WILFRID LAURIER UNIVERSITY

LAURIER

Inspiring Lives.

CP 422 – Programming for Big Data

Fall 2025

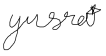
Group 10

Assignment #3

Member 1 Name: Faiza Azam ID: 169020565
Member 2 Name: Wardah Arain ID: 169016554
Member 3 Name: Amany Neeyamuthkhan ID: 169025721
Member 4 Name: Yusra Hassan ID: 169024293
Member 5 Name: Sara Aljaafari ID: 169044425
Member 6 Name: Anna Doneva ID: 169042350

Submission Date: 26-11-25

All listed members have contributed, read, and approved this submission. All listed members have acknowledged each team member's equal and reasonable contribution to this submission.

Member 1 Name: Faiza Azam	Signature: Faiza Azam
Member 2 Name: Wardah Arain	Signature: 
Member 3 Name: Amany Neeyamuthkhan	Signature: AN
Member 4 Name: Yusra Hassan	Signature: 
Member 5 Name: Sara Aljaafari	Signature: SA
Member 6 Name: Anna Doneva	Signature: AD

Task 1: Classification

For this task, we used the January 2015 Yellow Taxi dataset to build two classification models that predict whether a taxi trip will result in a high fare (greater than $20). The work involved preparing the dataset, creating new features, training two pipelines, tuning their hyperparameters, and comparing their performance.

Data Preparation

We loaded the dataset from the Databricks workspace and cleaned it by removing rows with missing or invalid information. Trips with zero distance, zero or negative fare, or unrealistic trip durations were filtered out. After ensuring data quality, we engineered several features that help the model understand the behavior of each trip, such as:

- Trip duration in minutes
- Trip speed (distance / duration)
- Pickup hour
- Day of the week

We also created the binary target variable high_fare, where 1 represents fares above $20 and 0 represents fares at or below $20. The dataset was then split into a 70% training set and a 30% test set.

Pipeline 1: Decision Tree Classifier

The first model was a Decision Tree pipeline that included a VectorAssembler followed by a Decision Tree Classifier. We performed cross-validation to select the best configuration.
The tuning grid explored variations in the tree's maximum depth and minimum instances per node. The best model selected:

- maxDepth = 10
- minInstancesPerNode = 1

When evaluated on the test set, the Decision Tree achieved excellent results:

- F1 Score: 0.9902
- Precision: 0.9941
- Recall: 0.9949

These results show that the Decision Tree was able to capture the non-linear relationships in the data extremely well. Features like trip duration and speed contributed strongly to the model's success.

Pipeline 2: Logistic Regression Classifier

The second model followed a similar pipeline structure, using a VectorAssembler and a Logistic Regression classifier. We tuned the regularization parameter and the number of iterations using cross-validation. The best Logistic Regression model used:

- regParam = 0.01
- maxIter = 10

Its performance on the test set was strong but not as high as the Decision Tree:

- F1 Score: 0.8343
- Precision: 0.8870
- Recall: 0.9994

Logistic Regression performed well overall, especially in recall, meaning it almost never missed an actual high-fare trip. However, because it is a linear model, it cannot capture complex patterns in the data as effectively as the Decision Tree. This explains the lower precision and F1 score.

Model Comparison

The Decision Tree outperformed Logistic Regression in overall balanced accuracy (F1), and it offered the best trade-off between precision and recall.

Logistic Regression provided a strong baseline and excellent recall but introduced more false positives.

Best classification pipeline: Decision Tree Classifier. It achieved the highest F1 score and balanced precision and recall, making it reliable for identifying high-fare trips with minimal error.

Task 2: Regression

For the regression task, the goal was to predict the exact taxi fare amount using trip characteristics such as trip distance, duration, pickup time, and day of week. The same dataset as Task 1 was used, but the modeling approach and evaluation metrics differed because the target variable here is continuous.

Data Preparation

We began by creating a new DataFrame specifically for regression. The following steps ensured the data was clean and suitable for continuous prediction:

- Removed trips with invalid or missing values, including zero distance, zero duration, or negative fares.
- Applied light outlier filtering to drop extreme, unrealistic values (e.g., fares over $200).
- Used the same engineered features as in the classification task, including:
    - trip_duration (in minutes)
    - Pickup_hour
    - Pickup_day_of_week
    - Trip_distance
    - passenger_count

After preprocessing, the dataset was split 70/30 into training and testing sets. This resulted in approximately 69,000 training rows and 29,000 test rows, which was a strong sample size for regression.

Pipeline 1: Linear Regression

A full regression pipeline was built using:

- VectorAssembler to combine numerical features
- StandardScaler to normalize feature ranges (helpful for optimization)
- Linear Regression as the model

Hyperparameter tuning was done using CrossValidator with a grid over:

- regParam (regularization strength), maxIter (maximum iterations)

Best model (Linear Regression)

- regParam = 0.0; maxIter = 50

Performance on the test set

- RMSE: 2.8211 & R^2: 0.9163

This means the Linear Regression model explains 91.6% of the variance in fare amounts. The predictions were stable, smooth, and consistently close to the actual fares. For this dataset, linear relationships (e.g., fare increasing proportionally with distance) helped Linear Regression perform extremely well.

Pipeline 2: Random Forest Regression

The second pipeline replaced the linear model with a Random Forest Regressor to capture potential nonlinear relationships.

The pipeline included: VectorAssembler, RandomForestRegressor

Hyperparameter tuning was done over: numTrees, maxDepth

Best model (Random Forest): numTrees = 50, maxDepth = 10

Performance on the test set

- RMSE: 3.1802 & R^2: 0.8936

Random Forest performed strongly and produced realistic predictions, but its overall accuracy was slightly lower than Linear Regression. This is expected when the underlying relationships are mostly linear.

Comparison of Models

Linear Regression provided the best numerical accuracy, with the lowest RMSE and the highest R^2 score. Random Forest performed well but did not surpass the linear model, likely because fare amounts in this dataset follow a mostly linear structure.

Best Regression Pipeline: Linear Regression. It produced the lowest RMSE and highest R^2, making it the most accurate and interpretable for predicting fare amounts.