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

Shared by Jensen on 24-05-2024

Question 1

Question Type: **MultipleChoice**

A data scientist has written a feature engineering notebook that utilizes the pandas library. As the size of the data processed by the notebook increases, the notebook's runtime is drastically increasing, but it is processing slowly as the size of the data included in the process increases.

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

Options:

A- PySpark DataFrame API

B- pandas API on Spark

C- Spark SQL

D- Feature Store

Answer:

B

Explanation:

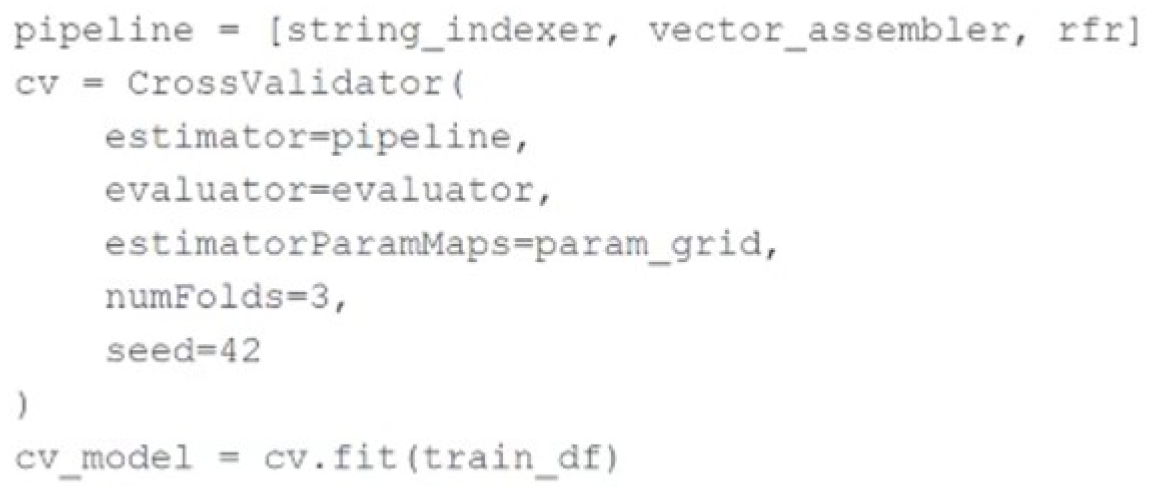
The pandas API on Spark provides a way to scale pandas operations to big data while minimizing the need for refactoring existing pandas code. It allows users to run pandas operations on Spark DataFrames, leveraging Spark's distributed computing capabilities to handle large datasets more efficiently. This approach requires minimal changes to the existing code, making it a convenient option for scaling pandas-based feature engineering notebooks.

Databricks documentation on pandas API on Spark: pandas API on Spark

Question 2

Question Type: MultipleChoice

A data scientist has developed a random forest regressor rfr and included it as the final stage in a Spark MLPipeline pipeline. They then set up a cross-validation process with pipeline as the estimator in the following code block:



```
pipeline = [string_indexer, vector_assembler, rfr]
cv = CrossValidator(
    estimator=pipeline,
    evaluator=evaluator,
    estimatorParamMaps=param_grid,
    numFolds=3,
    seed=42
)
cv_model = cv.fit(train_df)
```

Which of the following is a negative consequence of including pipeline as the estimator in the cross-validation process rather than rfr as the estimator?

Options:

A- The process will have a longer runtime because all stages of pipeline need to be refit or retransformed with each mode

B- The process will leak data from the training set to the test set during the evaluation phase

C- The process will be unable to parallelize tuning due to the distributed nature of pipeline

D- The process will leak data prep information from the validation sets to the training sets for each model

Answer:

A

Explanation:

Including the entire pipeline as the estimator in the cross-validation process means that all stages of the pipeline, including data preprocessing steps like string indexing and vector assembling, will be refit or retransformed for each fold of the cross-validation. This results in a longer runtime because each fold requires re-execution of these preprocessing steps, which can be computationally expensive.

If only the random forest regressor (rfr) were included as the estimator, the preprocessing steps would be performed once, and only the model fitting would be repeated for each fold, significantly reducing the computational overhead.

Databricks documentation on cross-validation: Cross Validation

Question 3

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 4

Question Type: **MultipleChoice**

Which of the following hyperparameter optimization methods automatically makes informed selections of hyperparameter values based on previous trials for each iterative model evaluation?

Options:

A- Random Search

B- Halving Random Search

C- Tree of Parzen Estimators

D- Grid Search

Answer:

C

Explanation:

Tree of Parzen Estimators (TPE) is a sequential model-based optimization algorithm that selects hyperparameter values based on the outcomes of previous trials. It models the probability density of good and bad hyperparameter values and makes informed decisions about which hyperparameters to try next.

This approach contrasts with methods like random search and grid search, which do not use information from previous trials to guide the search process.

Hyperopt and TPE

Question 5

Question Type: **MultipleChoice**

A data scientist learned during their training to always use 5-fold cross-validation in their model development workflow. A colleague suggests that there are cases where a train-validation split could be preferred over k-fold cross-validation when k > 2.

Which of the following describes a potential benefit of using a train-validation split over k-fold cross-validation in this scenario?

Options:

A- A holdout set is not necessary when using a train-validation split

B- Reproducibility is achievable when using a train-validation split

C- Fewer hyperparameter values need to be tested when using a train-validation split

D- Bias is avoidable when using a train-validation split

E- Fewer models need to be trained when using a train-validation split

Answer:

E

Explanation:

A train-validation split is often preferred over k-fold cross-validation (with k > 2) when computational efficiency is a concern. With a train-validation split, only two models (one on the training set and one on the validation set) are trained, whereas k-fold cross-validation requires training k models (one for each fold).

This reduction in the number of models trained can save significant computational resources and time, especially when dealing with large datasets or complex models.

Model Evaluation with Train-Test Split

Question 6

Question Type: MultipleChoice

A data scientist is performing hyperparameter tuning using an iterative optimization algorithm. Each evaluation of unique hyperparameter values is being trained on a single compute node. They are performing eight total evaluations across eight total compute nodes. While the accuracy of the model does vary over the eight evaluations, they notice there is no trend of improvement in the accuracy. The data scientist believes this is due to the parallelization of the tuning process.

Which change could the data scientist make to improve their model accuracy over the course of their tuning process?

Options:

A- Change the number of compute nodes to be half or less than half of the number of evaluations.

B- Change the number of compute nodes and the number of evaluations to be much larger but equal.

C- Change the iterative optimization algorithm used to facilitate the tuning process.

D- Change the number of compute nodes to be double or more than double the number of evaluations.

Answer:

C

Explanation:

The lack of improvement in model accuracy across evaluations suggests that the optimization algorithm might not be effectively exploring the hyperparameter space. Iterative optimization algorithms like Tree-structured Parzen Estimators (TPE) or Bayesian Optimization can adapt based on previous evaluations, guiding the search towards more promising regions of the hyperparameter space.

Changing the optimization algorithm can lead to better utilization of the information gathered during each evaluation, potentially improving the overall accuracy.

Hyperparameter Optimization with Hyperopt

Question 7

Question Type: **MultipleChoice**

A data scientist has a Spark DataFrame spark_df. They want to create a new Spark DataFrame that contains only the rows from spark_df where the value in column discount is less than or equal 0.

Which of the following code blocks will accomplish this task?

Options:

A- spark_df.loc[:,spark_df['discount'] <= 0]

B- spark_df[spark_df['discount'] <= 0]

C- spark_df.filter (col('discount') <= 0)

D- spark_df.loc(spark_df['discount'] <= 0, :]

Answer:

C

Explanation:

To filter rows in a Spark DataFrame based on a condition, the filter method is used. In this case, the condition is that the value in the 'discount' column should be less than or equal to 0. The correct syntax uses the filter method along with the col function from pyspark.sql.functions.

Correct code:

from pyspark.sql.functions import col filtered_df = spark_df.filter(col('discount') <= 0)

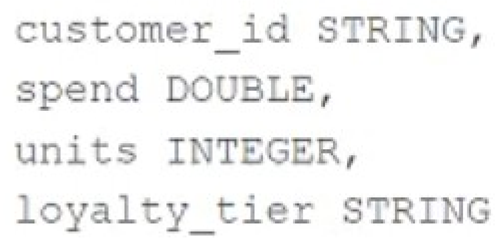
Option A and D use Pandas syntax, which is not applicable in PySpark. Option B is closer but misses the use of the col function.

PySpark SQL Documentation

Question 8

Question Type: **MultipleChoice**

A data scientist is working with a feature set with the following schema:



```
customer_id STRING,
spend DOUBLE,
units INTEGER,
loyalty_tier STRING
```

The customer_id column is the primary key in the feature set. Each of the columns in the feature set has missing values. They want to replace the missing values by imputing a common value for each feature.

Which of the following lists all of the columns in the feature set that need to be imputed using the most common value of the column?

Options:

A- customer_id, loyalty_tier

B- loyalty_tier

C- units

D- spend

E- customer_id

Answer:

B

Explanation:

For the feature set schema provided, the columns that need to be imputed using the most common value (mode) are typically the categorical columns. In this case, loyalty_tier is the only categorical column that should be imputed using the most common value. customer_id is a unique identifier and should not be imputed, while spend and units are numerical columns that should typically be imputed using the mean or median values, not the mode.

Databricks documentation on missing value imputation: Handling Missing Data

If you need any further clarification or additional questions answered, please let me know!

Question 9

Question Type: MultipleChoice

A data scientist has created a linear regression model that uses log(price) as a label variable. Using this model, they have performed inference and the predictions and actual label values are in Spark DataFrame preds_df.

They are using the following code block to evaluate the model:

regression_evaluator.setMetricName("rmse").evaluate(preds_df)

Which of the following changes should the data scientist make to evaluate the RMSE in a way that is comparable with price?

Options:

A- They should exponentiate the computed RMSE value

B- They should take the log of the predictions before computing the RMSE

C- They should evaluate the MSE of the log predictions to compute the RMSE

D- They should exponentiate the predictions before computing the RMSE

Answer:

D

Explanation:

When evaluating the RMSE for a model that predicts log-transformed prices, the predictions need to be transformed back to the original scale to obtain an RMSE that is comparable with the actual price values. This is done by exponentiating the predictions before computing the RMSE. The RMSE should be computed on the same scale as the original data to provide a meaningful measure of error.

Databricks documentation on regression evaluation: Regression Evaluation