UNIVERSIDAD POLITÉCNICA DE MADRID

Escuela Técnica Superior de Ingenieros Informáticos

SPARK PRACTICAL APPLICATION – FIRST SEMESTER

2023-2024

FLIGTHS DELAYS

BIG DATA

MSc. HEALTH AND MEDICAL DATA ANALYTICS

MSc. DATA SCIENCE

Laura Teresa Martínez Marquina

Andrés Román de Vicente

Enrique Martínez Martel

INDEX

1. Introduction

In the current era, where the aviation industry plays a crucial role, the efficiency and punctuality of flights are fundamental aspects. In this context, the use of big data techniques has become essential for analysing vast amounts of data and extracting meaningful patterns that can enhance flight management and reduce delays.

Our project focuses on addressing the challenge of flight delays using Apache Spark, specifically the PySpark API, a powerful tool for distributed processing and large-scale data analysis. The database we will employ is the "Data Expo 2009: Airline on-time data," accessible at:

https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/HG7NV7.

This database provides a comprehensive set of information about flights, including details such as delays, cancellations, airlines, airports, among others. Through Apache Spark, we aim to process this enormous amount of data efficiently and build a flight delay prediction model. This model will enable airlines and operators to make informed and proactive decisions to improve the punctuality of their services.

The distributed processing capabilities of Spark will allow handling large datasets in a scalable manner, while the flexibility of PySpark will facilitate the development and implementation of machine learning algorithms for delay prediction. This project will contribute not only to the optimization of airline operations but also to opening the door to future research and applications in the field of data-driven aviation.

2. Data

This dataset includes details about the arrival and departure of commercial flights in the United States, spanning from October 1987 to April 2008. It is an extensive dataset, comprising almost 120 million records in total, occupying 1.6 gigabytes zipped and a total amount of 12 gigabytes unzipped. Due to the substantial volume of data involved and the impossibility of processing it with the computer, **a small portion of this dataset** has been chosen, specifically the flights corresponding to the years **1988, 1998, and 2008**. The dataset contains 29 columns:

- **Year**: from 1987 to 2008 (we have only work with 3 years tough. 1988, 1998, 2008).
- **Month**: 1 (January) to 12 (December).
- **DayofMonth**: 1 to 31.
- **DayOfWeek**: 1 (Monday) to 7 (Sunday).
- **DepTime**: real departure time (hhmm).
- **CRSDepTime**: Schedule departure time (hhmm).
- **ArrTime**: real arrival time (hhmm).
- **CRSArrTime**: Schedule arrival time (hhmm).
- **UniqueCarrier**: airline code.
- **FlightNum**: flight number
- **TailNum**: airplane number
- **ActualElapsedTime**: real flight time in minutes (includes time in airport).
- **CRSElapsedTime**: schedule flight time in minutes (includes time in airport).
- **AirTime**: real time in the air in minutes.
- **ArrDelay**: arrival delay in minutes.

- **DepDelay**: departure delay in minutes.
- **Origin**: origin airport IATA code.
- **Dest**: destination airport IATA code.
- **Distance**: distance between origin and destination in miles.
- **TaxiIn**: airport entry time in minutes.
- **TaxiOut**: airport exit time in minutes.
- **Cancelled**: whether the flight was cancelled (1) or not (0)
- **CancellationCode**: cancelation code for carrier (A), weather (B), NAS (C) or security (D)
- **Diverted**: whether the flight was diverted (1) or not (0).
- **CarrierDelay**: delay due to the airline in minutes.
- **WeatherDelay**: delay due to the weather in minutes.
- **NASDelay**: delay due to the National Aviation System (NAS) in minutes.
- **SecurityDelay**: delay due to security in minutes.
- **LateAircraftDelay**: delay due to aircrafts in minutes.

Some columns have been filtered at the beginning of the analysis since they contain unknown information at the time the plane takes off, these are the variables in red: [**ArrTime, ActualElapsedTime, AirTime, TaxiIn, Diverted, CarrierDelay, WeatherDela, NASDelay, SecurityDelay, LateAircraftDelay**]

As previously explained, the main objective of this report is to detect patterns that can enhance flight management and reduce delays. Therefore, **ArrDelay** is the target variable for the prediction model.

3. Variables

3.1 Original Variables Selected

In the construction of the predictive model for estimating commercial flight arrival delays, a variable selection process was applied to identify and capture key attributes from the original dataset. The list of variables and the reasons for selecting them are detailed in Table 1.

Also, it is worth noting that some of the following variables were transformed since they could not be used in their original form. This transformation is detailed in following sections.

Variable	Reason
Year	The year might capture trends or patterns that affect flight delays, such as changes in airline policies, airport infrastructure upgrades, or advancements in aviation technology.
Month	Seasonal variations can impact flight operations. For example, weather conditions, holidays, and special events might influence the likelihood of delays.
DayofMonth	Certain days of the month may experience higher or lower air traffic, affecting delays. For instance, the beginning or end of the month might have different travel patterns due to business or financial reasons.

DayOfWeek	Weekdays and weekends can exhibit distinct travel patterns. Business-related flights might dominate weekdays, while weekends may see more leisure travel.
DepTime	It is crucial for capturing the departure delays, which can significantly impact arrival delays. It provides a direct measure of the departure performance and is likely to be a strong predictor of the overall flight delay.
CRSDepTime	Scheduled departure time is essential for understanding the planned timeline of a flight. Comparing actual departure time with the scheduled time can reveal insights into punctuality and deviations from the original flight plan.
CRSArrTime	Scheduled arrival time helps the model understand the intended timeline for the flight. Differences between actual and scheduled arrival times can highlight variations that contribute to arrival delays.
UniqueCarrier	The airline code is crucial for capturing variations in the operational efficiency and historical performance of different airlines, providing valuable insights into factors influencing arrival delays.
FlightNum	Different flights may have different historical performance records. Including flight numbers could help the model capture specific characteristics associated with certain flights, such as the airline's punctuality record.
CRSElapsedTime	Scheduled flight time represents the anticipated duration of the flight, serving as a fundamental parameter for predicting arrival delays. Comparing the scheduled flight time with the actual duration can reveal if a flight's performance aligns with its planned timeline.
DepDelay	The departure delay is a strong predictor of arrival delay. Flights that depart late often arrive late. Including this variable allows the model to account for the initial delay in its predictions.
Origin	The origin airport is a critical geographical factor that can influence flight operations. Different airports may have unique characteristics, such as varying levels of congestion, weather patterns, or infrastructure, all of which can impact arrival delays.
Dest	The destination airport is essential for capturing the geographical context of the flight. Arrival delays can be influenced by factors at the destination, such as airport capacity, weather conditions, and regional air traffic.
Distance	Longer flights might have different dynamics affecting delays compared to shorter flights. Including the distance variable helps the model differentiate between short-haul and long-haul flights.

Table 1: List of original variables selected.

3.2 Original Variables Excluded

The filtering of variables was driven by the objective of improving the model accuracy and interpretability, ensuring that each selected variable played a meaningful role in predicting commercial flight delays. Apart from the forbidden variables stated by the practical work's statement, in Table 2 the other excluded variables are listed.

Variable	Reason
TailNum	Airplane numbers may serve as unique identifiers but often lack predictive power for arrival delays. The model is more interested in operational aspects and airline-specific characteristics rather than the specific identification number of the aircraft. Also, it was noticed a substantial number of missing values, especially in the first years, making it particularly useless.
TaxiOut	It was decided to filter it out because of the huge proportion of missing values that this variable had (around 40% for the years selected). This makes the variable useless.
CancellationCode	The variable has been deleted because all cancelled flights have been dropped out from the dataset. The main reason is because they do not have arrival time therefore, they do not provide information.
Cancelled	This variable will be used to filter out all cancelled flights as these flights does not have a value for the arrival delay. Then it will be dropped.

Table 2: List of original variables excluded.

3.3 Original Variables Transformations Performed

In this section, a detailed explanation for the variable transformations applied in the analysis is provided, see Table 3.

Variable (new name after transformation)	Transformation	Justification
Year (year)	Converted from string to float.	These variables represent temporal aspects and are crucial for understanding patterns over time. Converting them to numerical format allows the model to capture temporal relationships.
Month (month)		
DayofMonth (day_of_month)		
DayOfWeek (day_of_week)		

DepTime (actual_departure_time_mins)	Convert time (string in hh:mm format) to minutes (float). To achieve this, a new column for each variable was created with the time in minutes and then dropped the original ones. To format the time in minutes the first two digits were multiply by 60 and then added to the last two digits.	These variables represent time-related information and converting them to a numerical format facilitates their use in the model, allowing it to recognize patterns associated with departure and arrival times.
CRSDepTime (scheduled_departure_time_mins)		
CRSArrTime (scheduled_arrival_time_mins)		
CRSElapsedTime (scheduled_flight_time_mins)		
UniqueCarrier (airline_encoded)	Encoded using StringIndexer and OneHotEncoding[1]	These variables are categorical and encoding them provides a numeric representation for the model to understand and incorporate airline and flight number information into the prediction.
FlightNum (flight_number)		
DepDelay (departure_delay)	Converted from string to float.	This feature provides crucial information about the departure delay, which is a key factor in predicting the arrival delays. Converting them to numerical format enables the model to utilize this information effectively.
Origin (origin_encoded)	Encoded using StringIndexer and OneHotEncoding.	Airport codes are categorical variables and encoding them allows the model to interpret the relationships between different origins and destinations in the prediction of arrival delays.
Dest (dest_encoded)		
Distance (distance)	Converted from string to float.	Distance is a numerical variable representing the flight distance and converting it to float enables the model to consider the linear relationship between distance and arrival delay.

Table 3: List of variables transformation.

Also, a *Missing Values* handling was performed. Some variables, as specified before, with too many values missing ("TailNum" 40%, "CancellationCode" 81%, "TaxiOut" 40%) were entirely dropped. Others with

[1] Since we are meant to use a linear model (LinearRegression), encoding our categorical variables using only StrinIndexer could lead to confusion, because the model could assume an ordinal relationship between the indexes. OneHotEncoding creates a set of binary columns (one for each unique value of the variable) indicating the presence/absence of the category.

fewer missing values (“ArrDelay”, “CRSElapsedTime”), were decided to be dropped only the rows containing them (considering the size of the dataset it is not a big loss).

3.4 New Variables Created

No new variables were added.

4. Machine Learning Technique

Linear Regression is a supervised machine learning algorithm used for predicting a continuous target variable based on one or more independent variables. In this context, it models the relationship between various features (such as temporal information, flight details, and distance) and the target variable (ArrDelay).

4.1 Parameters for the Linear Regression

These parameters and hyperparameters are essential for configuring the behaviour of the linear regression model. Adjusting these values can have a significant impact on the model's performance, and the choice of values often involves experimentation and tuning based on the characteristics of the dataset.

- **Features**: The chosen features for the linear regression model include temporal variables (year, month, day_of_month, day_of_week), time-related variables (actual_departure_time_mins, scheduled_departure_time_mins, scheduled_arrival_time_mins), categorical variables (airline_encoded, origin_encoded, dest_encoded), flight-specific variables (flight_number, scheduled_flight_time_mins, departure_delay), and distance. These features are selected based on their potential influence on flight delays.
- **Normalization**: The Normalizer is applied to normalize the features. Normalization is crucial to ensure that all features contribute equally to the model, especially when dealing with variables on different scales.
- **Regularization**: Elastic Net regularization is employed with parameters *regParam*=0.3 and *elasticNetParam*=0.8. Elastic Net combines L1 and L2 regularization, providing a balance between feature selection and preventing overfitting. It is particularly useful when dealing with datasets with potentially correlated features.
- **Model Evaluation Metrics**: The model is evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2). These metrics provide insights into the accuracy and goodness of fit of the regression model.

4.2 Justification for Linear Regression

These are the reasons why linear regression was chosen:

- **Interpretability**: Linear regression provides a straightforward interpretation of the relationship between each feature and flight delays, allowing for a clear understanding of the contributing factors.
- **Ease of Implementation**: Linear regression is computationally efficient and easy to implement, making it suitable for a baseline regression model.
- **Feature Importance**: The coefficients obtained from linear regression offer insights into the importance and direction of each feature's impact on flight delays.

- **Assumption Suitability**: Linear regression assumes a linear relationship between variables. Given the nature of the problem, where features may have a linear impact on flight delays, this assumption is reasonable.
- **Performance on Numeric Data**: Linear regression performs well when predicting continuous variables, making it suitable for predicting flight arrival delays.

5. Model validation

5.1 Description of the Validation Process

The validation process in the app.py involves the following steps:

- **Train-Test Split**: The data is split into training and testing sets using the randomSplit method with a ratio of 70% for training and 30% for testing. This is a frequent practice to assess the model's performance on unseen data.
- **Feature Vectorization and Normalization**: The selected features are vectorized into a single feature column using VectorAssembler. The feature vectors are then normalized using the Normalizer. Normalization is employed to ensure that each feature contributes proportionally to the model. L1 normalization is chosen, which is suitable for linear regression models and prevents large-magnitude features from dominating the model.
- **Model Evaluation Metrics**: MAE, MSE, RMSE, and R2 are chosen as evaluation metrics to provide a comprehensive understanding of the model's performance. These metrics offer insights into several aspects of prediction accuracy and quality of fit.
- **Linear Regression Model Fitting**: A linear regression model is created using the LinearRegression class from PySpark's MLlib. The model is trained on the normalized feature vectors with the label set to the 'arrival_delay' column.
- **Regularization Parameters**: The regularization parameters (regParam and elasticNetParam) are set to 0.3 and 0.8, respectively. These parameters control the degree of regularization applied to prevent overfitting. The specific values are chosen based on experimentation and tuning for the given problem.
- **Pipeline Usage**: The use of a Pipeline ensures that the entire process, from data preprocessing to model training, is encapsulated and can be easily reproduced. This improves code readability and maintainability.

5.2 Final Evaluation of the Prediction Model

The model's performance is evaluated on the test set using multiple metrics:

- **Mean Absolute Error (MAE):** Measures the average absolute errors between predicted and actual values. A lower MAE value indicates better predictive performance.
- **Mean Squared Error (MSE):** Measures the average squared differences between predicted and actual values. A lower MSE value indicates better predictive performance.
- **Root Mean Squared Error (RMSE):** Represents the square root of the MSE, providing a measure of the model's accuracy. A lower RMSE value indicates better predictive performance.
- **R-squared (R2):** Indicates the proportion of the variance in the target variable that is predictable from the independent variables. A higher R2 value closer to 1 suggests a better fit of the model to the data.

Features	Coefficients
dest_encoded	-9373.008164
flight_number	-5888.884106
distance	2948.021220
day_of_week	-1039.862825
month	-212.203436
airline_encoded	71.707069
scheduled_arrival_time_mins	2.796172
year	-2.582991
actual_departure_time_mins	2.290157
day_of_month	0.000000
scheduled_departure_time_mins	0.000000
scheduled_flight_time_mins	0.000000
departure_delay	0.000000

Table 4: List of variables features used and their coefficients in the Linear Regression model.

The linear regression model exhibits a set of coefficients that reveal the impact of each feature on flight delays. Some features, like "distance" show a positive coefficient, indicating that an increase in it, is associated with an increase in the delays. In other words, it seems that the farther away the destiny, the greater the delay. In the other hand, variables like "day_of_week" or "month" exhibit negative coefficients, suggesting that higher values in these variables are linked to decreases in delays. In other words, weekdays are more probable to have delays than weekend days and later months are also more probable than early months.

Also, it appears to be that some variables, in the context of our model, may not have a statistically significant impact, or they might be collinear with other features. This, however, not necessarily means that they are useless.

The **intercept of 2.903** signifies the estimated delay when all features are set to zero. The positive intercept suggests a baseline delay, even in the absence of specific features.

In terms of evaluation metrics, the **Mean Absolute Error (MAE) of 8.15** indicates the average absolute difference between predicted and actual delays. **The Mean Squared Error (MSE) of 188.427** measures the average squared difference, emphasizing larger errors. **The Root Mean Squared Error (RMSE) of 13.726** provides an interpretable scale for the errors. Lastly, the **R-squared (R2) value of 0.65** indicates that approximately 65% of the variance in flight delays is captured by the model.

Overall, the model demonstrates a reasonable level of predictive accuracy, considering the complexities of flight delay prediction. However, there is room for improvement, and further refinement or exploration of alternative models may enhance predictive performance.

6. Instructions

6.1 Project Structure

These instructions assume that the script is saved with the original named delivered "project.py". Also, it is preferable to have a "data" folder in the same directory as the .py file. The structure recommended is:

```
/app
  |- project.py
  |- requirements.txt
  |- data/
     |- file1.bz2
     |- ...
```

To have that structure, simply extract the folder "*app*" from the ZIP folder "*SparkPracticalWork_group25*" delivered. Then place your "*data*" folder in it.

6.2 Command-line Argument – Library "Click"

However, the data could be placed wherever the user decides since a command-line argument can be given with the data path. For that, "click" library should be installed (specified in *requirements.txt*):

```
pip install click
```

6.3 Compilation and Execution:

Finally, to compile the Spark Application, utilize the following command in the terminal, assuming the structure presented above:

```
python3 project.py
```

or adding an alternative data_path

```
python3 project.py -d <your_input_data_path>
```

To execute the application with **spark-submit**:

```
spark-submit \
     --master local[*] \
     --deploy-mode client \
     --driver-memory 4g \
     --executor-memory 4g \
     <your_path>/app/project.py
```

7. Conclusion

In conclusion, our Spark practical application for flight delay prediction using PySpark has provided valuable knowledge into the understanding of the aviation industry's flight delays. With the PySpark API, a substantial dataset, "Data Expo 2009: Airline on-time data", was successfully processed. It was decided to only select the years 1988, 1998, and 2008 due to computational constraints.

A meticulous variable selection, transformation, and feature engineering process was done to identify key predictors that affected the flight delays, such as temporal factors, airline details, departure delays, and geographical aspects. Further on, a machine learning model was created using linear regression. It was chosen for its interpretability, providing a straightforward understanding of the relationship between features and flight delays. The model evaluation metrics, including mean absolute error, mean squared error, root mean squared error and r-squared demonstrated a moderate level of predictive accuracy, capturing approximately 65% of the variance in flight delays.

This project contributes not only to the optimization of airline flight delays but also lays the groundwork for future research. The integration of big data techniques and distributed processing capabilities of PySpark opens avenues for tackling broader challenges in the aviation domain like exploring alternative machine learning algorithms which could enhance the predictive capabilities, offering insights beyond linear regression and potentially improving the accuracy of flight delay predictions.