Free Questions for Databricks-Machine-Learning-Associate by braindumpscollection

Shared by Cotton on 22-07-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A data scientist uses 3-fold cross-validation and the following hyperparameter grid when optimizing model hyperparameters via grid search for a classification problem:

Hyperparameter 1: [2, 5, 10]

Hyperparameter 2: [50, 100]

Which of the following represents the number of machine learning models that can be trained in parallel during this process?

Options:

A- 3

B- 5

C- 6

D- 18

Answer:

D

Explanation:

To determine the number of machine learning models that can be trained in parallel, we need to calculate the total number of combinations of hyperparameters. The given hyperparameter grid includes:

Hyperparameter 1: [2, 5, 10] (3 values)

Hyperparameter 2: [50, 100] (2 values)

The total number of combinations is the product of the number of values for each hyperparameter: 3(valuesofHyperparameter1)2(valuesofHyperparameter2)=63(valuesofHyperparameter1)2(valuesofHyperparameter2)=6

With 3-fold cross-validation, each combination of hyperparameters will be evaluated 3 times. Thus, the total number of models trained will be: 6(combinations)3(folds)=186(combinations)3(folds)=18

However, the number of models that can be trained in parallel is equal to the number of hyperparameter combinations, not the total number of models considering cross-validation. Therefore, 6 models can be trained in parallel.

Databricks documentation on hyperparameter tuning: Hyperparameter Tuning

Question 2

Question Type: MultipleChoice

A data scientist wants to efficiently tune the hyperparameters of a scikit-learn model in parallel. They elect to use the Hyperopt library to facilitate this process.

Which of the following Hyperopt tools provides the ability to optimize hyperparameters in parallel?

Options:

A- fmin

B- SparkTrials

C- quniform

D- search_space

E- objective_function

Answer:

B

Explanation:

The SparkTrials class in the Hyperopt library allows for parallel hyperparameter optimization on a Spark cluster. This enables efficient tuning of hyperparameters by distributing the optimization process across multiple nodes in a cluster.

from hyperopt import fmin, tpe, hp, SparkTrials search_space = { 'x': hp.uniform('x', 0, 1), 'y': hp.uniform('y', 0, 1) } def objective(params): return params['x'] ** 2 + params['y'] ** 2 spark_trials = SparkTrials(parallelism=4) best = fmin(fn=objective, space=search_space, algo=tpe.suggest, max_evals=100, trials=spark_trials)

Hyperopt Documentation

Question 3

Question Type: MultipleChoice

A data scientist is wanting to explore the Spark DataFrame spark_df. The data scientist wants visual histograms displaying the distribution of numeric features to be included in the exploration.

Which of the following lines of code can the data scientist run to accomplish the task?

Options:

A- spark_df.describe()

B- dbutils.data(spark_df).summarize()

C- This task cannot be accomplished in a single line of code.

D- spark_df.summary()

E- dbutils.data.summarize (spark_df)

Answer:

E

Explanation:

To display visual histograms and summaries of the numeric features in a Spark DataFrame, the Databricks utility function dbutils.data.summarize can be used. This function provides a comprehensive summary, including visual histograms.

Correct code:

dbutils.data.summarize(spark_df)

Other options like spark_df.describe() and spark_df.summary() provide textual statistical summaries but do not include visual histograms.

Databricks Utilities Documentation

Question 4

Question Type: **MultipleChoice**

A data scientist wants to use Spark ML to one-hot encode the categorical features in their PySpark DataFrame features_df. A list of the names of the string columns is assigned to the input_columns variable.

They have developed this code block to accomplish this task:

```
ohe = OneHotEncoder(
    inputCols=input_columns,
    outputCols=output_columns
)
ohe_model = ohe.fit(features_df)
ohe_features_df = ohe_model.transform(features_df)
```

The code block is returning an error.

Which of the following adjustments does the data scientist need to make to accomplish this task?

Options:

A- They need to specify the method parameter to the OneHotEncoder.

B- They need to remove the line with the fit operation.

C- They need to use StringIndexer prior to one-hot encodinq the features.

D- They need to use VectorAssembler prior to one-hot encoding the features.

Answer:

C

Explanation:

The OneHotEncoder in Spark ML requires numerical indices as inputs rather than string labels. Therefore, you need to first convert the string columns to numerical indices using StringIndexer. After that, you can apply OneHotEncoder to these indices.

Corrected code:

from pyspark.ml.feature import StringIndexer, OneHotEncoder # Convert string column to index indexers = [StringIndexer(inputCol=col, outputCol=col+'_index') for col in input_columns] indexer_model = Pipeline(stages=indexers).fit(features_df) indexed_features_df = indexer_model.transform(features_df) # One-hot encode the indexed columns ohe = OneHotEncoder(inputCols=[col+'_index' for col in input_columns], outputCols=output_columns) ohe_model = ohe.fit(indexed_features_df) ohe_features_df = ohe_model.transform(indexed_features_df)

PySpark ML Documentation

Question 5

Question Type: MultipleChoice

A data scientist wants to use Spark ML to impute missing values in their PySpark DataFrame features_df. They want to replace missing values in all numeric columns in features_df with each respective numeric column's median value.

They have developed the following code block to accomplish this task:

```
imputer = Imputer(
    strategy="median",
    inputCols=input_columns,
    outputCols=output_columns
)
imputed_features_df = imputer.transform(features_df)
```

The code block is not accomplishing the task.

Which reasons describes why the code block is not accomplishing the imputation task?

Options:

A- It does not impute both the training and test data sets.

B- The inputCols and outputCols need to be exactly the same.

C- The fit method needs to be called instead of transform.

D- It does not fit the imputer on the data to create an ImputerModel.

Answer:

D

Explanation:

In the provided code block, the Imputer object is created but not fitted on the data to generate an ImputerModel. The transform method is being called directly on the Imputer object, which does not yet contain the fitted median values needed for imputation. The correct approach is to fit the imputer on the dataset first.

Corrected code:

imputer = Imputer(strategy='median', inputCols=input_columns, outputCols=output_columns) imputer_model = imputer.fit(features_df) # Fit the imputer to the data imputed_features_df = imputer_model.transform(features_df) # Transform the data using the fitted imputer

PySpark ML Documentation

Question 6

Question Type: **MultipleChoice**

Which of the following evaluation metrics is not suitable to evaluate runs in AutoML experiments for regression problems?

Options:

A- F1

B- R-squared

C- MAE

D- MSE

Answer:

A

Explanation:

The code block provided by the machine learning engineer will perform the desired inference when the Feature Store feature set was logged with the model at model_uri. This ensures that all necessary feature transformations and metadata are available for the model to make predictions. The Feature Store in Databricks allows for seamless integration of features and models, ensuring that the required features are correctly used during inference.

Databricks documentation on Feature Store: Feature Store in Databricks

Question 7

Question Type: MultipleChoice

A machine learning engineer is trying to perform batch model inference. They want to get predictions using the linear regression model saved at the path model_uri for the DataFrame batch_df.

batch_df has the following schema:

customer_id STRING

The machine learning engineer runs the following code block to perform inference on batch_df using the linear regression model at model_uri:

```
predictions = fs.score_batch(
    model_uri,
    batch_df
)
```

In which situation will the machine learning engineer's code block perform the desired inference?

Options:

A- When the Feature Store feature set was logged with the model at model_uri

B- When all of the features used by the model at model_uri are in a Spark DataFrame in the PySpark

C- When the model at model_uri only uses customer_id as a feature

D- This code block will not perform the desired inference in any situation.

E- When all of the features used by the model at model_uri are in a single Feature Store table

Answer:

A

Explanation:

The code block provided by the machine learning engineer will perform the desired inference when the Feature Store feature set was logged with the model at model_uri. This ensures that all necessary feature transformations and metadata are available for the model to make predictions. The Feature Store in Databricks allows for seamless integration of features and models, ensuring that the required features are correctly used during inference.

Databricks documentation on Feature Store: Feature Store in Databricks

Question 8

Question Type: MultipleChoice

A data scientist is using Spark SQL to import their data into a machine learning pipeline. Once the data is imported, the data scientist performs machine learning tasks using Spark ML.

Which of the following compute tools is best suited for this use case?

Options:

A- Single Node cluster

B- Standard cluster

C- SQL Warehouse

D- None of these compute tools support this task

Answer:

B

Explanation:

For a data scientist using Spark SQL to import data and then performing machine learning tasks using Spark ML, the best-suited compute tool is a Standard cluster. A Standard cluster in Databricks provides the necessary resources and scalability to handle large

datasets and perform distributed computing tasks efficiently, making it ideal for running Spark SQL and Spark ML operations.

Databricks documentation on clusters: Clusters in Databricks

Question 9

Question Type: MultipleChoice

A machine learning engineering team has a Job with three successive tasks. Each task runs a single notebook. The team has been alerted that the Job has failed in its latest run.

Which of the following approaches can the team use to identify which task is the cause of the failure?

Options:

A- Run each notebook interactively

B- Review the matrix view in the Job's runs

C- Migrate the Job to a Delta Live Tables pipeline

D- Change each Task's setting to use a dedicated cluster

Answer:

B

Explanation:

To identify which task is causing the failure in the job, the team should review the matrix view in the Job's runs. The matrix view provides a clear and detailed overview of each task's status, allowing the team to quickly identify which task failed. This approach is more efficient than running each notebook interactively, as it provides immediate insights into the job's execution flow and any issues that occurred during the run.

Databricks documentation on Jobs: Jobs in Databricks

Question 10

Question Type: MultipleChoice

A new data scientist has started working on an existing machine learning project. The project is a scheduled Job that retrains every day. The project currently exists in a Repo in Databricks. The data scientist has been tasked with improving the feature engineering of the pipeline's preprocessing stage. The data scientist wants to make necessary updates to the code that can be easily adopted into the project without changing what is being run each day.

Which approach should the data scientist take to complete this task?

Options:

A- They can create a new branch in Databricks, commit their changes, and push those changes to the Git provider.

B- They can clone the notebooks in the repository into a Databricks Workspace folder and make the necessary changes.

C- They can create a new Git repository, import it into Databricks, and copy and paste the existing code from the original repository before making changes.

D- They can clone the notebooks in the repository into a new Databricks Repo and make the necessary changes.

Answer:

A

Explanation:

The best approach for the data scientist to take in this scenario is to create a new branch in Databricks, commit their changes, and push those changes to the Git provider. This approach allows the data scientist to make updates and improvements to the feature engineering part of the preprocessing pipeline without affecting the main codebase that runs daily. By creating a new branch, they can work on their changes in isolation. Once the changes are ready and tested, they can be merged back into the main branch through a pull request, ensuring a smooth integration process and allowing for code review and collaboration with other team members.

Databricks documentation on Git integration: Databricks Repos

Question 11

Question Type: MultipleChoice

A machine learning engineer has identified the best run from an MLflow Experiment. They have stored the run ID in the run_id variable and identified the logged model name as "model". They now want to register that model in the MLflow Model Registry with the name "best_model".

Which lines of code can they use to register the model associated with run_id to the MLflow Model Registry?

Options:

A- mlflow.register_model(run_id, 'best_model')

B- mlflow.register_model(f'runs:/{run_id}/model'', 'best_model'')

C- millow.register_model(f'runs:/{run_id)/model')

D- mlflow.register_model(f'runs:/{run_id}/best_model', 'model')

Answer:

B

Explanation:

To register a model that has been identified by a specific run_id in the MLflow Model Registry, the appropriate line of code is:

mlflow.register_model(f'runs:/{run_id}/model', 'best_model')

This code correctly specifies the path to the model within the run (runs:/{run_id}/model) and registers it under the name 'best_model' in the Model Registry. This allows the model to be tracked, managed, and transitioned through different stages (e.g., Staging, Production) within the MLflow ecosystem.

Reference

MLflow documentation on model registry: https://www.mlflow.org/docs/latest/model-registry.html#registering-a-model

Question 12

Question Type: **MultipleChoice**

A machine learning engineer has been notified that a new Staging version of a model registered to the MLflow Model Registry has passed all tests. As a result, the machine learning engineer wants to put this model into production by transitioning it to the Production stage in the Model Registry.

From which of the following pages in Databricks Machine Learning can the machine learning engineer accomplish this task?

Options:

A- The home page of the MLflow Model Registry

B- The experiment page in the Experiments observatory

C- The model version page in the MLflow Model Registry

D- The model page in the MLflow Model Registry

Answer:

C

Explanation:

The machine learning engineer can transition a model version to the Production stage in the Model Registry from the model version page. This page provides detailed information about a specific version of a model, including its metrics, parameters, and current stage. From here, the engineer can perform stage transitions, moving the model from Staging to Production after it has passed all necessary tests.

Reference

Databricks documentation on MLflow Model Registry: https://docs.databricks.com/applications/mlflow/model-registry.html#model-version