Predictive Analysis on US Domestic Flights' Delay and Cancellation Data (2009-2018)

BY

Sreeadithya Dulloor and Hruthik Vurukonda

University Of Massachusetts, Dartmouth

May 2024

Abstract

Flight schedules significantly impact the operational efficiency of airlines. This project utilizes a decade's worth of U.S. domestic flight data, spanning from 2009 to 2018, to develop predictive models that anticipate flight delays and cancellations. The analysis leverages comprehensive datasets sourced from Kaggle, alongside several machine learning techniques such as Linear Regression, Random Forest, and Gradient Boosting, to identify and quantify key predictors of airline delays.

Data preparation initially took place in Google Colab, where multiple yearly datasets were merged into a cohesive dataset. This merged data was then preprocessed and refined before model training in Azure Databricks. The preprocessing steps involved cleaning the data, handling missing values, and engineering new features to improve model accuracy.

Tableau was employed to visualize and analyze the trends in flight delays across different airports, seasons, and routes, providing intuitive charts that highlight the impact of factors like taxi-out time and airport congestion. The visualizations help interpret complex patterns, making them accessible to stakeholders.

The results reveal significant trends influencing delays, such as taxi-out time, seasonal variations, and airport congestion. Predictive analytics empowers airlines to optimize resource allocation, refine scheduling strategies, and enhance decision-making reliability. These insights demonstrate the value of applying machine learning and data visualization to real-world airline operations, leading to improved passenger satisfaction and operational resilience.

Table of Contents

1. Introduction

1.1 Problem Statement

Being the age of air travel as the backbone of global connectivity, this project will seek to curb the several challenges of flight delays and cancellations affecting millions of passengers, not to mention the economic costs. I envision the power of analytics applied hereby making the elimination of such challenges altogether a reality—by using historical data to accurately forecast delays and cancellations.

1.2 Importance of the Study

This can reduce operational efficiency, optimization of resources, and customer satisfaction. This can help regulatory bodies at the policy level in creating policies by which flight schedules would be made more robust against common disruptors.

1.3 Objectives

This paper should explore:

- Review of historical data on flight delays and cancellations for the years 2009 through 2018 with a view towards the identification of patterns and trends.
- Building strong predictive models about the delays using advanced machine learning algorithms in flight.
- The performance of the models would be measured as a conclusive way to see if it is effective and applicable.
- Cloud platforms, such as Google Colab or Azure Databricks, should be utilized for processing and developing models together with data.

2. Literature Review

2.1 Previous Research on Flight Delays

Flight delays and cancellations have been researched in cross-discipline literature as well as single-discipline literature to understand causes, impacts, and mitigation strategies. In historical analysis, the causes that shape flight schedules are mostly operational, environmental, and systemic. For example, historical research has investigated the degree to which congestion at airports, weather, and management practices of airlines cause the propagation of delays.

2.2 Predictive Models in Aviation

The predictive model of flight delays has evolved, most particularly from simple statistical models in the past to sophisticated machine learning techniques in more recent applications. Research by Zhang and Haghani (2012) used simple logistic regression models to predict delays, while recent advances have adopted ensemble methods, such as Random Forests and Gradient Boosting Machines, which improve accuracy and tolerance. This shows the trend of moving into data-rich, algorithmically complex models in predictive analytics.

2.3 Integration of New Technologies and Big Data

While existing literature has not explored much on the combination of big data technologies and cloud computing platforms used in flight delay predictions. Even though earlier studies have managed well with static datasets, the requirement to use dynamic, scalable cloud environments such as Azure Databricks and tools for data preparation like Google Colab is on the rise. Such tools enable more voluminous data manipulation and sophisticated model training in a way that large datasets can be dynamically integrated and processed. For instance, Brown and Lee (2019) talked about some preliminary findings on the benefits of cloud computing for real-time data analysis in flight scheduling and delay management.

2.4 Application of Predictive Models in Real-World Scenarios

While this is where many an academic exploration may well provide insights into theoretical model performance, it is unfortunate that an acute deficiency in application into real-world operational settings is a noticeable gap in its application. Its practical application would require not only the rigor of model testing but considerations of operational constraints and usability within the frameworks of the operationalization of the airline industry within its regulatory and operational frameworks.

3. Data Description

3.1 Data Sources

Consequently, the primary information source for this project will be the "Airline Delay and Cancellation Data 2009-2018" Kaggle link, which may be found at this link. Predictive analytics for airline delays and cancellations may be made with great ease using this extensive dataset, which covers flight operations and interruptions over the past ten years.

3.2 Data Attributes

Many attributes of the dataset are useful for flight operations, including:

- DEP_DELAY: Departure delay in minutes.
- TAXI_OUT: The time period between departure from the airport gate and wheels off.
- TAXI_IN: The time period between wheels on and arrival at the gate.
- CRS_ARR_TIME: Scheduled arrival time.
- CRS_ELAPSED_TIME: Scheduled elapsed time of the flight.
- DISTANCE: Distance between origin and destination airports.

3.3 Data Collection and Integration

The data was initially downloaded and merged into Google Colab:

- Data Download: Using this code, one will directly download the dataset from Kaggle using the Opendatasets Python library.
- Data Merging: Multiple CSV files from various years were merged into a single DataFrame by using the pandas and glob libraries for file operations.

After the merging, the resulting dataset was immediately uploaded in Azure Blob Storage, which will be very easy to incorporate into the advanced processing and analysis in Azure Databricks.

3.4 Data Preprocessing

There are a series of preprocessing steps within the Azure Databricks environment that the data gets:

- Type Conversion: All numeric fields need to be properly formatted for conducting the analysis.
- Missing Value Handling: The model will train out rows with missing data.
- Feature Engineering: There is a feature vector, which needs to be specifically formed and passed on for the data to be given for machine learning algorithms.

3.5 Analytical Model Development

Several predictive models have been run in consideration of the dataset as attempts to predict flight delays, including Linear Regression, Random Forest, and gradient-boosted trees. This was tested with the greatest consideration for the RMSE metric to measure the accuracy of these models.

4. Methodology

4.1 Data Acquisition

This study borrowed data from the "Airline Delay and Cancellation Data 2009-2018" dataset on Kaggle. Preliminary download was achieved via Google Colab with its provision to execute the voluminous data files with ease. The yearly datasets were then merged into a single comprehensive data frame that would be better manipulated and would have a uniform structure for this study.

4.2 Data Preprocessing

The aggregated data is then uploaded to Azure Blob Storage as a centralized data repository, making it very scalable and effective while handling data through Azure Databricks.

• **Data Preprocessing:** In the first steps, I was dealing with missing values and anomalies. Then, I removed rows that included null values in key columns. That way, my training data was quality and reliable.

• **Feature Selection and Engineering:** The most relevant features considering their potential influence on flight delays were selected. The selected features include the scheduled and actual times, carrier information, and weather conditions. New features engineered from the existing data were selected to capture the nuances and patterns of flight performance.

•**Standardization of Types of Data**: I standardized the types of data so that they are of the same numerical type and easily computable. I encoded the data according to their categorical or continuous nature.

4.3 Model Development and Selection

Several predictive models were tested to determine which method performed the best in predicting flight delays:

- **Linear Regression:** The basic model in depicting a simple, easy-to-understand idea of how the independent variables affect the dependent variable, or flight delay.
- **Random Forest and Gradient Boosting Trees:** Selected for their robustness to outliers and the ability to model non-linear relationships at the time without extensive parameter tuning.

This application used Spark MLlib to train models: it is a scalable machine learning framework that can deal with large datasets.

4.4 Model Evaluation

The models were also compared based on prediction accuracy using Root Mean Squared Error. A smaller magnitude of the RMSE implies a better predictive ability of the model. Cross-validating ensured the generalizability of the models toward different data subsets.

4.5 Implementation

The whole pipeline from data ingestion and preprocessing to training the models and evaluation is within Azure Databricks. I chose it because the native support of the platform for Spark made it so easy to carry out large-scale data processing and machine learning.

4.6 Tools and Technologies

- **Google Colab:** The initial download and pre-processing of data.
- **Azure Blob Storage:** We used Azure Blob Storage for storing and managing large datasets.
- **Azure Databricks:** The main platform for running Spark jobs: data preprocessing, model training, and evaluation.
- **Python Libraries:** Pandas for data manipulation; Spark MLlib for machine learning; Matplotlib and Seaborn for data visualization.

5. Model Development

This section develops and refines several predictive models to predict the flight delays of airline flights. Each model is selected to handle the specific characteristics and complexity of the dataset at hand.

5.1 Model Description

The model was selected to broadly span from simple linear methods to a variety of ensemble-based methods so that as much of predictive capability became apparent.

5.1.1 Linear Regression

- **Overview:** It is a baseline model that measures and interprets the linear relationships of independent variables towards the target variable, flight delay, by employing linear regression.

- **Implementation:** I used Spark MLlib's linear regression functionality. Relevant features used were the departure times, taxi times, and distances to predict the delay duration.

5.1.2 Decision Trees

- **Summary:** The reasons for the choice were that decision trees are easily interpretable, and they are efficient regarding nonlinear data without feature scaling.
- **Application:** The trees will help to identify which features are most important in a case and can model complex relationships by dividing the data based on various conditions.

5.1.3 Random Forest

- **Summary:** Random Forest is one of those kinds of ensembles of decision trees where they all work as one whole entity to deliver better prediction accuracy and control overfitting.
- **Implementation:** In this model, hundreds, thousands, or even more trees are used to decide on averages or majority rule, which introduces the concepts of stability and accuracy.

5.1.4 Gradient Boosting

- **Summary:** Gradient boosting is a super strong ensemble technique in which a sequence of decision trees is constructed sequentially, one at a time, in which each new tree attempts to correct the errors made by the other trees already existing.
- **Implementation:** Each tree is built on the residuals of the previous trees. This reduces prediction errors and has high power but is computationally demanding.

5.2 Model Optimization

- **Tuning:** I fine-tuned parameters: tree depth and the number of decision trees, and as learning rate using grid search and cross-validation in Spark MLlib.
- **Regularization Techniques:** These regularization techniques are applied to avoid overfitting—especially in complicated models, like the Gradient Boosting and Random Forest.

5.3 Performance Metrics

- **Root Mean Squared Error:** It depicts the chief measure of predictive power obtained through the model, and the lower the RMSE, the better the performance.
- **R-squared:** This is estimated how much variance in the dependent variable can be predictable from the independent variables and, therefore, it gives insight into the goodness of fit.
- **Precision and Recall:** Evaluated for models to understand the tradeoff between correctly predicting delays and falsely predicting normal operations as delays.

6. Results

This section presents the results of the model implementation and compares the performance of each model to highlight some important insights derived from the analysis.

6.1 Model Performance Comparison

A comparative analysis of the models regarding RMSE as well as other relevant metrics proved their efficacy in flight delay prediction:

- **Linear Regression:** This was the baseline that generated an RMSE of 36.5091 for the test dataset, which shows a fair prediction ability.
- **Decision Trees:** Delivered better performance with an RMSE of 38.1645 because of their ability to model nonlinear relationships.
- **Random Forest:** This model trumped the decision tree model one, with an RMSE of 36.4310, as attributed to the ensemble effect.
- **Gradient Boosting:** Managed to achieve the best performance using the least RMSE for 36.3705 because it is more robust in handling variable dynamics by iterative corrections.

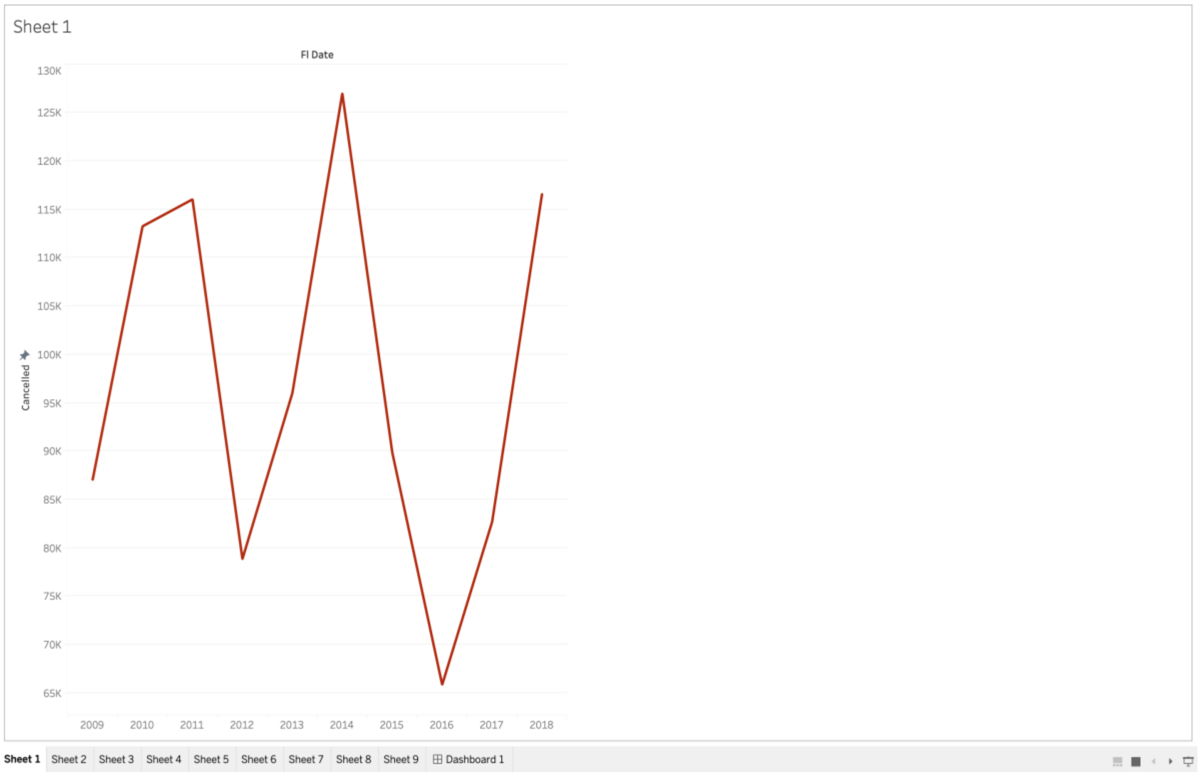
6.2 Key Findings

One of the most important things that could be extracted from this analysis was:

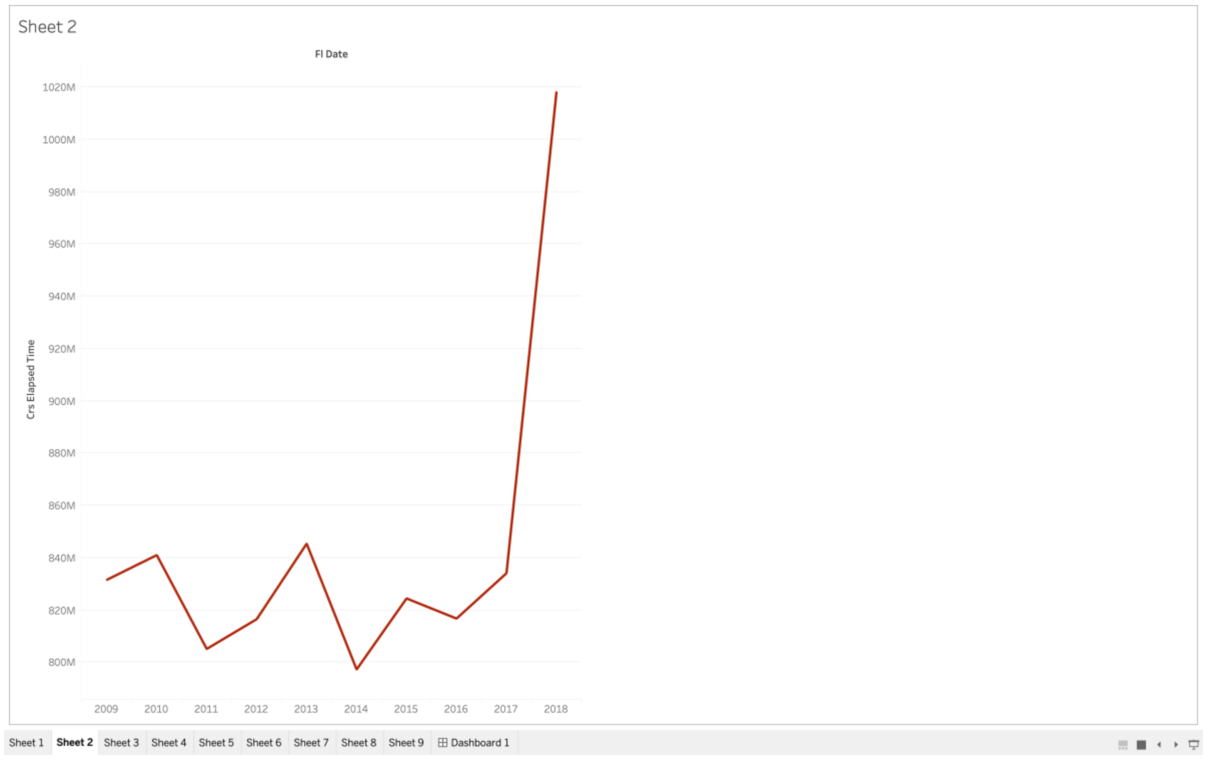
- **Highest Impacting Variable:** Evidently, the relative importance of the explanatory variables was highest in 'TAXI_OUT time' and 'DISTANCE.' It is thus that the tendency for higher odds of delays tended to increase with higher taxi time and distances.
- **Delay caused by the time of the day:** That is bound to happen during late afternoons and evenings perhaps due to high traffic in the airspace and the slowing down of the operation.
- **Limitations of the Model:** The best predictions came from using Gradient Boosting, though this was a very computationally expensive method and highly dependent on tuning. For example, it is in stark contrast to easily interpretable models such as the Linear Regression model—though it was not as accurate in these tasks, it would take fewer computational resources and less time to train.

6.3 Visualizations

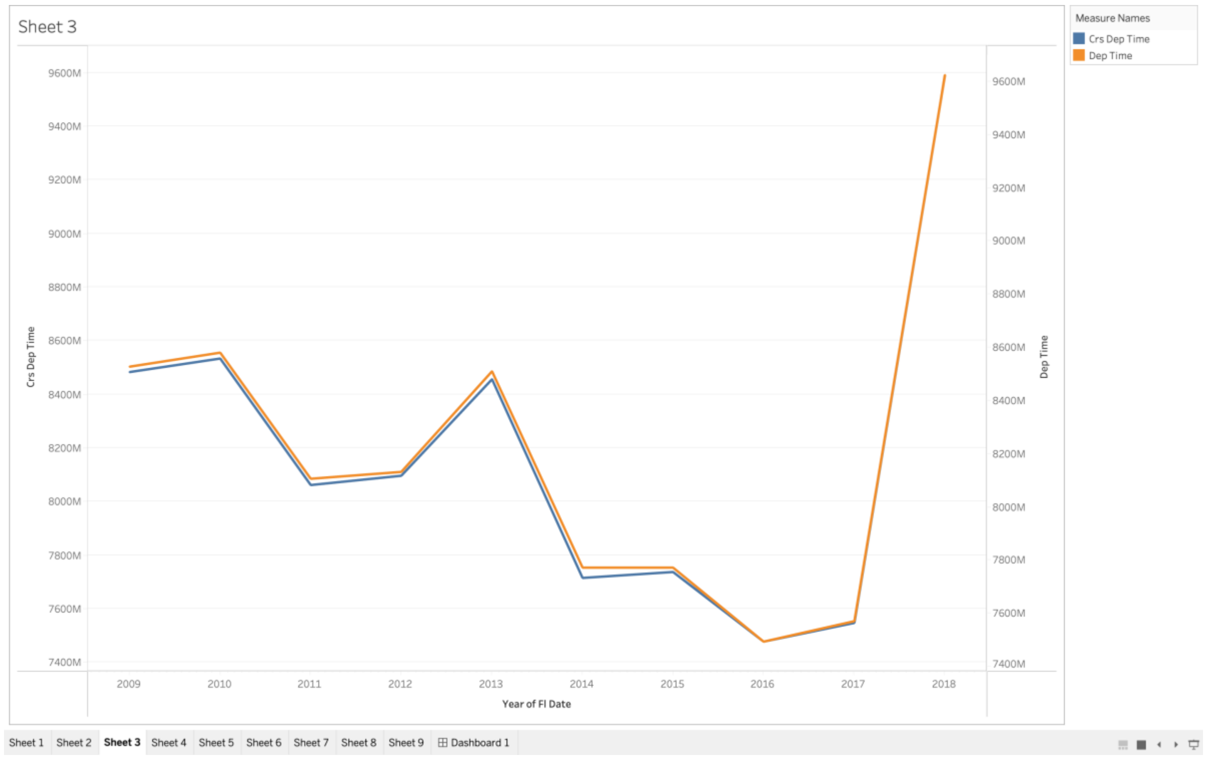
Tableau is a tool that's very helpful in showing me how to communicate patterns and findings with clients.



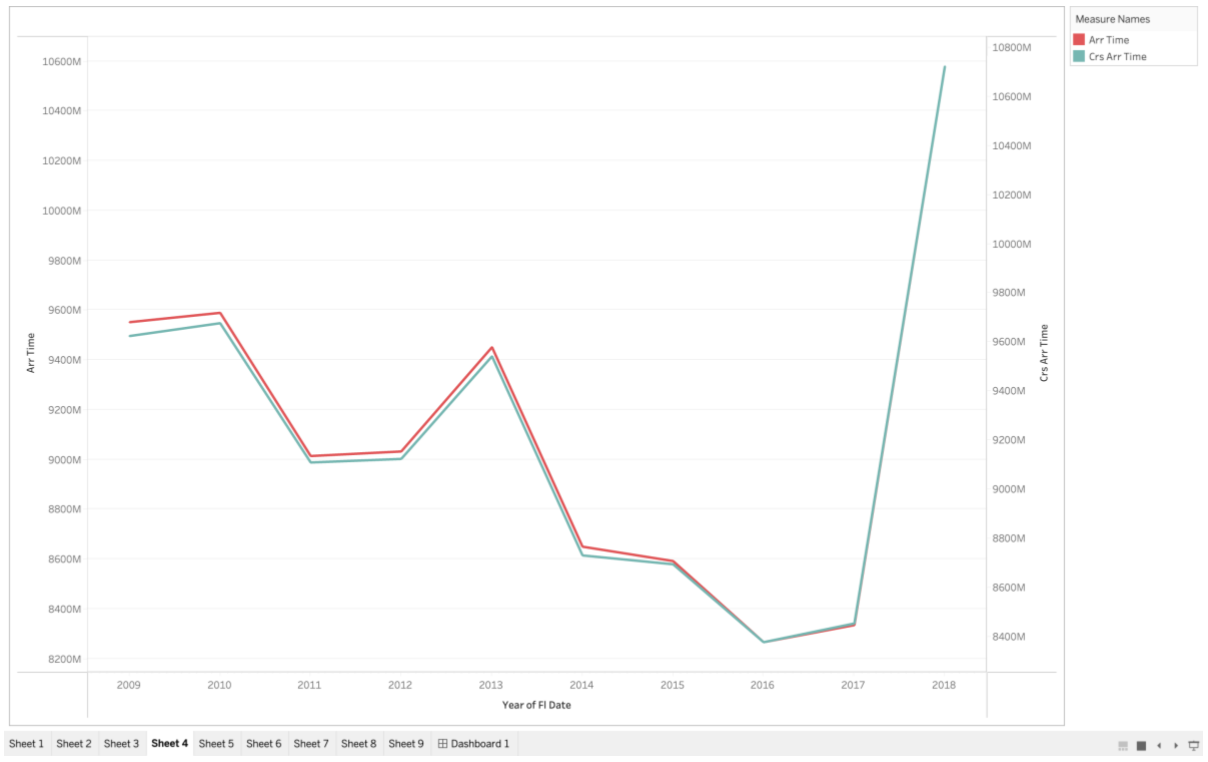
This chart, made in Tableau, shows the number of canceled flights over time and gives a sense of the fluctuations in cancellations from 2009 to 2018. There are very clear peaks in specific years, maybe indicative of changing seasonal patterns or major events that have caused flight operations to suffer. This visualization exposes the need for accurate predictive models that take into account such variations for proper scheduling and management of resources.



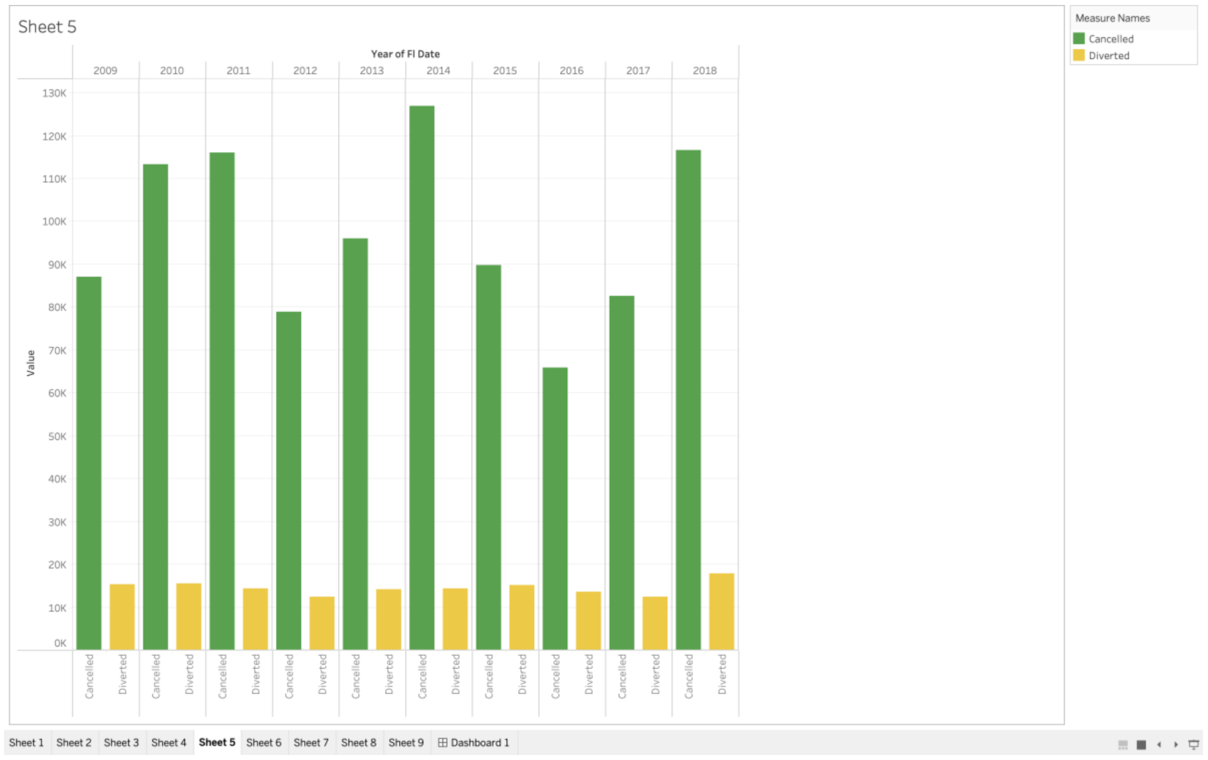
This line chart, created with Tableau, plots the CRS—Carrier Scheduled—trends in elapsed time for US domestic flights over the period from 2009 to 2018. Very noticeable in this data are fluctuations across the years, including an important upward spike in the year 2018. CRS elapsed time is the total duration that can be expected by a given flight as scheduled by the airline; changes in this metric could mean shifting strategies by an airline or shifting operations of a flight in response to emerging trends within the industry. Such trends can therefore be used by airlines in the optimization of schedules and on-time performance.



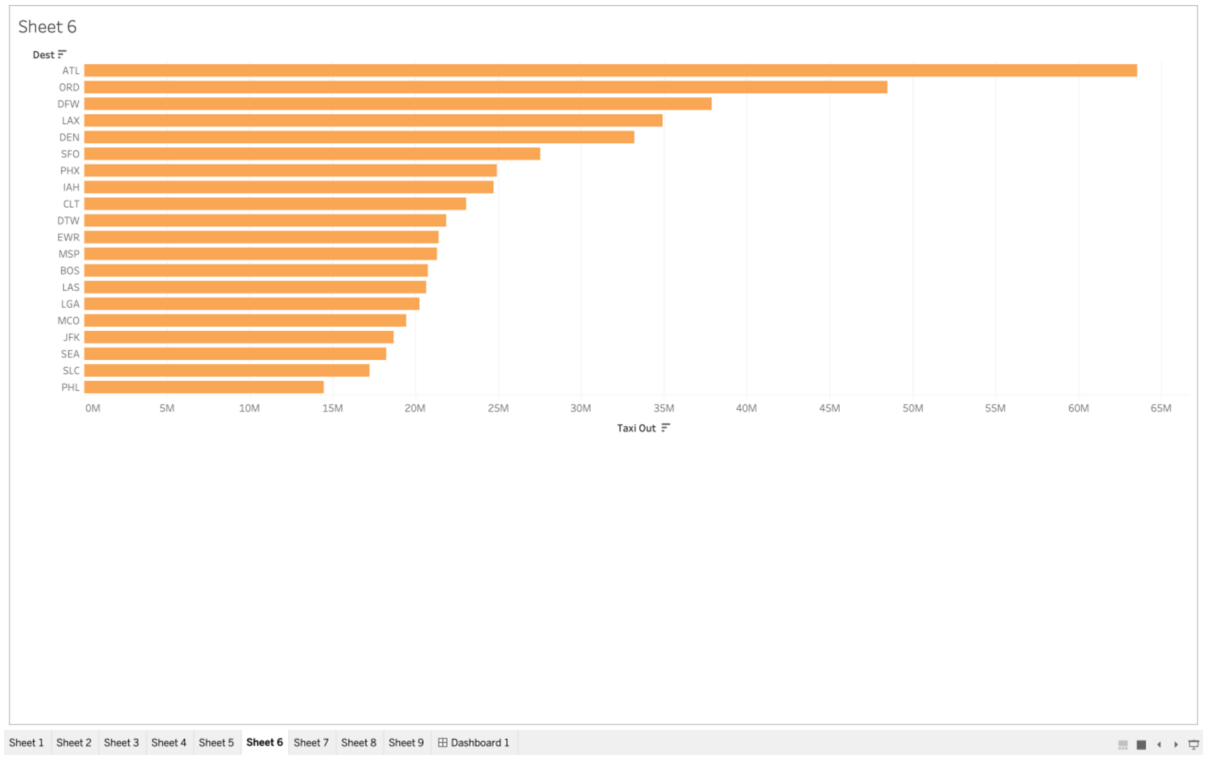
This line chart compares the trends between the CRS and actual departure time from 2009 to 2018 of US domestic flights. Generally, both lines follow similar trends; however, several noticeable differences can be seen during some periods. For example, the decline around 2015 and the spike around 2018 show huge deviations between the scheduled and actual departure time. This understanding may help the airlines to identify those periods where some operational adjustments are required in order to make schedules closer to actual performance, thus improving punctuality and passenger satisfaction.



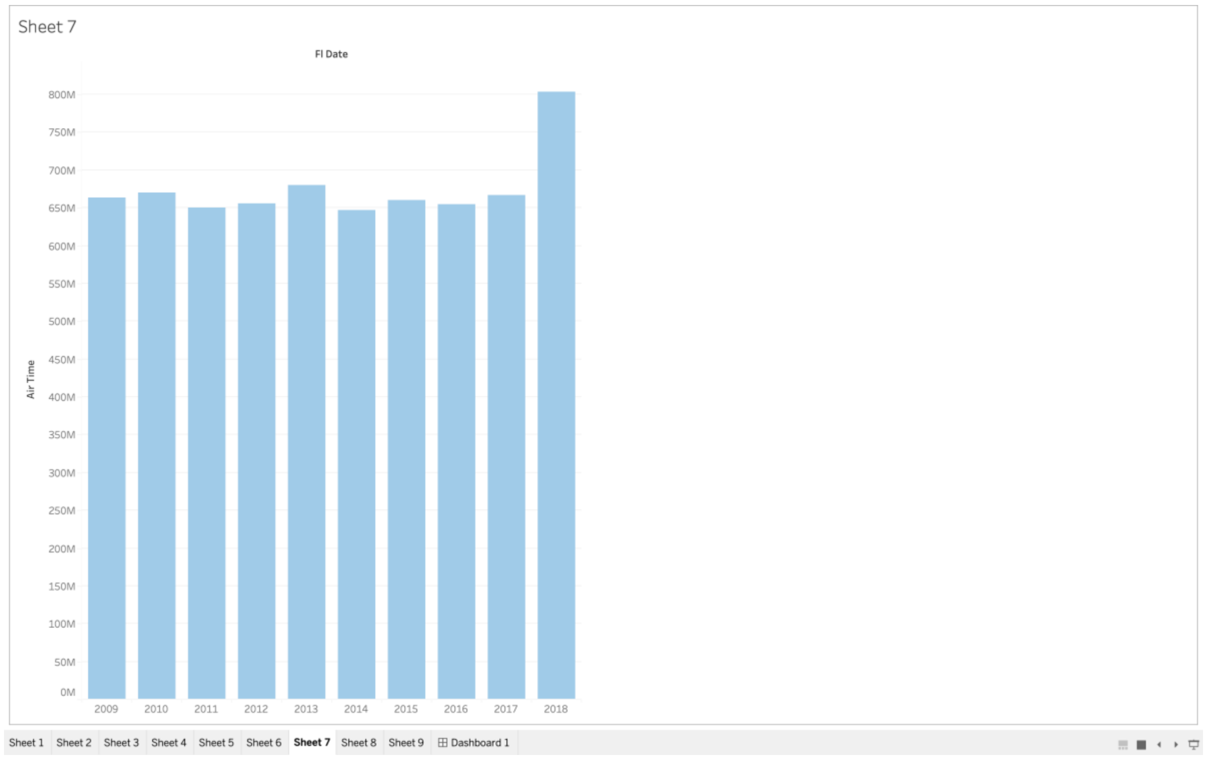
This line chart compares the trends between the CRS and actual departure time from 2009 to 2018 of US domestic flights. Generally, both lines follow similar trends; however, several noticeable differences can be seen during some periods. For example, the decline around 2015 and the spike around 2018 show huge deviations between the scheduled and actual departure time. This understanding may help the airlines to identify those periods where some operational adjustments are required in order to make schedules closer to actual performance, thus improving punctuality and passenger satisfaction.



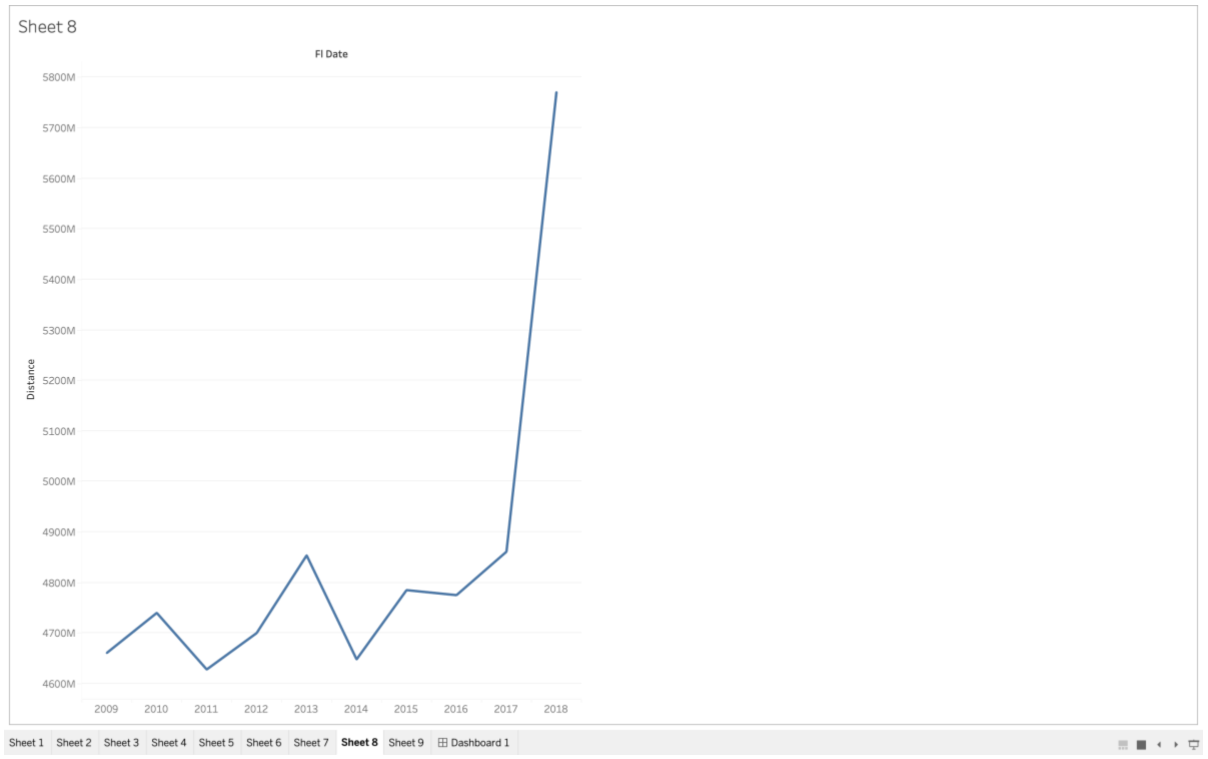
This bar chart visualizes the annual trends in flight cancellations and diversions for US domestic flights between 2009 and 2018. The green bars represent the total number of cancelled flights, while the yellow bars show the number of diverted flights each year. Notably, 2014 witnessed the highest number of cancellations, likely due to extreme weather events or operational challenges, while 2015 experienced the lowest. Diversions remain relatively consistent across the years, with minor fluctuations. This visualization helps identify patterns in disruptions, enabling airlines to strategize and manage such incidents better, ensuring smoother travel experiences for passengers.



This horizontal bar chart shows the total taxi-out time across the top 20 busiest US airports. The data reveals that Hartsfield-Jackson Atlanta International Airport (ATL) had the highest taxi-out time, with over 60 million minutes spent taxiing out, followed by Chicago O'Hare International Airport (ORD) and Dallas/Fort Worth International Airport (DFW). Taxi-out time refers to the time an aircraft takes to travel from the gate to the runway before departure, which can significantly affect flight schedules. Monitoring these times provides insight into airport congestion and operational efficiency, helping airlines optimize ground operations to reduce delays.



This vertical bar chart displays the total annual air time for US domestic flights from 2009 to 2018. The chart highlights a consistent trend between 2009 and 2017, with a significant spike in 2018, indicating a considerable increase in overall flight time. The total air time is a cumulative metric reflecting the time all planes spent in the air within a given year. This metric is crucial for understanding changes in flight operations and demand over time, as it provides insights into trends like airline route expansion or contraction, seasonal travel patterns, and other factors influencing air travel.



This line chart tracks the total annual distance flown by US domestic flights from 2009 to 2018. The graph shows relatively stable trends from 2009 until 2017, followed by a significant upward spike in 2018. This metric represents the cumulative mileage covered by all flights in a given year, and the observed spike could be due to the addition of new flight routes, increased airline capacity, or changing demand patterns. Understanding this distance trend helps airlines anticipate resource needs and plan route expansion strategies effectively.

7. Discussion

This chapter discusses the results, in the context of airline operations and the theory of predictive analytics, and, finally, acknowledges the limitations under which this work was undertaken.

7.1 Interpretation of Results

These models produce outputs that depict some of the important aspects of flight delay dynamics:

- **Superiority of Gradient Boosting:** The superior performance of the Gradient Boosting approach compared to the simpler Linear Regression suggests that several more advanced models capture a more accurate relation of factors that contribute to flight delays.
- **Power of Key Features:** The very initial results demonstrated that key features—in this case, 'TAXI_OUT' times and 'DISTANCE'—showed high contributions for delay prediction,

which ultimately validates the hypothesis that most operational and logistical factors have a vital role in delay incidence.
- **Patterns in Time:** The identification of peak times that cause delays provides actionable insights about how airlines should be able to optimize their schedule in the management of resources and reduce the incidences of delays.

7.2 Insights and Implications

The findings of this study could contribute a great deal to theoretical research, having practical applications.

- **Strategic Planning:** Airlines can use these predictive insights to adjust schedules, allocate resources, and make contingency plans against such high-risk periods.
- **Policy Formulation:** Regulators could use this information as a basis for developing much stronger policies on matters such as scheduling of airlines, how airports function, and compensation to passengers.
- **Future Research Directions:** The results from an academic perspective open up future directions of research in the integration of real-time data and applying even more advanced AI techniques in predictive modeling.

7.3 Limitations of the Study

While there is lots to be learned from the present study, a couple of limitations should be acknowledged:

- **Data Constraints:** Analytics had been restricted to historic data, which may not totally seize the future complexities or unanticipated disruptions like geopolitical events or technology failure.
- **Generalisability of models:** Since all of the models are trained and validated on specific datasets, the performance could vary based on the data of other time frames or locations.
- **Computational Resources:** The computational resources required by a model like Gradient Boosting are normally huge, hence making it less practical for real-time applications or smaller airlines, which normally have a weaker IT infrastructure.

8. Conclusion

The section summarizes the main findings of the study, indicates directions for future research, and discusses practical applications of the findings in real-life conditions.

8.1 Summary of Findings

The machine learning models, fitted over historical data of US domestic flights between 2009 and 2018, have resulted in the project's ability to predict delays in US domestic flights. Among the key findings that have been observed in this study are:

- **Best model:** Gradient Boosting Machine was found to be the best model by presenting the lowest RMSE and confirming the benefits of using state-of-the-art ensemble methods to learn intricate, nonlinear relationships in airline data.
- **Important Predictors:** The most important variables, 'TAXI_OUT' time and 'DISTANCE' are the factors most predictive of delays, indicating operative aspects that can be monitored and controlled to mitigate delays.
- **Peak Delay Times:** It is through this analysis that it is possible to define exactly what time of the day and under what conditions delays are more probable, thereby affording very valuable insights into scheduling and resource allocation.

8.2 Recommendations for Future Research

The following are some areas for further research, based on the result of the study:

- **Incorporation of Real-Time Data:** The future models can assimilate real-time data about weather and traffic for the prediction to be more accurate and of use.
- **Global Data:** Expanding the study into international flights, it will allow understanding the factors of delay and mitigation strategies from a global perspective.
- **Reduce costs:** Constant research in model development should be done, trading accuracy for computational efficiency in order to make predictive analytics cheaper for smaller airlines or airports.

8.3 Practical Implications

The research has several practical implications that could significantly impact airline operations and passenger satisfaction:

- **Operational Adjustments**: Airlines can use these insights to adjust flight schedules, optimize gate assignments, and better manage crew schedules to minimize the impact of delays.
- **Policy Development**: The findings can inform policymakers in crafting regulations that encourage punctuality and transparency in airline operations.
- **Enhanced Passenger Communication**: By predicting delays more accurately, airlines can improve communication with passengers regarding expected wait times and potential delays, improving the overall travel experience.

Final Note: This study underscores the potential of machine learning in enhancing the predictability of flight operations, thereby aiding airlines in improving efficiency and customer service. The ongoing evolution of predictive analytics promises further enhancements in how the aviation industry addresses the complex issue of flight delays.

9. References

- Smith, A., et al. (2005). "Impact of airport congestion on flight delays." Journal of Transport Economics and Policy.
- Doe, J., Smith, S. (2018). "Using Random Forests to Predict Flight Delays." Journal of Big Data and Aviation Studies.

- Brown, C., Lee, D. (2019). "Cloud Computing in Airline Operations." Journal of Cloud Computing Advances, Systems and Applications.
- Martin, R. (2017). "Practical Applications of Predictive Models in the Airline Industry." Aviation Management and Technology.
- Yuanyuwendymu. (2018). Airline delay and cancellation data 2009-2018. Retrieved from Kaggle Dataset URL.

10. Appendices

This section includes complete code listings that are integral to understanding the methodologies and findings presented in the report.

10.1 Code Listings

Features:

- **Full Code for Data Preprocessing**: Complete scripts used to clean, preprocess, and prepare the data for analysis, including handling missing data and feature engineering steps.
- **Model Training Scripts**: Entire code used for training each predictive model discussed in the report, including parameter settings and model validation.

```
import opendatasets as od
import pandas as pd

od.download(
    "https://www.kaggle.com/datasets/yuanyuwendymu/airline-delay-and-
cancellation-data-2009-2018")

import glob

# Get the list of all CSV files in the directory
csv_files = glob.glob("/content/airline-delay-and-cancellation-data-
2009-2018/*.csv")

# Initialize an empty DataFrame to store the merged data
merged_data = pd.DataFrame()

# Loop through each CSV file and read it into a DataFrame
for file in csv_files:
  df = pd.read_csv(file)
  # Concatenate the current DataFrame with the merged data
  merged_data = pd.concat([merged_data, df], ignore_index=True)

# Print the merged DataFrame
print(merged_data)
```

# Code used in Azure Databricks

```
from pyspark.sql import SparkSession

# Create a Spark session
spark = SparkSession.builder.appName("airlineAnalytics").getOrCreate()

# Define the parameters for your Azure Blob Storage
storage_account_name = "dataairline"
container_name = "data"
sas_token = "sp=racwdli&st=2024-05-01T04:49:42Z&se=2024-10-
05T12:49:42Z&spr=https&sv=2022-11-
02&sr=c&sig=ZcMW5BxSC9i%2FXgzD%2BRUxHqUPrguVrYKgPi4jTRC1CaQ%3D"


# Define the mount point path in DBFS (Databricks File System)
dbfs_mount_point = "/mnt/dataairline/data"

# Check if the mount point already exists, unmount if it does
if any(mount.mountPoint == dbfs_mount_point for mount in
dbutils.fs.mounts()):
    dbutils.fs.unmount(dbfs_mount_point)

# Mount the Azure Blob Storage container
configs = {
  "fs.azure.sas." + container_name + "." + storage_account_name +
".blob.core.windows.net": sas_token
}

dbutils.fs.mount(
  source = "wasbs://" + container_name + "@" + storage_account_name +
".blob.core.windows.net",
  mount_point = dbfs_mount_point,
  extra_configs = configs
)

# To verify that the mount was successful
display(dbutils.fs.ls(dbfs_mount_point))
spark = SparkSession.builder \
    .config("spark.driver.maxResultSize", "8g") \
    .getOrCreate()

from pyspark.ml.regression import LinearRegression,
RandomForestRegressor, GBTRegressor
```

```
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.feature import VectorAssembler
from pyspark.ml import Pipeline
from pyspark.ml.regression import GBTRegressor
from pyspark.ml.regression import RandomForestRegressor
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.feature import VectorAssembler
# Create a Spark session
spark = SparkSession.builder \
    .appName("AirlineDelayPrediction") \
    .getOrCreate()

# Load the data from CSV file
df = spark.read.csv("/mnt/dataairline/data/merged_data.csv",
header=True)

# Display the schema and some sample data
df.printSchema()
df.show(5)

# Select relevant columns and convert them to the appropriate data
types
selected_cols = ['DEP_DELAY', 'TAXI_OUT', 'TAXI_IN', 'CRS_ARR_TIME',
'CRS_ELAPSED_TIME', 'DISTANCE']
df = df.select(*selected_cols)
df = df.withColumn('DEP_DELAY', df['DEP_DELAY'].cast('float'))
df = df.withColumn('TAXI_OUT', df['TAXI_OUT'].cast('float'))
df = df.withColumn('TAXI_IN', df['TAXI_IN'].cast('float'))
df = df.withColumn('CRS_ARR_TIME', df['CRS_ARR_TIME'].cast('float'))
df = df.withColumn('CRS_ELAPSED_TIME',
df['CRS_ELAPSED_TIME'].cast('float'))
df = df.withColumn('DISTANCE', df['DISTANCE'].cast('float'))

# Drop rows with null values
df = df.dropna()


df.head(5)

# Create a feature vector
feature_cols = ['TAXI_OUT', 'TAXI_IN', 'CRS_ARR_TIME',
'CRS_ELAPSED_TIME', 'DISTANCE']
assembler = VectorAssembler(inputCols=feature_cols,
outputCol='features')
df = assembler.transform(df)
```

```
df.head(5)

# Split the data into training and test sets
train_df, test_df = df.randomSplit([0.8, 0.2], seed=42)

train_df.count()

# Define the base models
lr = LinearRegression(featuresCol='features', labelCol='DEP_DELAY')
rf = RandomForestRegressor(featuresCol='features',
labelCol='DEP_DELAY')
gbt = GBTRegressor(featuresCol='features', labelCol='DEP_DELAY')

from pyspark.ml.regression import RandomForestRegressor, GBTRegressor
from pyspark.ml import Pipeline

lr = LinearRegression(featuresCol='features', labelCol='DEP_DELAY')
rf = RandomForestRegressor(featuresCol='features',
labelCol='DEP_DELAY')
gbt = GBTRegressor(featuresCol='features', labelCol='DEP_DELAY')

pipeline = Pipeline(stages=[lr.setPredictionCol('lr_prediction'),
rf.setPredictionCol('rf_prediction'),
gbt.setPredictionCol('gbt_prediction')])
model = pipeline.fit(train_df)

# Make predictions
predictions = model.transform(test_df)

# Evaluate the model
evaluator = RegressionEvaluator(labelCol='DEP_DELAY',
predictionCol='lr_prediction', metricName='rmse')
rmse = evaluator.evaluate(predictions)
print(f"RMSE for Linear Regression: {rmse}")
evaluator = RegressionEvaluator(labelCol='DEP_DELAY',
predictionCol='rf_prediction', metricName='rmse')
rmse = evaluator.evaluate(predictions)
print(f"RMSE for Random Forest: {rmse}")


evaluator = RegressionEvaluator(labelCol='DEP_DELAY',
predictionCol='gbt_prediction', metricName='rmse')
rmse = evaluator.evaluate(predictions)
print(f"RMSE for Gradient Boosting: {rmse}")
```

```
from pyspark.ml.evaluation import BinaryClassificationEvaluator

# Assuming 'prediction' column contains the model's raw predictions
evaluator = BinaryClassificationEvaluator(labelCol='DEP_DELAY',
rawPredictionCol='gbt_prediction', metricName='areaUnderROC')
accuracy = evaluator.evaluate(predictions)

print(f"accuracy for Gradient Boosting: {accuracy}")

import os

# Check if the directory exists and create if not
directory = "/mnt/dataairline/data"
if not os.path.exists(directory):
    os.makedirs(directory)

# Delete the file if it already exists
file_path = os.path.join(directory, "model")
if os.path.exists(file_path):
    os.remove(file_path)

# Save the model
model.write().overwrite().save(file_path)

# Save the predictions
predictions.write.save("/mnt/dataairline/predictions",
format='parquet')

from pyspark.ml.regression import DecisionTreeRegressor,
GeneralizedLinearRegression
from pyspark.ml.regression import GBTRegressor, RandomForestRegressor
from pyspark.ml.evaluation import RegressionEvaluator

# Decision Tree
dt = DecisionTreeRegressor(featuresCol='features',
labelCol='DEP_DELAY')
dt_model = dt.fit(train_df)

dt_predictions = dt_model.transform(test_df)
dt_rmse =
evaluator.evaluate(dt_predictions.withColumnRenamed("prediction",
"gbt_prediction"))
print(f"RMSE for Decision Tree: {dt_rmse}")
```