Big Data Management

Project nr 4: Airline Delay and Cancellation Prediction with Spark ML

Team members: Eidi Paas, Pirjo Vainjärv

GitHub link: https://github.com/pirjo2/bdm-projects

Provided dataset

The dataset used for this project consists of U.S. domestic flight data from the year 2009, provided in CSV format. It includes multiple features related to flight schedules, delays, cancellations, and routes. The dataset was loaded into Apache Spark for distributed processing and analysis.

Key attributes include:

- FL_DATE: Flight date
- ORIGIN and DEST: Origin and destination airport codes
- DEP_DELAY and ARR_DELAY: Departure and arrival delays
- CANCELLED: Cancellation status
- DISTANCE: Flight distance
- Several other operational timing details.

1. Data Ingestion and Preparation

The dataset is ingested using PySpark's SparkSession. After loading the CSV file, an initial schema is displayed and the first few records are inspected to understand the structure and contents. The schema includes multiple columns detailing flight schedules, delays, cancellations, and additional flight-related data, such as origin, destination, and carrier.

After loading the data, the df2009 DataFrame is partitioned by the month of flight dates and saved in the Parquet format for more efficient querying and processing.

2. Cleaning and Preprocessing

The dataset is cleaned by:

- Dropping unnecessary columns like "Unnamed: 27."
- Renaming columns for clarity, such as renaming OP_CARRIER to UniqueCarrier and OP_CARRIER_FL_NUM to UniqueCarrierFlightNumber.
- Removing rows where critical information like flight dates and carrier information is missing.
- Filtering out diverted flights as they don't contribute to the analysis of cancellations and delays.

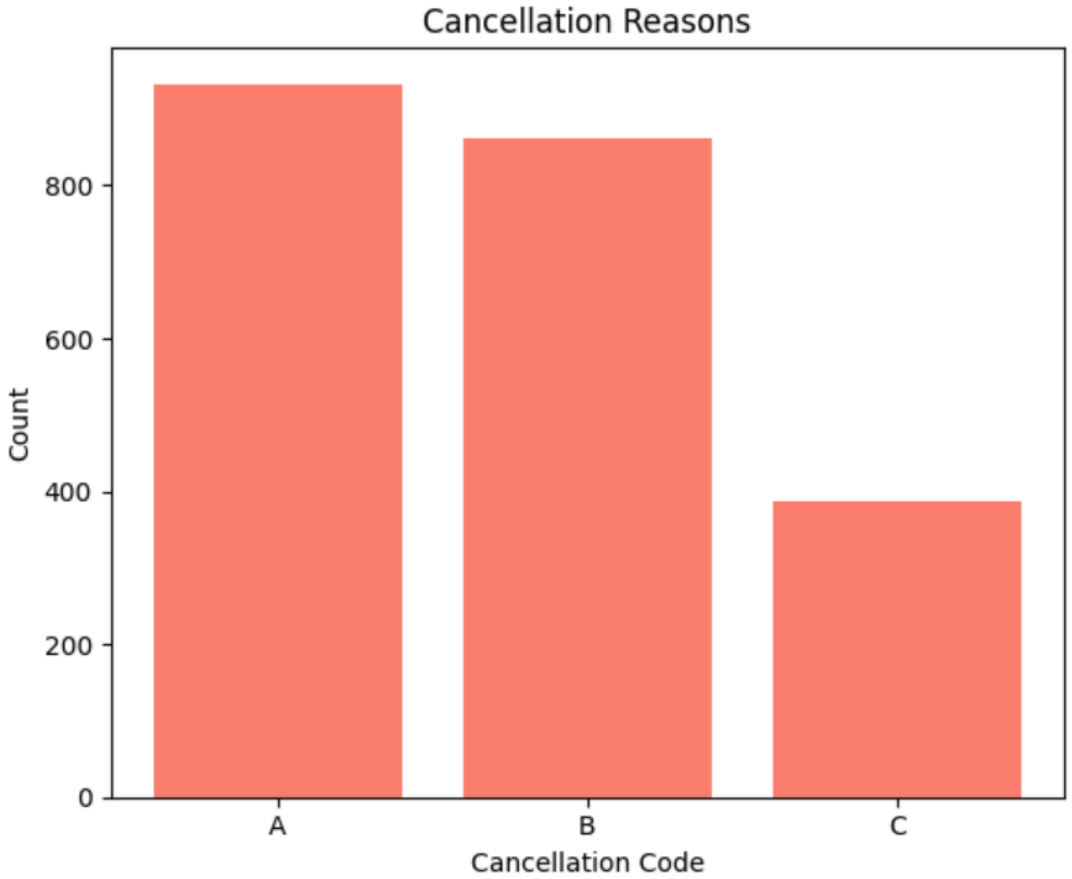
Additionally, new columns for "Month" and "DayOfWeek" are derived from the FL_DATE column, which aids in further analysis and feature creation.

3. Exploratory Analysis

Initial analysis reveals that cancellations are rare, with only 2,183 cancellations out of 125,874 flights. The cancellation reasons are categorized under different codes, such as:

A. Airline/Carrier issues
B. Weather-related problems
C. National Air System issues

The distribution of cancellations by carrier and cancellation reason is visualized, showing the breakdown of reasons for cancellations.



4. Feature Engineering

To prepare the data for machine learning, several categorical columns (UniqueCarrier, ORIGIN, DEST) are encoded using StringIndexer and OneHotEncoder. Numerical features such as CRS_ELAPSED_TIME, CRS_DEP_TIME, and CRS_ARR_TIME are directly included for prediction purposes. These features are combined into a single vector using VectorAssembler, which is a necessary step for Spark ML algorithms.

The final df_prepared DataFrame includes the encoded features, which are then used to build and train machine learning models. Data cleaning steps have also removed any rows with null values in the critical columns.

5. Modeling

In this section, we focus on building and evaluating several machine learning models to predict the target variable using the features from the dataset. The task is to train and evaluate four models: Logistic Regression, Decision Tree, Random Forest, and Gradient Boosted Trees (GBT).

5.1. Data Splitting

To ensure that our model generalizes well, we begin by splitting the dataset into training and test sets. This allows us to train the models on one subset of the data and evaluate their performance on a separate, unseen subset. We use a 70/30 split, where 70% of the data is used for training, and 30% is reserved for testing. This is a common practice to ensure the model's performance is validated on data it hasn't seen during training.

5.2. Model Training

We will now train four models. These models include:

- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier
- Gradient Boosting Trees (GBT)

Logistic Regression

Logistic Regression is a simple yet effective model that works well with linearly separable data. We will train this model and evaluate its performance.

Decision Tree Classifier

A Decision Tree is a non-linear model that can capture complex relationships in the data. We will train a basic Decision Tree model.

Random Forest Classifier

A Random Forest is an ensemble model that aggregates predictions from multiple Decision Trees. It tends to perform better than a single Decision Tree.

Gradient Boosting Trees (GBT)

GBT builds an ensemble of trees sequentially, where each tree attempts to correct the mistakes of the previous ones. This method often performs well in terms of both speed and accuracy.

5.3. Model Evaluation

We evaluate the models using two key metrics: Accuracy and AUC (Area Under the Curve).

- Accuracy measures the proportion of correctly classified instances.
- AUC provides a measure of how well the model discriminates between classes.

5.4. Hyperparameter Tuning with Cross-Validation

To improve the performance of these models, we can perform hyperparameter tuning. We will use a 3-fold CrossValidator to tune at least one hyperparameter for each model. This helps to find the best model configuration for each algorithm.

For simplicity, we can tune hyperparameters like the maximum depth for the Decision Tree and Random Forest, and learning rate for the GBT.

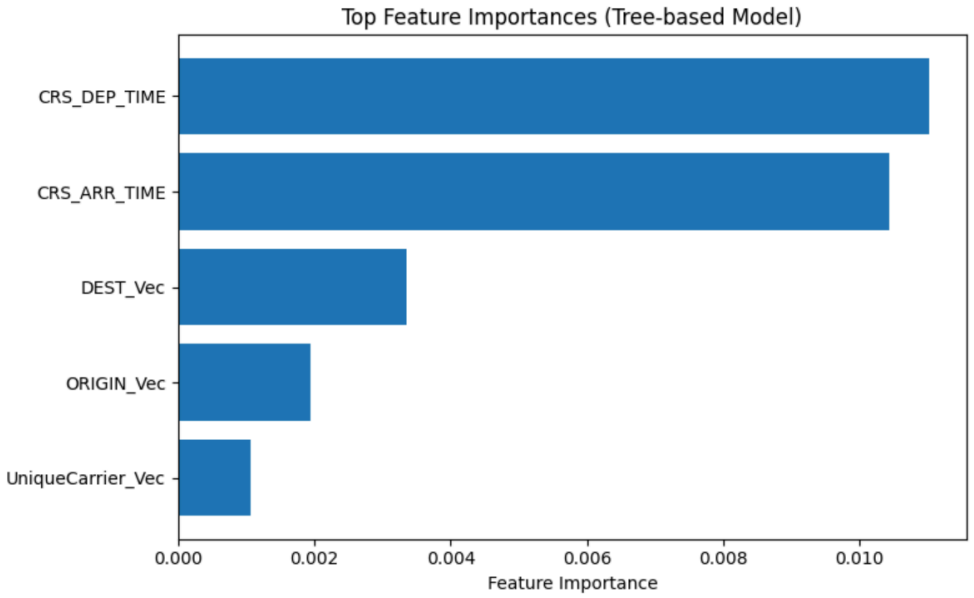
6. Explainability

For this section, we will focus on understanding and interpreting the importance of features in our best tree-based model. We will extract and visualize the feature importances from the Random Forest model, which is a tree-based model that is known to provide feature importance scores.

6.1. Feature Importance

Random Forest models provide an attribute called feature_importances_, which indicates how useful each feature is for making predictions. We will visualize these importances using a bar chart.

This visualization helps us identify which features are the most important in making predictions with the model. By understanding feature importance, we can gain insights into the underlying data and make informed decisions for further model improvements or feature engineering.



7. Model Persistence and Inference

7.1. Saving the Best Model

Once the model is trained, we save it for future use, allowing us to load and use the model without retraining it each time.

7.2. Loading and Scoring the 2010 Data

After training and saving the best-performing model, we used it to make predictions on unseen flight data from January 1, 2010. For each flight, the model outputs whether it is likely to be cancelled (1.0) or not (0.0), along with the probability scores for both outcomes. For example, a prediction of 0.0 with a probability like [0.95, 0.05] means the model is 95% confident the flight will not be cancelled. In our results, the model consistently predicted that the flights were not cancelled, with high confidence, indicating its ability to generalize well to new data.

```
+----------+-------------+------+----+----------+----------------------------------------+
|FL_DATE   |UniqueCarrier|ORIGIN|DEST|prediction|probability                             |
+----------+-------------+------+----+----------+----------------------------------------+
|2010-01-01|MQ           |LGA   |RDU |0.0       |[0.9483567518363261,0.05164324816367394] |
|2010-01-01|MQ           |DCA   |JFK |0.0       |[0.9654440185125537,0.03455598148744632] |
|2010-01-01|MQ           |LGA   |RDU |0.0       |[0.9487387266563063,0.05126127334369368] |
|2010-01-01|MQ           |RDU   |LGA |0.0       |[0.9412965754278627,0.058703424572137286]|
|2010-01-01|MQ           |JFK   |DCA |0.0       |[0.9628903077709916,0.037109692229008395]|
|2010-01-01|MQ           |DCA   |JFK |0.0       |[0.9651078244876456,0.03489217551235435] |
|2010-01-01|MQ           |RDU   |LGA |0.0       |[0.9420405100571779,0.05795948994282207] |
|2010-01-01|MQ           |DCA   |JFK |0.0       |[0.9671711537300101,0.03282884626998994] |
|2010-01-01|MQ           |DCA   |JFK |0.0       |[0.967945050857446,0.03205494914255402]  |
|2010-01-01|MQ           |JFK   |DCA |0.0       |[0.963827072752645,0.03617292724735499]  |
+----------+-------------+------+----+----------+----------------------------------------+
only showing top 10 rows
```

Conclusion

This pipeline sets the stage for building predictive models using machine learning techniques like logistic regression or decision trees, utilizing the processed features to predict airline cancellations. Further steps will include training these models, evaluating their performance, and tuning hyperparameters for the best prediction accuracy.