DBKS Certified Machine Learning Associate

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 learning work.
- [ ] 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 the 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 quite as fast as native Spark.
- [ ] Identify Pandas API on Spark as a solution for scaling data pipelines without much 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.
- [ ] Identify the benefits of using Pandas UDFs

Section 4: Scaling ML Models

Model 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.

DBKS Certified Machine Learning Professional

Section 1: Experimentation

Data Management

- [ ] Read and write a Delta table
- [ ] View Delta table history and load a previous version of a Delta table
- [ ] Create, overwrite, merge, and read Feature Store tables in machine learning workflows

Experiment Tracking

- [ ] Manually log parameters, models, and evaluation metrics using MLflow
- [ ] Programmatically access and use data, metadata, and models from MLflow experiments

Advanced Experiment Tracking

- [ ] Perform MLflow experiment tracking workflows using model signatures and input examples
- [ ] Identify the requirements for tracking nested runs
- [ ] Describe the process of enabling autologging, including with the use of Hyperopt
- [ ] Log and view artifacts like SHAP plots, custom visualizations, feature data, images, and metadata

Section 2: Model Lifecycle Management

Preprocessing Logic

- [ ] Describe an MLflow flavor and the benefits of using MLflow flavors
- [ ] Describe the advantages of using the pyfunc MLflow flavor
- [ ] Describe the process and benefits of including preprocessing logic and context in custom model classes and objects

Model Management

- [ ] ● Describe the basic purpose and user interactions with Model Registry ● Programmatically register a new model or new model version.
- [ ] ● Add metadata to a registered model and a registered model version
- [ ] ● Identify, compare, and contrast the available model stages
- [ ] ● Transition, archive, and delete model versions

Model Lifecycle Automation

- [ ] ● Identify the role of automated testing in ML CI/CD pipelines
- [ ] ● Describe how to automate the model lifecycle using Model Registry Webhooks and Databricks Jobs
- [ ] ● Identify advantages of using Job clusters over all-purpose clusters
- [ ] ● Describe how to create a Job that triggers when a model transitions between stages, given a scenario
- [ ] ● Describe how to connect a Webhook with a Job
- [ ] ● Identify which code block will trigger a shown webhook
- [ ] ● Identify a use case for HTTP webhooks and where the Webhook URL needs to come. ● Describe how to list all webhooks and how to delete a webhook

Section 3: Model Deployment

Batch

- [ ] ● Describe batch deployment as the appropriate use case for the vast majority of deployment use cases
- [ ] ● Identify how batch deployment computes predictions and saves them somewhere for later use
- [ ] ● Identify live serving benefits of querying pre-computed batch predictions
- [ ] ● Identify less performant data storage as a solution for other use cases
- [ ] ● Load registered models with load_model
- [ ] ● Deploy a single-node model in parallel using spark_udf

- [ ] • Identify z-ordering as a solution for reducing the amount of time to read predictions from a table
- [ ] • Identify partitioning on a common column to speed up querying
- [ ] • Describe the practical benefits of using the score_batch operation

Streaming

- [ ] • Describe Structured Streaming as a common processing tool for ETL pipelines
- [ ] • Identify structured streaming as a continuous inference solution on incoming data
- [ ] • Describe why complex business logic must be handled in streaming deployments
- [ ] • Identify that data can arrive out-of-order with structured streaming
- [ ] • Identify continuous predictions in a time-based prediction store as a scenario for streaming deployments
- [ ] • Identify continuous predictions in a time-based prediction store as a scenario for streaming deployments
- [ ] • Convert a batch deployment pipeline inference to a streaming deployment pipeline
- [ ] • Convert a batch deployment pipeline writing to a streaming deployment pipeline

Real-time

- [ ] • Describe the benefits of using real-time inference for a small number of records or when fast prediction computations are needed
- [ ] • Identify JIT feature values as a need for real-time deployment
- [ ] • Describe model serving deploys and endpoint for every stage
- [ ] • Identify how model serving uses one all-purpose cluster for a model deployment
- [ ] • Query a Model Serving enabled model in the Production stage and Staging stage
- [ ] • Identify how cloud-provided RESTful services in containers is the best solution for production-grade real-time deployments

Section 4: Solution and Data Monitoring

Drift Types

- [ ] • Compare and contrast label drift and feature drift
- [ ] • Identify scenarios in which feature drift and/or label drift are likely to occur
- [ ] • Describe concept drift and its impact on model efficacy

Drift Tests and Monitoring

- [ ] • Describe summary statistic monitoring as a simple solution for numeric feature drift
- [ ] • Describe mode, unique values, and missing values as simple solutions for categorical

feature drift

- ☐ ● Describe tests as more robust monitoring solutions for numeric feature drift than simple summary statistics
- ☐ ● Describe tests as more robust monitoring solutions for categorical feature drift than simple summary statistics
- ☐ ● Compare and contrast Jenson-Shannon divergence and Kolmogorov-Smirnov tests for numerical drift detection
- ☐ ● Identify a scenario in which a chi-square test would be useful

Comprehensive Drift Solutions

- ☐ ● Describe a common workflow for measuring concept drift and feature drift ● Identify when retraining and deploying an updated model is a probable solution to drift
- ☐ ● Test whether the updated model performs better on the more recent data

COMPLETED SECTIONS

Section 1: Databricks Machine Learning

Databricks ML

- ☑ ~~Identify when a standard cluster is preferred over a single-node cluster and vice 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.~~

- [x] ~~Train a model with features from a feature store table.~~
- [x] ~~Score a model using features from a feature store table.~~

Managed MLflow

- [x] ~~Identify the best run using the MLflow Client API.~~
- [x] ~~Manually log metrics, artifacts, and models in an MLflow Run.~~
- [x] ~~Create a nested Run for deeper Tracking organization.~~
- [x] ~~Locate the time a run was executed in the MLflow UI.~~
- [x] ~~Locate the code that was executed with a run in the MLflow UI.~~
- [x] ~~Register a model using the MLflow Client API.~~
- [x] ~~Transition a model's stage using the Model Registry UI page.~~
- [x] ~~Transition a model's stage using the MLflow Client API.~~
- [x] ~~Request to transition a model's stage using the ML Registry UI page.~~

Section 2: ML Workflows

Exploratory Data Analysis

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

Feature Engineering

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

Training

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

Evaluation and Selection

- [x] ~~Describe cross-validation and the benefits of downsides of using cross-validation over a train-validation split.~~
- [x] ~~Perform cross-validation as a part of model fitting.~~
- [x] ~~Identify the number of trained models in conjunction with a grid-search and cross-validation process.~~
- [x] ~~Describe Recall and F1 as evaluation metrics.~~
- [x] ~~Identify the need to exponentiate the RMSE when the log of the label variable is used.~~
- [x] ~~Identify that the RMSE has not been exponentiated when the log of the label variable is used.~~