Vendor: Databricks

Exam Code:DATABRICKS-MACHINE-LEARNING-ASSOCIATE

Exam Name: Databricks Certified Machine Learning Associate Exam

Version: Demo

QUESTION 1

A data scientist is using Spark ML to engineer features for an exploratory machine learning project.

They decide they want to standardize their features using the following code block: Upon code review, a colleague expressed concern with the features being standardized prior to splitting the data into a training set and a test set.

```
scaler = StandardScaler(
    withMean=True,
    inputCol="input_features",
    outputCol="output_features"
)
scaler_model = scaler.fit(features_df)
scaled_df = scaler_model.transform(features_df)
train_df, test_df = scaled_df.randomSplit([.8, .2], seed=42)
```

Which of the following changes can the data scientist make to address the concern?

A. Utilize the MinMaxScaler object to standardize the training data according to global minimum and maximum values

B. Utilize the MinMaxScaler object to standardize the test data according to global minimum and maximum values

C. Utilize a cross-validation process rather than a train-test split process to remove the need for standardizing data

D. Utilize the Pipeline API to standardize the training data according to the test data\\'s summary statistics

E. Utilize the Pipeline API to standardize the test data according to the training data\\'s summary statistics

Correct Answer: E

To address the concern about standardizing features prior to splitting the data, the correct approach is to use the Pipeline API to ensure that only the training data\\'s summary statistics are used to standardize the test data. This is achieved by

fitting the StandardScaler (or any scaler) on the training data and then transforming both the training and test data using the fitted scaler. This approach prevents information leakage from the test data into the model training process and

ensures that the model is evaluated fairly.

References:

Best Practices in Preprocessing in Spark ML (Handling Data Splits and Feature Standardization).

QUESTION 2

Which of the following tools can be used to parallelize the hyperparameter tuning process for single-node machine learning models using a Spark cluster?

A. MLflow Experiment Tracking

B. Spark ML

C. Autoscaling clusters

D. Autoscaling clusters

E. Delta Lake

Correct Answer: B

Spark ML (part of Apache Spark\\'s MLlib) is designed to handle machine learning tasks across multiple nodes in a cluster, effectively parallelizing tasks like hyperparameter tuning. It supports various machine learning algorithms that can be

optimized over a Spark cluster, making it suitable for parallelizing hyperparameter tuning for single-node machine learning models when they are adapted to run on Spark.

References:

Apache Spark MLlib Guide:https://spark.apache.org/docs/latest/ml-guide.html

Spark ML is a library within Apache Spark designed for scalable machine learning. It provides tools to handle large-scale machine learning tasks, including parallelizing the hyperparameter tuning process for single-node machine learning

models using a Spark cluster. Here\\'s a detailed explanation of how Spark ML can be used:

Hyperparameter Tuning with CrossValidator: Spark ML includes theCrossValidatorandTrainValidationSplitclasses, which are used for hyperparameter tuning. These classes can evaluate multiple sets of hyperparameters in parallel using a

Spark cluster. from pyspark.ml.tuning import CrossValidator, ParamGridBuilder from pyspark.ml.evaluation import BinaryClassificationEvaluator

Define the model

model = ...

Create a parameter grid

paramGrid = ParamGridBuilder() \

addGrid(model.hyperparam1, [value1, value2]) \

addGrid(model.hyperparam2, [value3, value4]) \

build()

Define the evaluator

evaluator = BinaryClassificationEvaluator()

Define the CrossValidator

crossval = CrossValidator(estimator=model,

estimatorParamMaps=paramGrid,

evaluator=evaluator,

numFolds=3)

Parallel Execution: Spark distributes the tasks of training models with different hyperparameters across the cluster\\'s nodes. Each node processes a subset of the parameter grid, which allows multiple models to be trained simultaneously.

Scalability: Spark ML leverages the distributed computing capabilities of Spark. This allows for efficient processing of large datasets and training of models across many nodes, which speeds up the hyperparameter tuning process significantly

compared to single-node computations.

References:

Apache Spark MLlib Documentation

Hyperparameter Tuning in Spark ML

QUESTION 3

The implementation of linear regression in Spark ML first attempts to solve the linear regression problem using matrix decomposition, but this method does not scale well to large datasets with a large number of variables.

Which of the following approaches does Spark ML use to distribute the training of a linear regression model for large data?

A. Logistic regression

B. Spark ML cannot distribute linear regression training

C. Iterative optimization

D. Least-squares method

E. Singular value decomposition

Correct Answer: C

For large datasets with many variables, Spark ML distributes the training of a linear regression model using iterative optimization methods. Specifically, Spark ML employs algorithms such as Gradient Descent or L-BFGS (Limited-memory

Broyden?letcher?oldfarb?hanno) to iteratively minimize the loss function. These iterative methods are suitable for distributed computing environments and can handle large-scale data efficiently by partitioning the data across nodes in a

cluster and performing parallel updates.References:

Spark MLlib Documentation (Linear Regression with Iterative Optimization).

QUESTION 4

Which of the following describes the relationship between native Spark DataFrames and pandas API on Spark DataFrames?

A. pandas API on Spark DataFrames are single-node versions of Spark DataFrames with additional metadata

B. pandas API on Spark DataFrames are more performant than Spark DataFrames

C. pandas API on Spark DataFrames are made up of Spark DataFrames and additional metadata

D. pandas API on Spark DataFrames are less mutable versions of Spark DataFrames

Correct Answer: C

The pandas API on Spark DataFrames are made up of Spark DataFrames with additional metadata. The pandas API on Spark aims to provide the pandas-like experience with the scalability and distributed nature of Spark. It allows users to

work with pandas functions on large datasets by leveraging Spark\\'s underlying capabilities.

References:

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

---

QUESTION 5

What is the name of the method that transforms categorical features into a series of binary indicator feature variables?

A. Leave-one-out encoding

B. Target encoding

C. One-hot encoding

D. Categorical

E. String indexing

Correct Answer: C

The method that transforms categorical features into a series of binary indicator variables is known as one-hot encoding. This technique converts each categorical value into a new binary column, which is essential for models that require

numerical input. One-hot encoding is widely used because it helps to handle categorical data without introducing a false ordinal relationship among categories.References:

Feature Engineering Techniques (One-Hot Encoding).

QUESTION 6

A data scientist is wanting to explore summary statistics for Spark DataFrame spark_df. The data scientist wants to see the count, mean, standard deviation, minimum, maximum, and interquartile range (IQR) for each numerical feature.

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

A. spark_df.summary ()

B. spark_df.stats()

C. spark_df.describe().head()

D. spark_df.printSchema()

E. spark_df.toPandas()

Correct Answer: A

Thesummary()function in PySpark\\'s DataFrame API provides descriptive statistics which include count, mean, standard deviation, min, max, and quantiles for numeric columns. Here are the steps on how it can be used:

Import PySpark:Ensure PySpark is installed and correctly configured in the Databricks environment.

Load Data:Load the data into a Spark DataFrame.

Apply Summary:Usespark_df.summary()to generate summary statistics. View Results:The output from thesummary()function includes the statistics specified in the query (count, mean, standard deviation, min, max, and potentially quartiles

which approximate the interquartile range).

References:

PySpark

Documentation:https://spark.apache.org/docs/latest/api/python/reference/api/pysp ark.sql.DataFrame.summary.html

---

QUESTION 7

A data scientist has been given an incomplete notebook from the data engineering team. The notebook uses a Spark DataFrame spark_df on which the data scientist needs to perform further feature engineering. Unfortunately, the data scientist has not yet learned the PySpark DataFrame API.

Which of the following blocks of code can the data scientist run to be able to use the pandas API on Spark?

A. import pyspark.pandas as ps df = ps.DataFrame(spark_df)

B. import pyspark.pandas as ps df = ps.to_pandas(spark_df)

C. spark_df.to_pandas()

D. import pandas as pd df = pd.DataFrame(spark_df)

Correct Answer: A

To use the pandas API on Spark, the data scientist can run the following code block:

importpyspark.pandasasps df = ps.DataFrame(spark_df) This code imports the pandas API on Spark and converts the Spark DataFramespark_df into a pandas-on-Spark DataFrame, allowing the data scientist to use familiar pandas

functions

for further feature engineering.

References:

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

---

QUESTION 8

Which statement describes a Spark ML transformer?

A. A transformer is an algorithm which can transform one DataFrame into another DataFrame

B. A transformer is a hyperparameter grid that can be used to train a model

C. A transformer chains multiple algorithms together to transform an ML workflow

D. A transformer is a learning algorithm that can use a DataFrame to train a model

Correct Answer: A

In Spark ML, a transformer is an algorithm that can transform one DataFrame into another DataFrame. It takes a DataFrame as input and produces a new DataFrame as output. This transformation can involve adding new columns, modifying

existing ones, or applying feature transformations. Examples of transformers in Spark MLlib include feature transformers likeStringIndexer,VectorAssembler, andStandardScaler.

References:

Databricks documentation on transformers: Transformers in Spark ML

QUESTION 9

A data scientist is attempting to tune a logistic regression model logistic using scikit-learn. They want to specify a search space for two hyperparameters and let the tuning process randomly select values for each evaluation.

They attempt to run the following code block, but it does not accomplish the desired task:

```
distributions = dict(C=uniform(loc=0, scale=4), penalty=['l2', 'l1'])
clf = GridSearchCV(logistic, distributions, random_state=0)
search = clf.fit(feature_data, target_data)
```

Which of the following changes can the data scientist make to accomplish the task?

A. Replace the GridSearchCV operation with RandomizedSearchCV

B. Replace the GridSearchCV operation with cross_validate

C. Replace the GridSearchCV operation with ParameterGrid

D. Replace the random_state=0 argument with random_state=1

E. Replace the penalty= [\\'12\\', \\'11\\'] argument with penalty=uniform (\\'12\\', \\'11\\')

Correct Answer: A

The user wants to specify a search space for hyperparameters and let the tuning process randomly select values.GridSearchCVsystematically tries every combination of the provided hyperparameter values, which can be computationally expensive and time-consuming.RandomizedSearchCV, on the other hand, samples hyperparameters from a distribution for a fixed number of iterations. This approach is usually faster and still can find very good parameters, especially when the search space is large or includes distributions. References: Scikit-Learn documentation on hyperparameter tuning: https://scikit-learn.org/stable/modules/grid_search.html#randomized-parameter-optimization

---

QUESTION 10

An organization is developing a feature repository and is electing to one-hot encode all categorical feature variables. A data scientist suggests that the categorical feature variables should not be one-hot encoded within the feature repository.

Which of the following explanations justifies this suggestion?

A. One-hot encoding is a potentially problematic categorical variable strategy for some machine learning algorithms.

B. One-hot encoding is dependent on the target variable\\'s values which differ for each apaplication.

C. One-hot encoding is computationally intensive and should only be performed on small samples of training sets for individual machine learning problems.

D. One-hot encoding is not a common strategy for representing categorical feature variables numerically.

Correct Answer: A

The suggestion not to one-hot encode categorical feature variables within the feature repository is justified because one-hot encoding can be problematic for some machine learning algorithms. Specifically, one-hot encoding increases the dimensionality of the data, which can be computationally expensive and may lead to issues such as multicollinearity and overfitting. Additionally, some algorithms, such as tree-based methods, can handle categorical variables directly without requiring one-hot encoding. References: Databricks documentation on feature engineering: Feature Engineering

---

QUESTION 11

A data scientist has been given an incomplete notebook from the data engineering team. The notebook uses a Spark DataFrame spark_df on which the data scientist needs to perform further feature engineering. Unfortunately, the data scientist has not yet learned the PySpark DataFrame API.

Which of the following blocks of code can the data scientist run to be able to use the pandas API on Spark?

A. import pyspark.pandas as ps df = ps.DataFrame(spark_df)

B. import pyspark.pandas as ps df = ps.to_pandas(spark_df)

C. spark_df.to_sql()

D. import pandas as pd df = pd.DataFrame(spark_df)

E. spark_df.to_pandas()

Correct Answer: A

To use the pandas API on Spark, which is designed to bridge the gap between the simplicity of pandas and the scalability of Spark, the correct approach involves importing the pyspark.pandas (recently renamed topandas_api_on_spark)

module and converting a Spark DataFrame to a pandas-on-Spark DataFrame using this API. The provided syntax correctly initializes a pandas-on-Spark DataFrame, allowing the data scientist to work with the familiar pandas-like API on large

datasets managed by Spark.

References:

Pandas API on Spark

Documentation:https://spark.apache.org/docs/latest/api/python/user_guide/pandas _on_spark/index.html

---

QUESTION 12

Which of the following machine learning algorithms typically uses bagging?

A. IGradient boosted trees

B. K-means

C. Random forest

D. Decision tree

Correct Answer: C

Random Forest is a machine learning algorithm that typically uses bagging (Bootstrap Aggregating). Bagging is a technique that involves training multiple base models (such as decision trees) on different subsets of the data and then combining their predictions to improve overall model performance. Each subset is created by randomly sampling with replacement from the original dataset. The Random Forest algorithm builds multiple decision trees and merges them to get a more accurate and stable prediction. References: Databricks documentation on Random Forest: Random Forest in Spark ML