DATABRICKS-MACHINE-LEARNING-ASSOCIATE$^{Q\&As}$

Databricks Certified Machine Learning Associate Exam

Pass Databricks DATABRICKS-MACHINE-LEARNING-ASSOCIATE Exam with 100% Guarantee

SATISFACTION GUARANTEED
100%
SATISFACTION GUARANTEED

QUESTION 1

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 not supported by most machine learning libraries.

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

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.

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

Correct Answer: E

One-hot encoding transforms categorical variables into a format that can be provided to machine learning algorithms to better predict the output. However, when done prematurely or universally within a feature repository, it can be

problematic:

Dimensionality Increase:One-hot encoding significantly increases the feature space, especially with high cardinality features, which can lead to high memory consumption and slower computation.

Model Specificity:Some models handle categorical variables natively (like decision trees and boosting algorithms), and premature one-hot encoding can lead to inefficiency and loss of information (e.g., ordinal relationships). Sparse Matrix

Issue:It often results in a sparse matrix where most values are zero, which can be inefficient in both storage and computation for some algorithms. Generalization vs. Specificity:Encoding should ideally be tailored to specific models and use

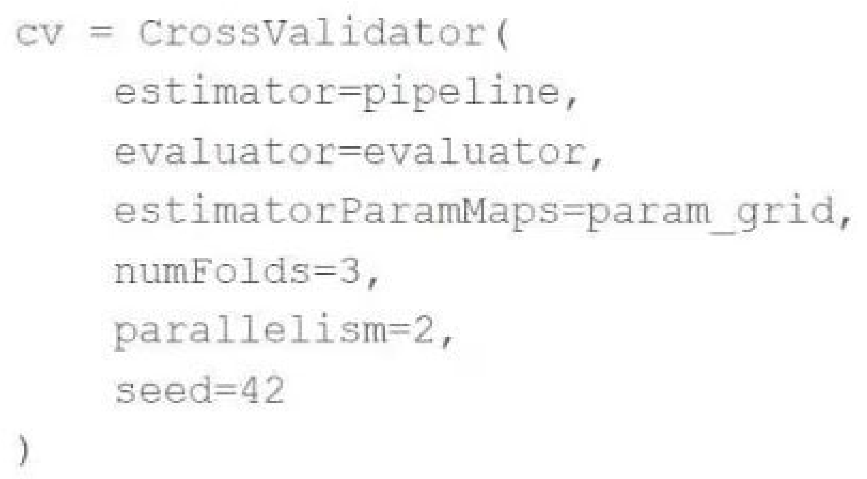
cases rather than applied generally in a feature repository.

References:

"Feature Engineering and Selection: A Practical Approach for Predictive Models" by Max Kuhn and Kjell Johnson (CRC Press, 2019).

QUESTION 2

A machine learning engineer is trying to scale a machine learning pipelinepipelinethat contains multiple feature engineering stages and a modeling stage. As part of the cross-validation process, they are using the following code block:



```
cv = CrossValidator(
     estimator=pipeline,
     evaluator=evaluator,
     estimatorParamMaps=param_grid,
     numFolds=3,
     parallelism=2,
     seed=42
)
```

A colleague suggests that the code block can be changed to speed up the tuning process by passing the model object to theestimatorparameter and then placing the updated cv object as the final stage of thepipelinein place of the original model.

Which of the following is a negative consequence of the approach suggested by the colleague?

A. The model will take longerto train for each unique combination of hvperparameter values

B. The feature engineering stages will be computed using validation data

C. The cross-validation process will no longer be

D. The cross-validation process will no longer be reproducible

E. The model will be refit one more per cross-validation fold

Correct Answer: B

If the model object is passed to the estimator parameter ofCrossValidator and the cross-validation object itself is placed as a stage in the pipeline, the feature engineering stages within the pipeline would be applied separately to each training

and validation fold during cross-validation. This leads to a significant issue: the feature engineering stages would be computed using validation data, thereby leaking information from the validation set into the training process. This would

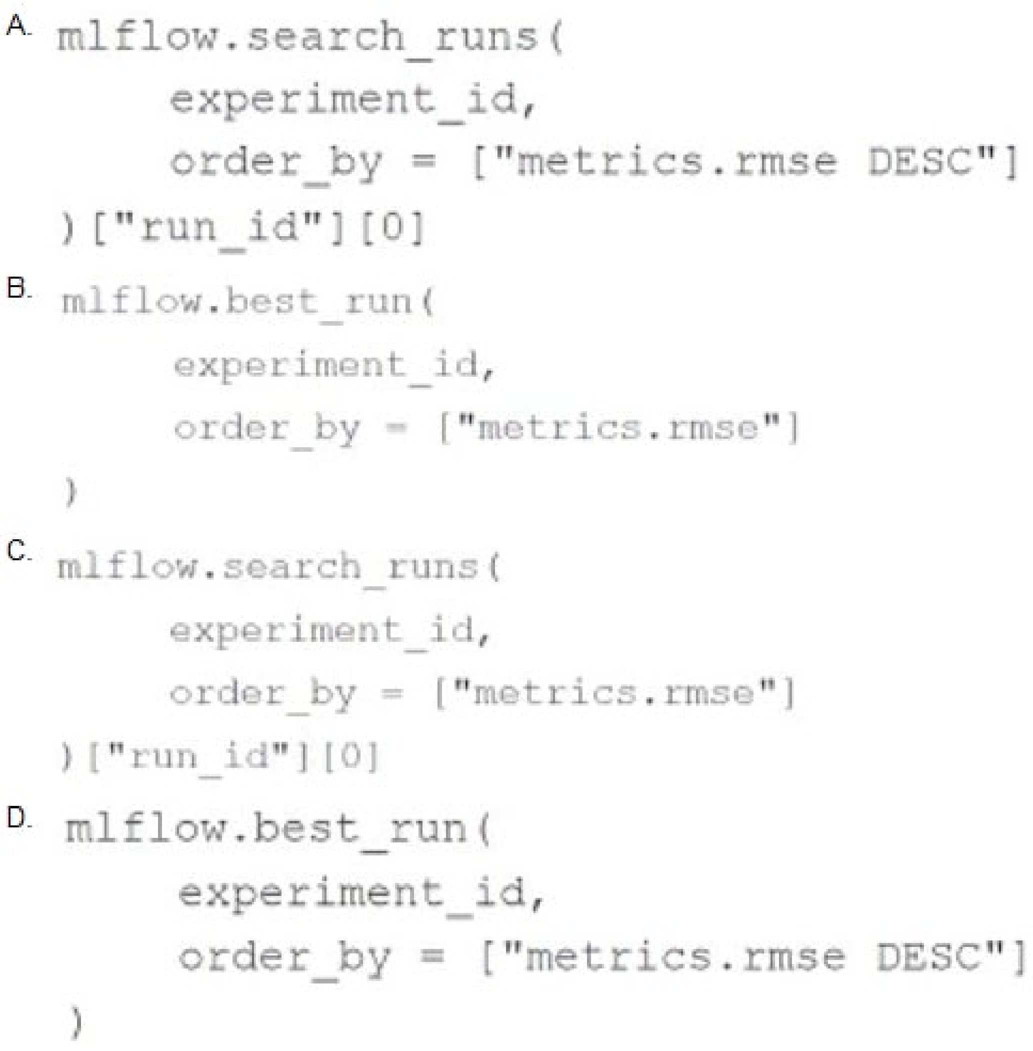
potentially invalidate the cross-validation results by giving an overly optimistic performance estimate.References:

Cross-validation and Pipeline Integration in MLlib (Avoiding Data Leakage in Pipelines).

QUESTION 3

A data scientist is utilizing MLflow Autologging to automatically track their machine learning experiments. After completing a series of runs for the experiment experiment_id, the data scientist wants to identify the run_id of the run with the best root-mean-square error (RMSE).

Which of the following lines of code can be used to identify the run_id of the run with the best RMSE in experiment_id?



A.
```
mlflow.search_runs(
      experiment_id,
      order_by = ["metrics.rmse DESC"]
)["run_id"][0]
```

B.
```
mlflow.best_run(
      experiment_id,
      order_by = ["metrics.rmse"]
)
```

C.
```
mlflow.search_runs(
      experiment_id,
      order_by = ["metrics.rmse"]
)["run_id"][0]
```

D.
```
mlflow.best_run(
      experiment_id,
      order_by = ["metrics.rmse DESC"]
)
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: C

To find the run_id of the run with the best root-mean-square error (RMSE) in an MLflow experiment, the correct line of code to use is:

mlflow.search_runs(experiment_id, order_by=["metrics.rmse"])["run_id"][0] This line of code searches the runs in the specified experiment, orders them by the RMSE metric in ascending order (the lower the RMSE, the better), and retrieves

the run_id of the best-performing run. Option C correctly represents this logic.

References:

MLflow documentation on tracking experiments:

https://www.mlflow.org/docs/latest/python_api/mlflow.html#mlflow.search_runs

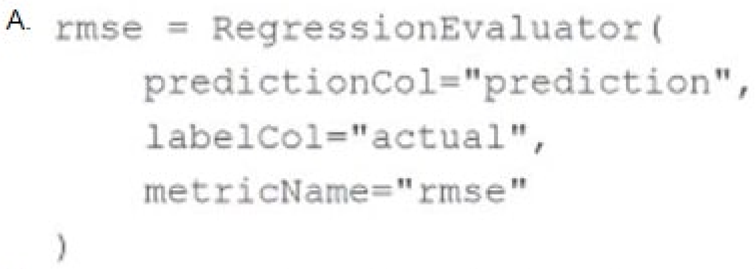
QUESTION 4

A data scientist has developed a linear regression model using Spark ML and computed the predictions in a Spark DataFrame preds_df with the following schema:

prediction DOUBLE actual DOUBLE

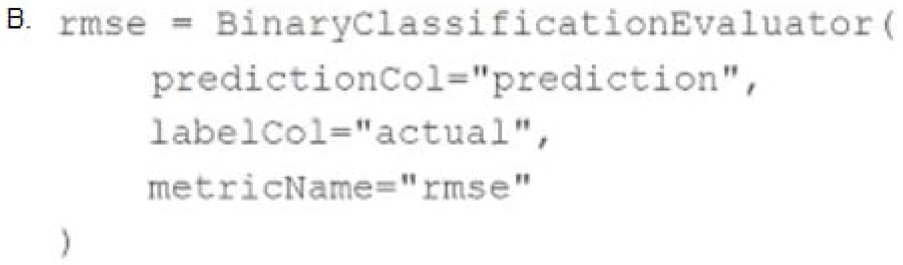
Which of the following code blocks can be used to compute the root mean-squared-error of the model according to the data in preds_df and assign it to the rmse variable?

A.



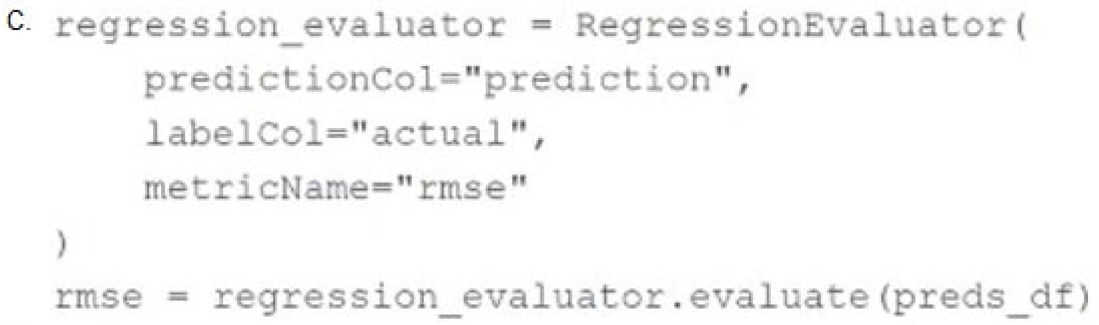
```
rmse = RegressionEvaluator(
    predictionCol="prediction",
    labelCol="actual",
    metricName="rmse"
)
```

B.



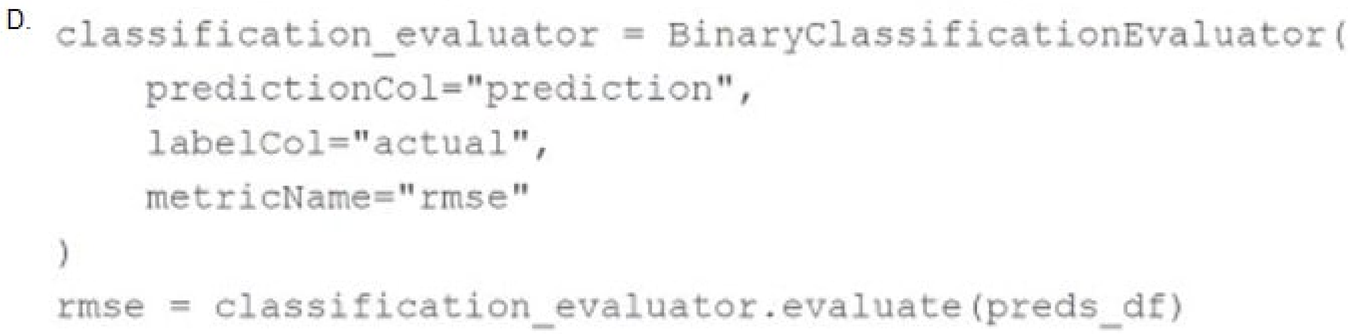
```
rmse = BinaryClassificationEvaluator(
    predictionCol="prediction",
    labelCol="actual",
    metricName="rmse"
)
```

C.



```
regression_evaluator = RegressionEvaluator(
    predictionCol="prediction",
    labelCol="actual",
    metricName="rmse"
)
rmse = regression_evaluator.evaluate(preds_df)
```

D.



```
classification_evaluator = BinaryClassificationEvaluator(
    predictionCol="prediction",
    labelCol="actual",
    metricName="rmse"
)
rmse = classification_evaluator.evaluate(preds_df)
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: C

To compute the root mean-squared-error (RMSE) of a linear regression model using Spark ML, theRegressionEvaluatorclass is used. TheRegressionEvaluator is specifically designed for regression tasks and can calculate various metrics,

including RMSE, based on the columns containing predictions and actual values. The correct code block to compute RMSE from thepreds_dfDataFrame is:

regression_evaluator = RegressionEvaluator(predictionCol="prediction", labelCol="actual", metricName="rmse") rmse = regression_evaluator.evaluate(preds_df) This code creates an instance ofRegressionEvaluator, specifying the prediction

and label columns, as well as the metric to be computed ("rmse"). It then evaluates the predictions in preds_dfand assigns the resulting RMSE value to thermsevariable. Options A and B incorrectly useBinaryClassificationEvaluator, which is

not suitable for regression tasks. Option D also incorrectly usesBinaryClassificationEvaluator.

References:

PySpark ML Documentation

QUESTION 5

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?

A. fmin

B. SparkTrials

C. quniform

D. search_space

E. objective_function

Correct Answer: B

TheSparkTrialsclass 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.

fromhyperoptimportfmin, tpe, hp, SparkTrials search_space = {\\'x\\': hp.uniform(\\'x\\',0,1),\\'y\\':

hp.uniform(\\'y\\',0,1) }defobjective(params):returnparams[\\'x\\'] **2+ params[\\'y\\'] **2spark_trials = SparkTrials(parallelism=4) best = fmin(fn=objective, space=search_space, algo=tpe.suggest, max_evals=100, trials=spark_trials) References:

Hyperopt Documentation

QUESTION 6

A machine learning engineer is converting a decision tree from sklearn to Spark ML. They notice that they are receiving different results despite all of their data and manually specified hyperparameter values being identical.

Which of the following describes a reason that the single-node sklearn decision tree and the Spark ML decision tree can differ?

A. Spark ML decision trees test every feature variable in the splitting algorithm

B. Spark ML decision trees automatically prune overfit trees

C. Spark ML decision trees test more split candidates in the splitting algorithm

D. Spark ML decision trees test a random sample of feature variables in the splitting algorithm

E. Spark ML decision trees test binned features values as representative split candidates

Correct Answer: E

One reason that results can differ between sklearn and Spark ML decision trees, despite identical data and hyperparameters, is that Spark ML decision trees test binned feature values as representative split candidates. Spark ML uses a

method called "quantile binning" to reduce the number of potential split points by grouping continuous features into bins. This binning process can lead to different splits compared to sklearn, which tests all possible split points directly. This

difference in the splitting algorithm can cause variations in the resulting trees.References:

Spark MLlib Documentation (Decision Trees and Quantile Binning).

QUESTION 7

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. Singular value decomposition

C. Iterative optimization

D. Least-squares method

Correct Answer: C

For large datasets, Spark ML uses iterative optimization methods to distribute the training of a linear regression model. Specifically, Spark MLlib employs techniques like Stochastic Gradient Descent (SGD) and Limited-memory Broyden

QUESTION 8

A machine learning engineer has created a Feature Table new_table using Feature Store Client fs. When creating the table, they specified a metadata description with key information about the Feature Table. They now want to retrieve that metadata programmatically.

Which of the following lines of code will return the metadata description?

A. There is no way to return the metadata description programmatically.

B. fs.create_training_set("new_table")

C. fs.get_table("new_table").description

D. fs.get_table("new_table").load_df()

E. fs.get_table("new_table")

Correct Answer: C

To retrieve the metadata description of a feature table created using the Feature Store Client (referred here asfs), the correct method involves callingget_tableon thefsclient with the table name as an argument, followed by accessing

thedescription attribute of the returned object. The code snippetfs.get_table("new_table").description correctly achieves this by fetching the table object for "new_table" and then accessing its description attribute, where the metadata is stored.

The other options do not correctly focus on retrieving the metadata description.References:

Databricks Feature Store documentation (Accessing Feature Table Metadata).

QUESTION 9

In which of the following situations is it preferable to impute missing feature values with their median value over the mean value?

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 missingno values

Correct Answer: C

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.

References:

Data Imputation Techniques (Dealing with Outliers).

QUESTION 10

A data scientist has developed a machine learning pipeline with a static input data set using Spark ML, but the pipeline is taking too long to process. They increase the number of workers in the cluster to get the pipeline to run more efficiently. They notice that the number of rows in the training set after reconfiguring the cluster is different from the number of rows in the training set prior to reconfiguring the cluster.

Which of the following approaches will guarantee a reproducible training and test set for each model?

A. Manually configure the cluster

B. Write out the split data sets to persistent storage

C. Set a speed in the data splitting operation

D. Manually partition the input data

Correct Answer: B

To ensure reproducible training and test sets, writing the split data sets to persistent storage is a reliable approach. This allows you to consistently load the same training and test data for each model run, regardless of cluster reconfiguration

or other changes in the environment.

Correct approach:

Split the data.

Write the split data to persistent storage (e.g., HDFS, S3). Load the data from storage for each model training session. train_df, test_df = spark_df.randomSplit([0.8,0.2], seed=42) train_df.write.parquet("path/to/train_df.parquet")

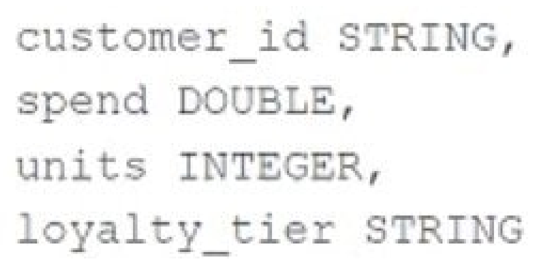
test_df.write.parquet("path/to/test_df.parquet")# Later, load the datatrain_df = spark.read.parquet("path/to/train_df.parquet") test_df = spark.read.parquet("path/to/test_df.parquet")

References:

Spark DataFrameWriter Documentation

QUESTION 11

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



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

Thecustomer_idcolumn 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?

A. customer_id, loyalty_tier

B. loyalty_tier

C. units

D. spend

E. customer_id

Correct Answer: B

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_tieris the only categorical column that should be imputed using the

most common value.customer_idis a unique identifier and should not be imputed, whilespendandunits are numerical columns that should typically be imputed using the mean or median values, not the mode.

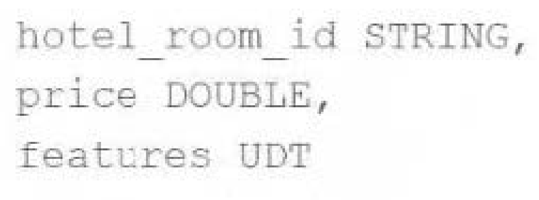
References:

Databricks documentation on missing value imputation: Handling Missing Data If you need any further clarification or additional questions answered, please let me know!

QUESTION 12

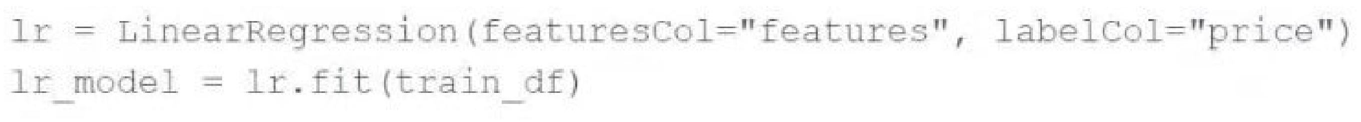
A machine learning engineer would like to develop a linear regression model with Spark ML to predict the price of a hotel room. They are using the Spark DataFrametrain_dfto train the model.

The Spark DataFrametrain_dfhas the following schema:



```
hotel_room_id STRING,
price DOUBLE,
features UDT
```

The machine learning engineer shares the following code block:



```
lr = LinearRegression(featuresCol="features", labelCol="price")
lr_model = lr.fit(train_df)
```

Which of the following changes does the machine learning engineer need to make to complete the task?

A. They need to call the transform method on train df

B. They need to convert the features column to be a vector

C. They do not need to make any changes

D. They need to utilize a Pipeline to fit the model

E. They need to split thefeaturescolumn out into one column for each feature

Correct Answer: B

In Spark ML, the linear regression model expects the feature column to be a vector type. However, if thefeaturescolumn in the DataFrametrain_dfis not already in this format (such as being a column of type UDT or a non-vectorized type), the engineerneeds to convert it to a vector column using a transformer likeVectorAssembler. This is a critical step in preparing the data for modeling as Spark ML models require input features to be combined into a single vector column. References: Spark MLlib documentation forLinearRegression:https://spark.apache.org/docs/latest/ml-classification-regression.html#linear-regression

QUESTION 13

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

A. F1

B. R-squared

C. MAE

D. MSE

Correct Answer: A

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.

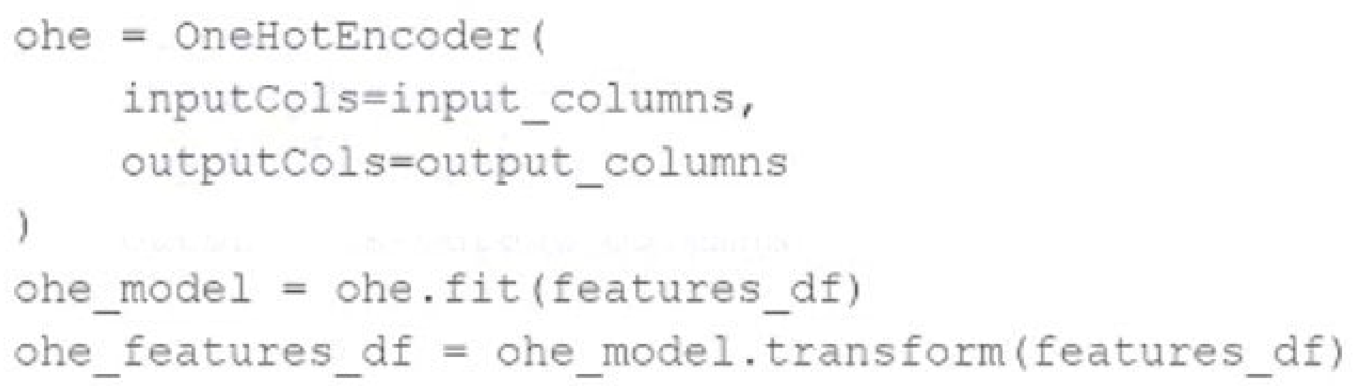
References:

Databricks documentation on Feature Store: Feature Store in Databricks

QUESTION 14

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

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?

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 useVectorAssemblerprior to one-hot encoding the features.

Correct Answer: C

TheOneHotEncoderin Spark ML requires numerical indices as inputs rather than string labels. Therefore, you need to first convert the string columns to numerical indices usingStringIndexer. After that, you can applyOneHotEncoderto these

indices.

Corrected code:

frompyspark.ml.featureimportStringIndexer, OneHotEncoder# Convert string column to indexindexers = [StringIndexer(inputCol=col,

outputCol=col+"_index")forcolininput_columns] indexer_model = Pipeline(stages=indexers).fit(features_df) indexed_features_df = indexer_model.transform(features_df)# One-hot encode the indexed columnsohe = OneHotEncoder

(inputCols=[col+"_index"forcolininput_columns], outputCols=output_columns) ohe_model = ohe.fit(indexed_features_df) ohe_features_df = ohe_model.transform(indexed_features_df)

References:

PySpark ML Documentation

QUESTION 15

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