Databricks Certified Machine Learning Associate

Section 1: Databricks Machine Learning

Databricks ML

- Identify when a standard cluster is preferred over a single-node cluster andvice versa
- Connect a repo from an external Git provider to Databricks repos.
- Commit changes from a Databricks Repo to an external Git provider.
- Create a new branch and commit changes to an external Git provider.
- Pull changes from an external Git provider back to a Databricks workspace.
- Orchestrate multi-task ML workflows using Databricks jobs.

Databricks Runtime for Machine Learning

- Create a cluster with the Databricks Runtime for Machine Learning.
- Install a Python library to be available to all notebooks that run on a cluster.

AutoML

- Identify the steps of the machine learning workflow completed by AutoML.
- Identify how to locate the source code for the best model produced by AutoML.
- Identify which evaluation metrics AutoML can use for regression problems.
- Identify the key attributes of the data set using the AutoML data exploration notebook.

Feature Store

- Describe the benefits of using Feature Store to store and access features for machine learning pipelines.
- Create a feature store table.
- Write data to a feature store table.
- Train a model with features from a feature store table.
- Score a model using features from a feature store table.

Managed MLfiow

- Identify the best run using the MLflow Client API.
- Manually log metrics, artifacts, and models in an MLflow Run.
- Create a nested Run for deeper Tracking organization.
- Locate the time a run was executed in the MLflow UI.
- Locate the code that was executed with a run in the MLflow UI.
- Register a model using the MLflow Client API.

- Transition a model’s stage using the Model Registry UI page.
- Transition a model’s stage using the MLflow Client API.
- Request to transition a model’s stage using the ML Registry UI page.

Section 2: ML Workfiows

Exploratory Data Analysis

- Compute summary statistics on a Spark DataFrame using .summary()
- Compute summary statistics on a Spark DataFrame using dbutils data summaries.
- Remove outliers from a Spark DataFrame that are beyond or less than a designated threshold.

Feature Engineering

- Identify why it is important to add indicator variables for missing values thathave been imputed or replaced.
- Describe when replacing missing values with the mode value is an appropriate way to handle missing values.
- Compare and contrast imputing missing values with the mean value or median value.
- Impute missing values with the mean or median value.
- Describe the process of one-hot encoding categorical features.
- Describe why one-hot encoding categorical features can be inefficient fortree-based models.

Training

- Perform random search as a method for tuning hyperparameters.
- Describe the basics of Bayesian methods for tuning hyperparameters.
- Describe why parallelizing sequential/iterative models can be difficult.
- Understand the balance between compute resources and parallelization.
- Parallelize the tuning of hyperparameters using Hyperopt and SparkTrials.
- Identify the usage of SparkTrials as the tool that enables parallelization fortuning single-node models.

Evaluation and Selection

- Describe cross-validation and the benefits of downsides of usingcross-validation over a train-validation split.
- Perform cross-validation as a part of model fitting.
- Identify the number of models being trained in conjunction with a grid-search and cross-validation process.
- Describe Recall and F1 as evaluation metrics.

- Identify the need to exponentiate the RMSE when the log of the label variableis used.
- Identify that the RMSE has not been exponentiated when the log of the label variable is used.

Section 3: Spark ML

Distributed ML Concepts

- Describe some of the difficulties associated with distributing machine learning models.
- Identify Spark ML as a key library for distributing traditional machine learningwork.
- Identify scikit-learn as a single-node solution relative to Spark ML.

Spark ML Modeling APIs

- Split data using Spark ML.
- Identify key gotchas when splitting distributed data using Spark ML.
- Train / evaluate a machine learning model using Spark ML.
- Describe Spark ML estimator and Spark ML transformer.
- Develop a Pipeline using Spark ML.
- Identify key gotchas when developing a Spark ML Pipeline.

Hyperopt

- Identify Hyperopt as a solution for parallelizing the tuning of single-node models.
- Identify Hyperopt as a solution for Bayesian hyperparameter inference for distributed models.
- Parallelize the tuning of hyperparameters for Spark ML models using Hyperopt and Trials.
- Identify the relationship between the number of trials and model accuracy.

Pandas API on Spark

- Describe key differences between Spark DataFrames and Pandas on Spark DataFrames.
- Identify the usage of an InternalFrame making Pandas API on Spark not quiteas fast as native Spark.
- Identify Pandas API on Spark as a solution for scaling data pipelines withoutmuch refactoring.
- Convert data between a PySpark DataFrame and a Pandas on Spark DataFrame.
- Identify how to import and use the Pandas on Spark APIs.

Pandas UDFs/Function APIs

- Identify Apache Arrow as the key to Pandas <-> Spark conversions.
- Describe why iterator UDFs are preferred for large data.
- Apply a model in parallel using a Pandas UDF.
- Identify that pandas code can be used inside of a UDF function.
- Train / apply group-specific models using the Pandas Function API.

Section 4: Scaling ML

ModelsModel

Distribution

- Describe how Spark scales linear regression.
- Describe how Spark scales decision trees.

Ensembling Distribution

- Describe the basic concepts of ensemble learning.
- Compare and contrast bagging, boosting, and stacking.