- B) To design effective prompts that guide the model's responses
- C) To increase the size of the training dataset
- D) To reduce the number of parameters in the model

Answer: B) To design effective prompts that guide the model's responses

Question 2: Which technique can improve the effectiveness of prompts in language models?

- A) Using ambiguous phrasing
- B) Providing clear instructions and context
- C) Reducing the length of the prompt
- D) Avoiding examples in the prompt

Answer: B) Providing clear instructions and context

Question 3: What is the purpose of prompt flow in optimizing language models?

- A) To test multiple prompts and select the best-performing one
- B) To reduce the computational cost of the model
- C) To increase the model's training speed
- D) To simplify the model's architecture

Answer: A) To test multiple prompts and select the best-performing one

Question 4: Which feature of prompt flow allows users to evaluate the model's responses systematically?

- A) Randomized testing
- B) Automated scoring and comparison
- C) Manual evaluation
- D) Model retraining

Answer: B) Automated scoring and comparison

Question 5: How can examples in a prompt improve the model's output?

- A) By reducing the model's accuracy
- B) By providing a clear pattern for the model to follow
- C) By increasing the complexity of the prompt
- D) By limiting the model's ability to generalize

Answer: B) By providing a clear pattern for the model to follow

Question 6: What is the primary purpose of defining chaining logic in the Prompt Flow SDK?

- A) To train the model on new data
- B) To create workflows that combine multiple prompts and model responses
- C) To reduce the size of the model
- D) To increase the number of model parameters

Answer: B) To create workflows that combine multiple prompts and model responses

Question 7: Which feature in the Prompt Flow SDK allows chaining logic to handle conditional workflows?

- A) Static prompts
- B) Decision nodes
- C) Model retraining
- D) Batch processing

Answer: B) Decision nodes

Question 8: How does chaining logic improve the performance of language models in complex tasks?

- A) By simplifying the model architecture
- B) By enabling sequential processing of prompts and responses
- C) By reducing the computational cost of the model
- D) By increasing the size of the training dataset

Answer: B) By enabling sequential processing of prompts and responses

Question 9: What is the role of input validation in chaining logic workflows?

- A) To ensure the model generates diverse outputs
- B) To verify the correctness of inputs before processing
- C) To increase the model's training speed
- D) To reduce the number of prompts in the workflow

Answer: B) To verify the correctness of inputs before processing

Optimize through Retrieval Augmented Generation (RAG)

Question 1: What is the main benefit of using Retrieval-Augmented Generation (RAG) in a data science solution?

- A) It replaces the need for embedding models
- B) It augments model responses with up-to-date external knowledge
- C) It reduces latency by eliminating retrieval steps
- D) It guarantees 100% factual accuracy

Answer: B) It augments model responses with up-to-date external knowledge

Question 2: In a typical RAG pipeline on Azure, which service is used to store and index vectors for fast similarity search?

- A) Azure Blob Storage
- B) Azure SQL Database
- C) Azure AI Search
- D) Azure Data Lake

Answer: C) Azure AI Search

Question 3: During model evaluation, you notice hallucinations in RAG outputs. Which approach is most effective to reduce them?

- A) Removing the retrieval step entirely
- B) Expanding the retrieval corpus indiscriminately
- C) Adding a filtering layer for relevance
- D) Training the model exclusively on generative tasks

Answer: C) Adding a filtering layer for relevance

Question 4: Which metric combination provides the best insight into the effectiveness of your RAG implementation?

- A) Retrieval recall and end-to-end answer accuracy
- B) Model parameter count and embedding vector size
- C) Latency and storage cost only
- D) Number of documents retrieved and CPU utilization

Answer: A) Retrieval recall and end-to-end answer accuracy

Optimize through fine-tuning

Question 1: What is the primary goal of fine-tuning a pre-trained model in a machine learning solution?

- A) Train a model from scratch on a large dataset
- B) Adapt the model to a specific domain or task with minimal data
- C) Compress model parameters for deployment
- D) Generate synthetic data for training

Answer: B) Adapt the model to a specific domain or task with minimal data

Question 2: Which hyperparameter is most critical to tune in order to avoid unstable updates and ensure convergence during fine-tuning?

- A) Batch size
- B) Learning rate
- C) Number of epochs
- D) Embedding dimension

Answer: B) Learning rate

Question 4: During fine-tuning, which dataset split is used to validate the model and tune hyperparameters while avoiding overfitting?

- A) Training set
- B) Validation set
- C) Test set
- D) Unseen set

Answer: B) Validation set

Use Case: Legal Contract Review Assistant

Instructions: Read the use case and answer the questions below. Reflect on how to balance precision, latency, and cost when using only existing models in Azure AI Foundry. Use the following discussion point for discussion:

- *Design & Implementation*
- *Retrieval-Augmented Prompt Flow*
- *Prompt Engineering*
- *Performance Optimization*
- *Cost Management*
- *Monitoring & Evaluation*
- *Trade-offs between smaller vs. larger models in Foundry*

The use case

A corporate legal team needs an AI assistant/Agent that:

- Ingests large volumes of contracts (PDFs, Word docs)
- Extracts and highlights key clauses (e.g., termination, indemnification)
- Answers ad-hoc, contract-specific questions ("What's the notice period?")
- Flags high-risk language based on custom policy rules

Some tentative answers

Design & Implementation Suggestions

- Data Ingestion & Indexing
  - Store raw contracts in Azure Blob Storage.
  - Use Azure AI Search to extract text, split into 300–500 token chunks, and index with semantic vectors.

Retrieval-Augmented Prompt Flow

  - In Foundry, chain a "Retriever" block (top K=5) with a "Prompt" block.
  - Craft a system prompt that instructs the model to:

1. Read retrieved chunks
2. Extract the requested clause
3. Highlight any risk terms against a company policy

- Add a “Filter” block to drop low-relevance documents (score < 0.3).

-Prompt Engineering

- Provide 2–3 few-shot examples of question/answer pairs using company-style language.
- Use placeholders for variables (e.g., `<ClauseType>`, `<RiskTerm>`).
- Tune temperature (0.0–0.2) and max tokens (150–200) for concise, deterministic outputs.

Performance Optimization

- Batch user queries (up to 16) to leverage vector-search batching.
- Cache retrieval results for identical/similar queries to avoid repeated search.
- Monitor end-to-end latency; adjust Azure AI Search replica count or Foundry concurrency limits.

Cost Management

- Choose a base LLM tier in Foundry that meets latency SLAs without over-provisioning.
- Implement query rate-limiting and idle-endpoint shutdown.

Monitoring & Evaluation

- Log each query’s retrieval score, token usage, and response time in Application Insights.
- Periodically sample Q&A pairs to compute precision/recall on clause extraction.
- Set alerts for drift: e.g., if average retrieval score drops by > 10%.