Databricks-Machine-Learning-Associate Practice Questions

Shared by Mccarthy on 18-08-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A team is developing guidelines on when to use various evaluation metrics for classification problems. The team needs to provide input on when to use the F1 score over accuracy.

$$2 \times \frac{precision \times recall}{(precision + recall)}$$

Which of the following suggestions should the team include in their guidelines?

Options:

A- The F1 score should be utilized over accuracy when the number of actual positive cases is identical to the number of actual negative cases.
B- The F1 score should be utilized over accuracy when there are greater than two classes in the target variable.
C- The F1 score should be utilized over accuracy when there is significant imbalance between positive and negative classes and avoiding false negatives is a priority.
D- The F1 score should be utilized over accuracy when identifying true positives and true negatives are equally important to the business problem.

Answer:

C

Explanation:

The F1 score is the harmonic mean of precision and recall and is particularly useful in situations where there is a significant imbalance between positive and negative classes. When there is a class imbalance, accuracy can be misleading because a model can achieve high accuracy by simply predicting the majority class. The F1 score, however, provides a better measure of the test's accuracy in terms of both false positives and false negatives.

Specifically, the F1 score should be used over accuracy when:

There is a significant imbalance between positive and negative classes.

Avoiding false negatives is a priority, meaning recall (the ability to detect all positive instances) is crucial.

In this scenario, the F1 score balances both precision (the ability to avoid false positives) and recall, providing a more meaningful measure of a model's performance under these conditions.

Databricks documentation on classification metrics: Classification Metrics

Question 2

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 3

Question Type: MultipleChoice

A data scientist has produced three new models for a single machine learning problem. In the past, the solution used just one model. All four models have nearly the same prediction latency, but a machine learning engineer suggests that the new solution will be less time efficient during inference.

In which situation will the machine learning engineer be correct?

Options:

A- When the new solution requires if-else logic determining which model to use to compute each prediction
B- When the new solution's models have an average latency that is larger than the size of the original model
C- When the new solution requires the use of fewer feature variables than the original model
D- When the new solution requires that each model computes a prediction for every record
E- When the new solution's models have an average size that is larger than the size of the original model

Answer:

D

Explanation:

If the new solution requires that each of the three models computes a prediction for every record, the time efficiency during inference will be reduced. This is because the inference process now involves running multiple models instead of a single model, thereby increasing the overall computation time for each record.

In scenarios where inference must be done by multiple models for each record, the latency accumulates, making the process less time efficient compared to using a single model.

Model Ensemble Techniques

Question 4

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 5

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 6

Question Type: MultipleChoice

A machine learning engineer has grown tired of needing to install the MLflow Python library on each of their clusters. They ask a senior machine learning engineer how their notebooks can load the MLflow library without installing it each time. The senior machine learning engineer suggests that they use Databricks Runtime for Machine Learning.

Which of the following approaches describes how the machine learning engineer can begin using Databricks Runtime for Machine Learning?

Options:

A- They can add a line enabling Databricks Runtime ML in their init script when creating their clusters.
B- They can check the Databricks Runtime ML box when creating their clusters.
C- They can select a Databricks Runtime ML version from the Databricks Runtime Version dropdown when creating their clusters.
D- They can set the runtime-version variable in their Spark session to ''ml''.

Answer:

C

Explanation:

The Databricks Runtime for Machine Learning includes pre-installed packages and libraries essential for machine learning and deep learning, including MLflow. To use it, the machine learning engineer can simply select an appropriate Databricks Runtime ML version from the 'Databricks Runtime Version' dropdown menu while creating their cluster. This selection ensures that all necessary machine learning libraries, including MLflow, are pre-installed and ready for use, avoiding the need to manually install them each time.

Reference

Databricks documentation on creating clusters: https://docs.databricks.com/clusters/create.html

Question 7

Question Type: MultipleChoice

A machine learning engineer wants to parallelize the inference of group-specific models using the Pandas Function API. They have developed the apply_model function that will look up and load the correct model for each group, and they want to apply it to each group of DataFrame df.

They have written the following incomplete code block:

```
prediction_df = (df
    .groupby("device_id")
    .______(apply_model, schema=apply_return_schema)
)
```

Which piece of code can be used to fill in the above blank to complete the task?

Options:

A- applyInPandas
B- groupedApplyInPandas
C- mapInPandas
D- predict

Answer:

A

Explanation:

To parallelize the inference of group-specific models using the Pandas Function API in PySpark, you can use the applyInPandas function. This function allows you to apply a Python function on each group of a DataFrame and return a DataFrame, leveraging the power of pandas UDFs (user-defined functions) for better performance.

prediction_df = (df.groupby('device_id') .applyInPandas(apply_model, schema=apply_return_schema))

In this code:

groupby('device_id'): Groups the DataFrame by the 'device_id' column.

applyInPandas(apply_model, schema=apply_return_schema): Applies the apply_model function to each group and specifies the schema of the return DataFrame.

PySpark Pandas UDFs Documentation

Question 8

Question Type: MultipleChoice

A data scientist has written a data cleaning notebook that utilizes the pandas library, but their colleague has suggested that they refactor their notebook to scale with big data.

Which of the following approaches can the data scientist take to spend the least amount of time refactoring their notebook to scale with big data?

Options:

A- They can refactor their notebook to process the data in parallel.
B- They can refactor their notebook to use the PySpark DataFrame API.
C- They can refactor their notebook to use the Scala Dataset API.
D- They can refactor their notebook to use Spark SQL.
E- They can refactor their notebook to utilize the pandas API on Spark.

Answer:

E

Explanation:

The data scientist can refactor their notebook to utilize the pandas API on Spark (now known as pandas on Spark, formerly Koalas). This allows for the least amount of changes to the existing pandas-based code while scaling to handle big data using Spark's distributed computing capabilities. pandas on Spark provides a similar API to pandas, making the transition smoother and faster compared to completely rewriting the code to use PySpark DataFrame API, Scala Dataset API, or Spark SQL. Reference:

Databricks documentation on pandas API on Spark (formerly Koalas).

Question 9

Question Type: MultipleChoice

A data scientist uses 3-fold cross-validation when optimizing model hyperparameters for a regression problem. The following root-mean-squared-error values are calculated on each of the validation folds:

* 10.0

* 12.0

* 17.0

Which of the following values represents the overall cross-validation root-mean-squared error?

Options:

A- 13.0
B- 17.0
C- 12.0
D- 39.0
E- 10.0

Answer:

A

Explanation:

To calculate the overall cross-validation root-mean-squared error (RMSE), you average the RMSE values obtained from each validation fold. Given the RMSE values of 10.0, 12.0, and 17.0 for the three folds, the overall cross-validation RMSE is calculated as the average of these three values:

OverallCVRMSE=10.0+12.0+17.03=39.03=13.0OverallCVRMSE=310.0+12.0+17.0=339.0=13.0

Thus, the correct answer is 13.0, which accurately represents the average RMSE across all folds. Reference:

Cross-validation in Regression (Understanding Cross-Validation Metrics).

Question 10

Question Type: MultipleChoice

In Which option best situations is it preferable to impute missing feature values with their

median value over the mean value?

Options:

A- When the features are of the categorical type
B- When the features are of the boolean type
C- When the features contain a lot of extreme outliers
D- When the features contain no outliers
E- When the features contain no missing no values

Answer:

C

Explanation:

Imputing missing values with the median is often preferred over the mean in scenarios where the data contains a lot of extreme outliers. The median is a more robust measure of central tendency in such cases, as it is not as heavily influenced by outliers as the mean. Using the median ensures that the imputed values are more representative of the typical data point, thus preserving the integrity of the dataset's distribution. The other options are not specifically relevant to the question of handling outliers in numerical data. Reference:

Data Imputation Techniques (Dealing with Outliers).