Flight Delay Predictor

Author: Vamsidhar Venkataraman

Abstract

The objective of this project is to predict flight delays for 15 airports in the United States using a two-stage machine learning model. This model combines classification and regression techniques to provide comprehensive predictions. Initially, a binary classifier determines if a flight will be delayed by 15 minutes or more, using the target variable ***Arrdel15***. Subsequently, for flights classified as delayed, a regression model predicts the exact delay duration in minutes, denoted by ***ArrDelminutes***. The project involves extensive data acquisition and preprocessing of flight and weather information. The final ***classifier*** and ***regressor*** has an accuracy of ***0.92*** and ***0.94*** respectively . The table below lists the airports for which weather data is available:

ATL	CLT	DEN	DFW	EWR
IAH	JFK	LAS	LAX	MCO
MIA	ORD	PHX	SEA	SFO

Table 1: The airports for which weather data is available.

Contents

1 Introduction

Flight delays pose significant challenges in the aviation industry, causing widespread disruption to passengers, airlines, and airport operations. These delays not only inconvenience travelers but also result in substantial financial losses for airlines, necessitating complex logistical adjustments including resource reallocation and crew rescheduling. Additionally, airlines may face compensation claims from affected passengers, further impacting their bottom line.

To address this critical issue, this project proposes an innovative two-stage machine learning solution. The first stage employs a classification model to determine the likelihood of a flight delay. If a delay is predicted, the second stage utilizes a regression model to estimate the duration of the delay in minutes, providing a more precise forecast for operational planning.

The model is trained on a comprehensive dataset comprising historical flight information from 15 major U.S. airports, spanning the years 2016 to 2017. This flight data is augmented with corresponding weather information to enhance predictive accuracy. The project involves rigorous data preprocessing, feature engineering, and model selection processes. Various machine learning algorithms are evaluated for both the classification and regression stages, with the best-performing models selected to construct the final two-stage prediction system.

By leveraging advanced analytics and machine learning techniques, this project aims to provide a robust tool for airlines and airports to anticipate and mitigate the impact of flight delays, ultimately improving operational efficiency and passenger satisfaction in the dynamic aviation sector.

2 Dataset

The dataset for this project is a comprehensive collection of flight and weather data, meticulously curated to support the prediction of flight delays. It includes key flight-related columns such as ***FlightDate***, ***Quarter***, ***Year***, ***Month***, and ***DayofMonth***, which provide temporal context and help in capturing seasonal trends in delays. Critical logistical details like ***DepTime***, ***CRSDepTime***, ***ArrTime***, and ***CRSArrTime*** are included to compare actual and scheduled times, enabling the identification of delays. The dataset also features ***DepDelayMinutes*** and ***ArrDelayMinutes***, which record the delay durations, and ***ArrDel15***, a binary indicator of delays over 15 minutes, serving as targets for the classification and regression models, respectively. Unique identifiers such as ***OriginAirportID*** and ***DestAirportID*** link flights to their specific airports, allowing for airport-specific delay analysis.

The weather-related columns provide detailed meteorological data relevant to flight operations. Variables such as ***WindSpeedKmph***, ***WindDirDegree***, ***Visibility***, and ***WeatherCode*** offer insights into the weather conditions affecting flight schedules. These factors, including precipitation (***precipMM***), atmospheric pressure (***Pressure***),

FlightDate	Quarter	Year	Month
DayofMonth	DepTime	DepDel15	CRSDepTime
DepDelayMinutes	OriginAirportID	DestAirportID	ArrTime
CRSArrTime	ArrDel15 (label)	ArrDelayMinutes (target)	

Table 2: Flight-related columns in the dataset

and cloud cover (***Cloudcover***), are crucial for understanding the environmental impacts on flight delays. Temperature-related data (***tempF***, ***WindChillF***, ***DewPointF***), along with humidity levels, provide further context on the operational conditions faced by flights. This detailed weather information is precisely matched with flight data based on the ***date***, ***time***, and ***airport*** columns, ensuring an accurate alignment of weather and flight events.

WindSpeedKmph	WindDirDegree	WeatherCode	precipMM
Visibilty	Pressure	Cloudcover	DewPointF
WindGustKmph	tempF	WindChillF	Humidity
date	time	airport	

Table 3: Weather-related columns in the dataset

The dataset integrates these flight and weather variables to form a robust framework for the project. It includes specific airport codes like ATL, JFK, and LAX, which facilitate location-specific analysis and link flight records to their respective airports.

3 Classifier

The first stage of the model involves a classifier that functions as a judge, determining whether a flight will be delayed or not. The criteria are straightforward: any flight arriving more than 15 minutes behind schedule is labeled as "delayed." To facilitate the model's learning process, each flight is encoded using a label encoder. Flights arriving on time are assigned a code of "0" for the target variable "ArrDel15," while delayed flights are assigned a code of "1."

Various models are then tested and evaluated based on specific performance metrics. This comparison enables the selection of the most accurate model for the classification stage, ensuring the classifier effectively distinguishes between on-time and delayed flights.

From Table 4 , we can infer that **XGBoost** demonstrated superior power over the other classification algorithms with an accuracy of **0.92** along with a precision of **0.90** for delayed flights . All the other four algorithms performed equally well . We can see that the recall metric is quite low for all the algorithms , this is proven to be improved after smote .

Algorithm	Class	Precision	Recall	F1-score	Accuracy
Logistic	**0.0**	0.92	0.98	0.95	0.92
	1.0	0.89	0.68	0.77	
Extra Trees	**0.0**	0.92	0.97	0.94	0.91
	1.0	0.86	0.69	0.76	
Random Forest	**0.0**	0.92	0.97	0.95	0.92
	1.0	0.88	0.70	0.78	
Decision Trees	**0.0**	0.92	0.98	0.95	0.92
	1.0	0.89	0.68	0.77	
XGBoost	**0.0**	0.92	0.98	0.95	0.92
	1.0	0.90	0.69	0.78	

Table 4: Evaluation Metrics for Different Classifiers with Class Imbalance

4 Evaluation Metrics

A confusion matrix is a grid-like table that summarizes the performance of a classification model. It visually breaks down the number of correct and incorrect predictions for each category or class in the data. It shows how many flights were predicted correctly or incorrectly, categorized by their actual status (delayed or on-time). Here's a breakdown of the key terms:

- **FP (False Positive)**: On-time flights mistakenly classified as "delayed."
- **TN (True Negative)**: On-time flights correctly identified as "on-time."
- **FN (False Negative)**: Delayed flights incorrectly classified as "on-time."
- **TP (True Positive)**: Delayed flights correctly identified as "delayed."

Using the confusion matrix, researchers calculated different scores to assess the performance of the various models they tested. These scores are:

- **Accuracy**: This is the most straightforward metric, indicating the overall percentage of correct predictions the model makes.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision**: This score focuses on the quality of positive predictions (predicting a flight as delayed). It tells you what percentage of flights the engine labeled as "delayed" were actually delayed. In simpler terms, how often was the engine right when it said a flight would be delayed?

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall**: This score focuses on how well the engine captures all the actual delays. It tells you what percentage of the truly delayed flights the engine correctly identified as delayed. Imagine how often the engine identified a delayed flight as "delayed" compared to all the actual delays.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1 Score**: This score combines both precision and recall, providing a balanced view of the model's performance. It's like a single score summarizing how good the engine is at both correctly predicting delays and not missing any.

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

5 Class Imbalance

In many machine learning applications, especially those involving classification tasks, dealing with class imbalance is a significant challenge. Class imbalance occurs when the number of data points in one class is significantly higher than in other classes. In such cases, classifiers tend to become biased towards the majority class, often leading to inaccurate and unreliable predictions for the minority class.

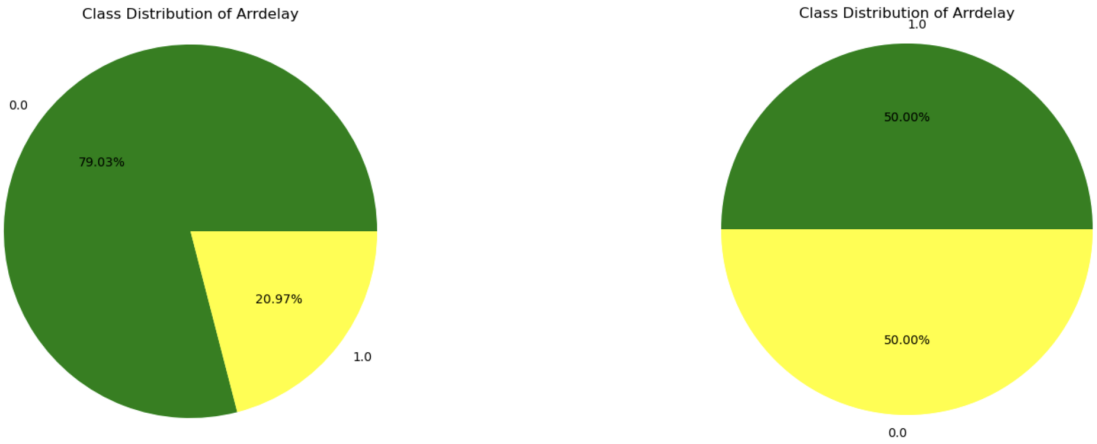
The majority class here is '0' meaning the flight is not delayed and the minority class here is '1' meaning the flight is delayed.

- **Resampling Techniques**:
  - *Oversampling*: Increase the number of instances in the minority class by duplicating them or generating synthetic samples using methods like SMOTE (Synthetic Minority Over-sampling Technique).
  - *Undersampling*: Reduce the number of instances in the majority class by randomly removing them.

6 Smote

Removing values from the dataset to balance classes, known as undersampling, is generally not advisable because it can lead to loss of important information and reduce the overall dataset size, which may negatively impact the model's performance. To address this issue, oversampling techniques are often employed. One of the most effective and widely used oversampling methods is SMOTE (Synthetic Minority Over-sampling Technique).

SMOTE is an advanced oversampling technique that creates synthetic samples of the minority class instead of merely duplicating existing instances. This approach helps to generate a more balanced dataset without reducing the size of the original dataset.



(a) Illustration of Class Imbalance (b) SMOTE

Figure 1: Class Distribution of Arrdelay

Algorithm	**Class**	**Precision**	**Recall**	**F1-score**	**Accuracy**
Logistic	**0.0**	0.94	0.93	0.93	0.90
	1.0	0.74	0.78	0.76	
Extra Trees	**0.0**	0.93	0.95	0.94	0.91
	1.0	0.81	0.73	0.76	
Random Forest	**0.0**	0.93	0.96	0.95	0.91
	1.0	0.83	0.73	0.78	
Decision Trees	**0.0**	0.93	0.96	0.94	0.91
	1.0	0.83	0.71	0.77	
XGBoost	**0.0**	0.92	0.98	0.95	0.92
	1.0	0.89	0.70	0.78	

Table 5: Evaluation Metrics for Different Classifiers after SMOTE

XGBoost emerged as the top performer with a maximum accuracy of **0.92**. This was marginally better than the other tree-based models (Extra Trees, Random Forest, and Decision Trees), which all achieved an accuracy of 0.91.

7 Regression

The second stage of the machine learning model focuses on regression, where the goal is to predict the exact arrival delay in minutes. Unlike the classification stage, which simply determines whether a flight is delayed, this stage provides a more detailed prediction by quantifying the delay duration.

To train the regression model, the flight data includes both delayed and non-delayed flights. This comprehensive approach ensures the model learns from a diverse range of scenarios, improving its predictive accuracy. The target variable in this stage is the actual delay time in minutes.

Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

MAE measures the average magnitude of errors in a set of predictions, without considering their direction. It's less sensitive to outliers compared to MSE.

Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

MSE measures the average squared difference between the estimated values and the actual value. It penalizes larger errors more heavily than MAE.

Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3}$$

RMSE is the square root of MSE. It's in the same units as the response variable, making it easier to interpret.

Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4}$$

R^2 represents the proportion of the variance in the dependent variable that is predictable from the independent variable(s). It ranges from 0 to 1, with 1 indicating perfect prediction.

These metrics help assess the performance of regression models, with lower values of MAE, MSE, and RMSE indicating better fit, while higher R^2 values (closer to 1) suggest better explanatory power of the model.

Metric	**Linear Regression**	**Extra Trees**	**Random Forest**	**XGBoost**
MAE	5.5808	5.6842	5.7339	5.7574
MSE	117.8361	110.1059	108.1586	199.4544
RMSE	10.8552	10.4931	10.3999	14.1228
R-squared (R^2)	0.9325	0.9369	0.9380	0.9380

Table 6: Evaluation Metrics for Different Regression Models before Pipelining

In the **regression** task, the **Random Forest** algorithm along with **XGBoost** algorithm demonstrated superior performance, achieving the highest R-squared value of **0.9380**, along with an MAE of 5 minutes indicating a strong fit to the data. This was closely followed by the Extra Trees model, with an R-squared of 0.9369. The accuracy was improved when a pipe-lined model was used .

8 Pipelining

This study proposes a novel two-stage machine learning approach for flight delay prediction. The system is designed to first classify whether a flight will be delayed and then predict the delay duration (in minutes) exclusively for flights identified as delayed. Unlike the previous section where all the data is considered, this sequential approach focuses solely on flights that are expected to be delayed. By leveraging the strengths of different machine learning algorithms, the proposed method aims to enhance prediction accuracy.

Figure 2 illustrates the flow of the pipeline model that is built for this purpose. Additionally, Table 7 presents the results of various models when predicting the delay duration for flights classified as "Delayed". This comprehensive approach not only improves the precision of delay predictions but also offers a more targeted analysis of flight delays, contributing to more reliable and actionable insights for the aviation industry.

Metric	**Linear Regression**	**Extra Trees**	**Random Forest**	**XGBoost**
MAE	13.0935	13.0129	12.9485	12.9279
MSE	344.3991	331.8444	328.5659	474.1362
RMSE	18.5580	18.2166	18.1264	21.7747
R^2	0.9433	0.9453	0.9459	0.9219

Table 7: Evaluation Metrics for Different Regression Models after Pipeling

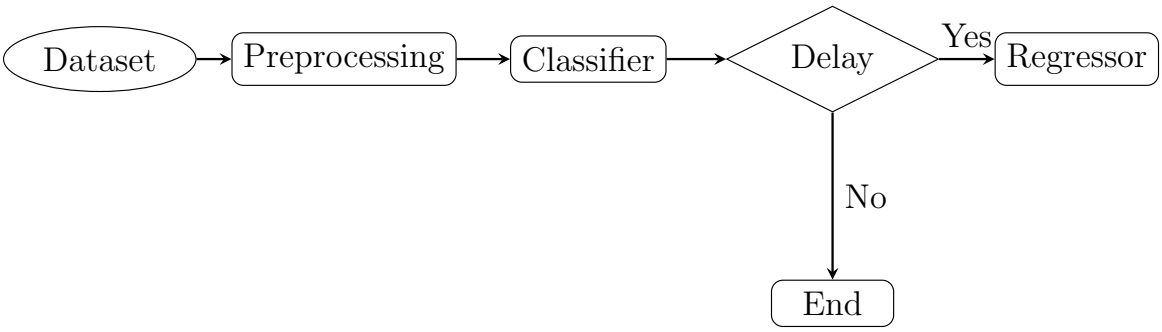


Figure 2: Flow of the pipeline model

The **Random Forest** algorithm demonstrated superior performance, achieving the highest R-squared value of **0.9459**, indicating a strong fit to the data. This was closely followed by the Extra Trees model, with an R-squared of 0.9453.The high F1-scores, particularly for class 0.0 (non-delayed flights), suggest that the models are effective in balancing precision and recall. XGBoost and Random Forest exhibited the best F1-scores of 0.95 for class 0.0, demonstrating their robustness in predicting both delayed and non-delayed flights.

9 Regression Analysis

The histogram presented in Figure 3 offers a detailed visualization of predicted flight arrival delays, providing valuable insights into the distribution and frequency of these delays across an extensive dataset. The x-axis, labeled "ArrDelayMinutes," denotes the duration of delays in minutes, while the y-axis indicates the frequency of occurrences for each delay interval.

The highly skewed nature of the distribution is immediately apparent, with a pronounced peak at the lower end of the delay spectrum. This suggests that the majority of flights experience relatively short delays, likely under 200 minutes.

The graph also exhibits a long right tail, extending beyond 2000 minutes, indicating that although extreme delays are possible, they are relatively rare. This right-skewed distribution is typical in flight delay data, reflecting the reality that most flights arrive close to their scheduled time, with a decreasing probability of longer delays.

As the number of flights are decreasing , we can see that the MAE metric increases.

Minutes	Count	MAE	MSE	RMSE
0–50	159163	8.889951	123.709669	11.122485
50–100	81775	9.373064	133.092989	11.536593
100–200	47016	14.728534	349.720236	18.700808
200–400	13085	18.818398	631.225131	25.124194
400–800	1686	19.496802	819.033326	28.618758
800+	459	14.790021	402.107003	20.052606

Table 8: Evaluation Metrics for Each Flight Delay Interval

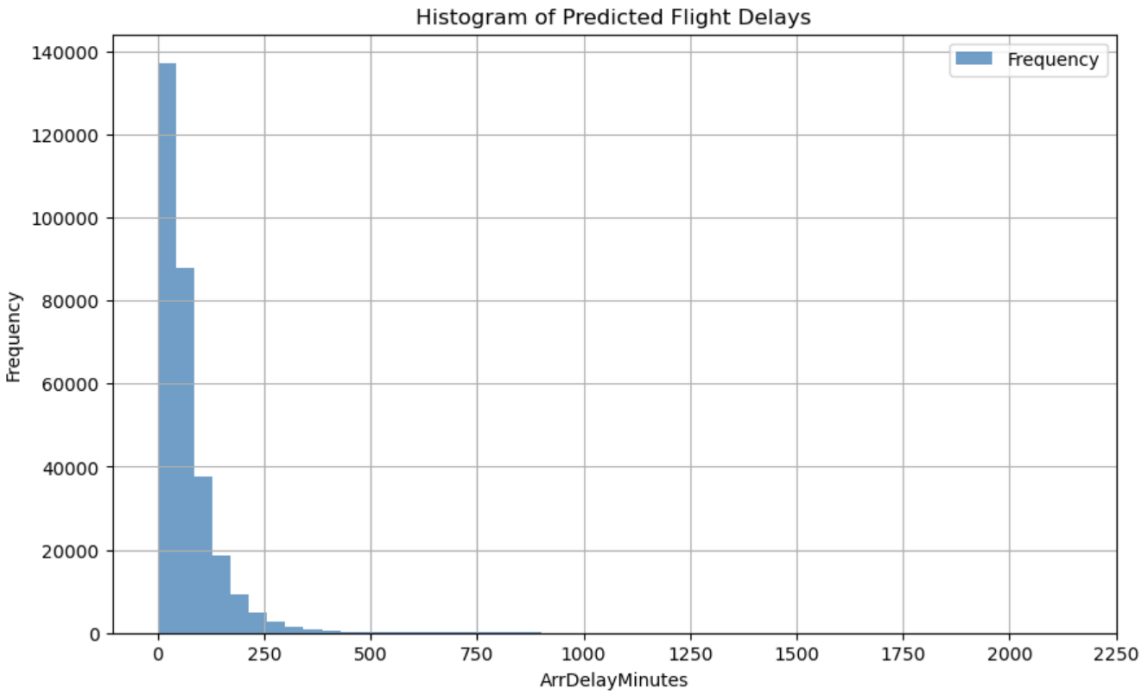


Figure 3: Distribution of Predicted Flight Arrival Delays

10 Conclusion

The comparative analysis of various machine learning models for flight delay prediction reveals promising results. In the **regression** task, the **Random Forest** algorithm demonstrated superior performance, achieving the highest R-squared value of **0.9459**, indicating a strong fit to the data.When the regressor was used for the complete dataset , we had an overall accuracy of 0.98 and a MAE of 3 minutes . This proves that pipelining has helped in increasing the accuracy of the model .

For the **classification** task, after applying SMOTE to address class imbalance, **XGBoost** emerged as the top performer with a maximum accuracy of **0.92**.

Furthermore , the Random forest Regressor was further analysed in different intervals and plotted using an histogram .From an operational and customer service

perspective, this distribution can be invaluable for airlines and airports in resource allocation, schedule planning, and setting passenger expectations . This comprehensive visualization of predicted flight arrival delays offers critical insights that can enhance operational efficiency and improve the overall passenger experience.

References

[1] Paper Name. "SMOTE: Synthetic Minority Over-sampling Technique", Available online: `https://arxiv.org/pdf/1106.1813`