WILFRID LAURIER UNIVERSITY

LAURIER

Inspiring Lives.

Assignment #3

CP422 – Programming for Big Data

Department of Physics and Computer Science, Faculty of Science, Waterloo Campus

Fall – 2024

Classification and Regression on the NYC Taxi Trip Data (January 2015) using Apache Spark on Databricks

Instructions:

Submission Folder: Please submit to this folder on the Dropbox: “CP422-A3”
File Name and Format: Please rename your submission file to “Group-ID-CP422-A3.pdf.” Please make sure to replace the “ID” with your group number. The only accepted file type is PDF.

1. Use appropriate comments and markdowns to explain your code and findings.
2. Visualizations should be clear and well-labelled.
3. Submit the notebook with the final report summarizing the analyses in PDF format.

This programming assignment must be performed on the “yellow_tripdata_2015-01.csv” dataset using **Spark on Databricks**.

Objective: This assignment focuses on using Spark and Databricks to build, optimize, and evaluate classification and regression pipelines on the "yellow_tripdata_2015-01.csv" dataset. You will apply various stages of data preparation, feature engineering, model building, and evaluation. The goal is to build reusable and optimized pipelines that can predict specific outcomes from the dataset and save these pipelines for future use.

Dataset: The dataset "yellow_tripdata_2015-01.csv" contains information on NYC yellow taxi trips during January 2015. It includes features such as pick-up and drop-off times, locations, fare amounts, and more.

Task Breakdown

This assignment consists of two primary tasks: **Classification** and **Regression**. For each task, you will develop and compare two different pipelines, each optimized through hyperparameter tuning. In both cases, you’ll measure performance and save the best-performing pipeline for future use.

Task 1: Classification

Objective: Build a classification pipeline to predict whether a trip resulted in a high fare (e.g., over $20) based on trip characteristics.

Instructions:

1. **Data Preparation:**

    a) **Load Dataset:** Load the CSV file into a Spark DataFrame.
    b) **Data Exploration:** Explore the dataset to understand the variables and handle missing or invalid values.

c) **Feature Engineering:** Create new features or modify existing ones to improve the model. For example:
   - Time-based features (e.g., hour, day of the week) from the pickup or drop-off times.
   - Distance calculation between pickup and drop-off coordinates.

d) **Target Variable Creation: Define a binary target column high_fare where:**
   - `1` if the fare is above $20.
   - `0` if the fare is $20 or below.

e) **Data Splitting: Split the data into training (70%) and testing (30%) sets.**

2. **Pipeline 1:** Decision Tree Classifier Pipeline
   a. Define the pipeline stages:
      i. **Feature Transformers** (VectorAssembler, StandardScaler, etc.).
      ii. **Model:** Decision Tree Classifier.
   b. **Hyperparameter Tuning:** Use CrossValidator with GridSearch to find the best parameters (e.g., max depth, min instances per node).
   c. **Model Training:** Train the pipeline on the training data.
   d. **Model Evaluation:** Evaluate the performance on test data using metrics like F1 Score, Precision, and Recall.
   e. **Save Pipeline:** Save the trained pipeline.

3. **Pipeline 2:** Logistic Regression Pipeline
   a) Define a new pipeline with Logistic Regression as the classifier.
   b) Perform hyperparameter tuning on Logistic Regression parameters (e.g., regularization parameter, max iterations).
   c) Evaluate the model performance on test data and compare with the Decision Tree Classifier pipeline.
   d) Save the trained pipeline.

4. **Report Findings:**

   a) Discuss the performance of each pipeline and which hyperparameters were chosen.
   b) State the best-performing pipeline and explain why it performed better.

Task 2: Regression

Objective: Build a regression pipeline to predict the fare amount based on trip characteristics.

Instructions:

1. **Data Preparation:**
   a) **Load Dataset:** Use the same dataset and load it into a new DataFrame.
   b) **Feature Engineering:** Similar to the classification task, but focus on features that might help predict fare amount.
      - Time-based features (pickup hour, day of week).
      - Trip distance and trip duration.
   c) Data Splitting: Split the data into training (70%) and testing (30%) sets.

2. **Pipeline 1:** Linear Regression Pipeline

   a) Define the pipeline stages:

        - **Feature Transformers** (VectorAssembler, StandardScaler, etc.).
        - **Model:** Linear Regression.

    b) Hyperparameter Tuning: Use CrossValidator with GridSearch to tune hyperparameters (e.g., regularization parameter, max iterations).

    c) Model Training: Train the pipeline on the training data.

    d) Model Evaluation: Evaluate the model using RMSE (Root Mean Squared Error) and R^2 Score on the test data.

    e) Save Pipeline: Save the trained pipeline.

3. **Pipeline 2:** Random Forest Regressor Pipeline

    a) Define a new pipeline with Random Forest Regressor.

    b) Perform hyperparameter tuning on Random Forest parameters (e.g., number of trees, max depth).

    c) Evaluate the model performance on test data and compare it with the Linear Regression pipeline.

    d) Save the trained pipeline.

4. **Report Findings:**

    a) Discuss the performance of each pipeline and provide a comparison.

    b) Justify which pipeline you would choose for future predictions based on the evaluation metrics.

Additional Requirements

1. **Performance Comparison:** Compare and discuss the performance of each model on the test data for both classification and regression tasks. Summarize the best pipeline for each task.
2. **Saving and Loading Pipelines:** Demonstrate how to save each pipeline and provide code to reload it. Explain why saving pipelines is important for real-world applications.
3. **Documentation:** Provide documentation in your code that explains each step, the purpose of each pipeline stage, and the chosen hyperparameters.
4. **Presentation:** Prepare a brief report with visualizations of model performance and a summary of hyperparameter tuning results.

Submission Requirements

a) **Databricks Notebook:** Export your Databricks Notebook as a html file, then to PDF and merge it with the PDF from b.

b) **Summary Report (PDF):** Provide a brief report summarizing your process, model evaluation results, hyperparameter tuning results, and findings.

Grading Rubric

1. **Pipeline Construction (40%)**

    a) Completeness and correctness of each pipeline.

    b) Appropriateness of feature engineering and data transformation steps.

2. **Hyperparameter Tuning (20%)**

    a) Correct implementation of hyperparameter tuning for both pipelines.

    b) Quality of selected hyperparameters based on performance.

3. **Model Evaluation and Comparison (20%)**

    a) Correct calculation of performance metrics.

    b) Clear presentation and interpretation of results.

4. **Documentation and Presentation (20%)**

    a) Clarity and detail in code documentation.

    b) Quality of the report summarizing findings and supporting your choices.