Report On

Airline Flight Delay Predictor using Machine Learning

Regression and Classification

Author: Sadakopa Ramakrishnan T
Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam

Abstract

This project aims to predict flight delays at 15 major airports in the United States using a dual-stage machine learning model that combines classification and regression methods. Initially, a binary classifier determines whether a flight will be delayed by 15 minutes or more, identified by the target variable ***ArrDel15***. For flights predicted to be delayed, a subsequent regression model estimates the exact delay duration in minutes, represented by ***ArrDelMinutes***. The project involves comprehensive data collection and preprocessing of both flight and weather data. SMOTE is utilized to address class imbalance for the classification task. The resulting ***classifier*** and ***regressor*** achieve accuracies of ***0.92*** and ***0.94***, respectively.

Contents

1 Introduction

In today's fast-paced world, ensuring timely arrivals can be a challenging task—unless you have a reliable model to predict flight delays. This model uses various features to forecast delays, focusing on optimizing and normalizing input variables to achieve the best fit for both regression and classification tasks.

Regression and classification serve different purposes. Classification addresses yes-or-no questions, such as predicting whether a flight will be delayed or not. On the other hand, regression provides an estimate of the delay duration in minutes.

The project begins with data preparation and exploratory data analysis to refine the features. To handle class imbalance in classification, SMOTE (Synthetic Minority Over-sampling Technique) is applied to the training data, enhancing the model's performance. The predicted delays are stored in a dataframe for further pipelining. Linear regression is then employed to calculate the Root Mean Square Error (RMSE), providing insight into the delay duration. Ultimately, the model with the best accuracy from logistic regression and the highest R^2 value from linear regression is selected for pipelining.

2 Dataset

The dataset for this project is an extensive compilation of flight and weather data, carefully curated to facilitate the prediction of flight delays.The dataset has data for the year 2016 and 2017.

2.1 Flight Data

The flight data contains complete details of the flight schedules and on time performance for 15 U.S. Airports throughout 2016 and 2017. Flight Data, **DayofMonth, Time CRSDepTime, Quarter, Year** are used to help show seasonal variation in delays. **DepTime, CRSDepTime ArrTime** and **CRSArrTime** serve as critical operational information that allows comparisons between actual times and scheduled times thus helping spot delays. The dataset includes **DepDelayMinutes, ArrDelayMinutes** indicating the durations of delays and some binary variable like **ArrDel15** as a target for classification models that capture flights delayed above 15 minutes. Airport-specific delay analysis can be made possible through unique identifiers like **OriginAirportID** and **DestAirportID** which link flights with certain airports.

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

Table 1: Recommended Flight Columns

Below are the 15 U.S. airports:

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

Table 2: Airport Codes

2.2 Weather Data

The dataset contains in-depth meteorological information important for flight operations. It includes features such as **WindSpeedKmph, WindDirDegree, Visibility**, and **WeatherCode**, providing valuable insights into weather conditions that can impact flight schedules. Precipitation (**precipMM**), atmospheric pressure (**Pressure**), and cloud cover (**Cloudcover**) play crucial roles in understanding how the weather affects flight delays. Additionally, temperature-related data (**tempF, WindChillF, DewPointF**) and **humidity** levels offer further context on the operational conditions for flights. This detailed weather information is matched with flight data based on the **date, time, and airport**, ensuring accurate synchronization of weather and flight events.

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

Table 3: Recommended Weather Columns

3 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:

	Predicted Delay	**Predicted On-time**
Actual Delay	TP	FN
Actual On-time	FP	TN

Table 4: Confusion Matrix

- **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.

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

4 Classification

The objective here is to model a classifier that determines whether a given flight will be delayed or not using the features. 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." Before classification, SMOTE is performed to handle class imbalance in the classification data.

The classification columns:

Column Names
Year
Month
DayofMonth
OriginAirportID
DestAirportID
DepDelayMinutes
CRSArrTime
ArrDel15
rounded_CRSDepTime
WindSpeedKmph
WindDirDegree
WeatherCode
precipMM
Visibilty
Pressure
Cloudcover
WindChillF
Humidity

Table 5: List of Classification Columns

Synthetic Minority Over-sampling Technique (SMOTE) is applied on training data rather than on the full dataset because information from the testing set can leak into the training set. This can lead to overly optimistic performance metrics because the model has effectively "seen" part of the test data during training.

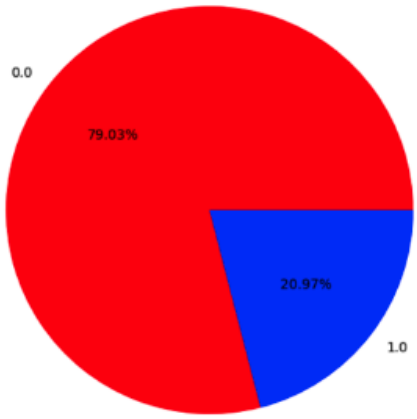

Figure 1: Class Imbalance in the dataset

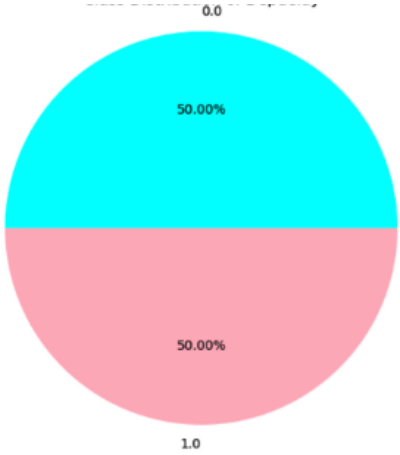

Figure 2: Class Balance after SMOTE

4.1 Logistic Regression

Logistic regression is a statistical method used for binary classification problems. It models the probability of a binary outcome based on one or more predictor variables. Logistic regression uses the logistic function to output probabilities that are then mapped to binary classes (e.g., 0 or 1).

Accuracy for Logistic Regression came out to be: 0.895

4.2 Decision Trees Classifier

A decision tree classifier is a machine learning model used for classification tasks. It works by splitting data into branches based on feature values, leading to a decision node or a leaf node. Each leaf node represents a class label, and each internal node represents a decision based on a feature. The process continues until it reaches a stopping criterion,

Algorithm	Class	Precision	Recall	F1-score	Accuracy
Logistic	**0.0**	0.94	0.93	0.93	0.89
	1.0	0.74	0.78	0.76	
Extra Trees	**0.0**	0.93	0.95	0.94	0.90
	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	0.91
	1.0	0.89	0.70	0.78	

Table 6: Evaluation Metrics for Different Classifiers before SMOTE

such as maximum depth or minimum samples per leaf, allowing the model to predict the class of new data based on the learned splits. These are highly sensitive to training data which could result in high variance.

Accuracy for Decision Trees Classifier: 0.901

4.3 Extra Trees Classifier

An Extra Trees Classifier, or Extremely Randomized Trees Classifier, is an ensemble learning method similar to a Random Forest. It builds multiple decision trees using the whole original dataset (without bootstrapping) and introduces extra randomness by selecting the split points randomly for each tree. The final prediction is made by averaging the predictions of all trees for regression tasks or by majority voting for classification tasks. This method can lead to faster training times and often better generalization.

Accuracy for model using Extra Tress Classifier: 0.905

4.4 XGBoost Classifier

The XGBoost Classifier is a sophisticated and efficient machine learning algorithm based on the gradient boosting framework. It builds an ensemble of decision trees in a sequential manner, where each new tree is trained to correct the errors made by the previous trees. This process focuses on improving the overall model by minimizing a specified loss function using gradient descent techniques.

Accuracy for XGBoost Classifier: 0.912

4.5 Random Forest

A Random Forest Classifier is an ensemble learning method that combines multiple decision trees to improve classification accuracy and prevent overfitting. Each tree in

the forest is trained on a random subset of the data, and the final prediction is made based on the majority vote of all the trees. Random Forest is called RANDOM because it uses 2 random processes: **Bootstrapping** and **Aggregating**. By training each tree on a different subset of data and combining their predictions, the model becomes less prone to overfitting compared to a single decision tree.

Accuracy for model using Random Forest: 0.908

The same dataset after SMOTE will yield the following results:

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 7: Evaluation Metrics for Different Classifiers with Class Balance

5 Regression

Regression is a statistical method used to model the relationship between a dependent variable (target) and one or more independent variables (features). Objective here is train a regression engine which predicts the Arrival delay period (in minutes) for delayed flights The variable are the features except 'ArrDelayMinutes' and target is 'ArrDelayMinutes'.

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.

5.1 Linear Regression

Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables by fitting a straight line to the data. The equation of the line minimizes the difference between predicted and actual values, allowing for predictions of continuous outcomes based on input features.

Linear Regression R^2: 0.934

5.2 Extra Trees Regressor

The Extra Trees Regressor is a type of ensemble learning method that combines multiple decision trees. Unlike traditional decision trees, it introduces randomness by selecting random subsets of features and random thresholds for splitting nodes. This randomness helps to reduce overfitting and can often lead to improved performance on unseen data. It's particularly useful when you have a large number of features.

Extra Trees Regressor R^2: 0.935

5.3 XGBoost Regressor

The XGBoost Regressor is a powerful machine learning algorithm that builds an ensemble of decision trees sequentially to predict continuous outcomes. Each new tree corrects the errors of the previous ones, and the process uses gradient boosting to minimize a loss function. This results in highly accurate and efficient predictions.

XGBoost Regressor R^2: 0.894

5.4 Random Forest

Random Forest Regressor is a powerful machine learning algorithm used for predicting continuous outcomes. It works by constructing multiple decision trees during training and outputs the average prediction of the individual trees. Key advantages include handling non-linear relationships, robustness to outliers, and minimal hyperparameter tuning.

Random Forest Regressor R^2: 0.937

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)	0.9341	0.9352	0.9371	0.8946

Table 8: Evaluation Metrics for Different Regression Models

5.5 Test for over fitting

To test the regression model, CRSArrTime and ArrTime columns were excluded from the feature set to avoid data leakage, ensuring that only relevant features known before the flight arrival were used. Cross-validation was employed to evaluate the model's performance across different subsets of data, providing a robust assessment of its generalization capability. Additionally, ridge regression was utilized for regularization to handle potential overfitting by controlling the model's complexity.

6 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 3 illustrates the flow of the pipeline model that is built for this purpose. Additionally, Table 8 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.

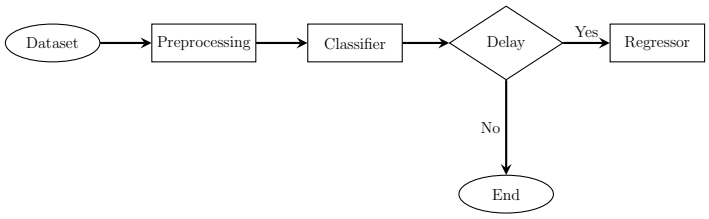

Figure 3: Pipeline model for flight delay prediction

Metric	**Linear Regression**	**Extra Trees**	**Random Forest**	**XGBoost**
MAE	12.8565	12.8946	12.7735	12.8574
MSE	329.2689	323.1660	317.2874	514.8283
RMSE	18.1457	17.9768	17.8126	22.6898
R^2	0.9450	0.9460	0.9470	0.91411

Table 9: Evaluation Metrics for Different Regression Models after Pipeling

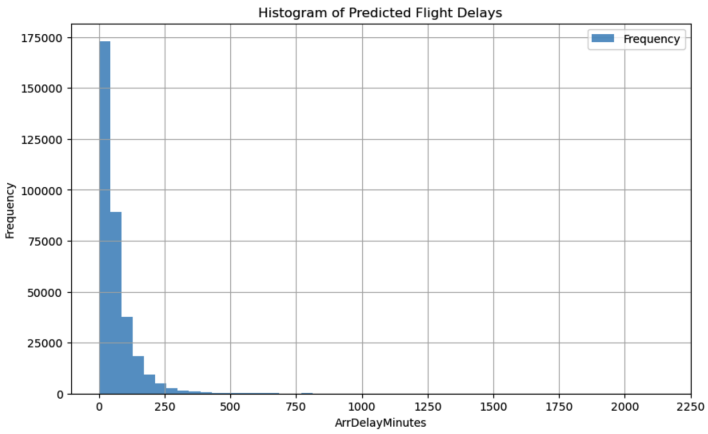

Figure 4: Regression Analysis

7 Regression Analysis

This histogram illustrates the distribution of predicted flight delays, measured in ArrDelayMinutes. The graph reveals a highly right-skewed distribution with a pronounced peak in the first bin, likely representing delays between 0 and 50 minutes. This indicates that while most flights are predicted to have minimal delays, there's a long tail extending to the right, showing increasingly rare occurrences of longer delays. The frequency of delays decreases rapidly as delay time increases, with the x-axis extending to 2250 minutes (about 37.5 hours), capturing even extreme delay predictions. The use of 50 bins provides a detailed view of the distribution, allowing for nuanced analysis. This pattern suggests that while the airline can expect most flights to be on time or only slightly delayed, it must also be prepared for a range of delay scenarios, including some rare but significant outliers. Understanding this distribution is crucial for the airline's resource allocation, customer communication strategies, and operational planning.

Minutes	Count	MAE	MSE	RMSE
0 - 50	182268	3.767565	85.252377	9.233221
50 - 100	82351	4.097892	108.831460	10.432232
100 - 200	46368	4.394075	124.113897	11.140642
200 - 400	12995	4.746977	147.648184	12.151057
400 - 800	1666	4.929616	160.218706	12.657753
800+	459	5.095926	197.933855	14.068897

Table 10: Regression Metrics for Different Ranges of Predicted Delay Minutes

The count is decreasing as the range in minutes are increasing suggesting very less flights have extremely long delays. The increase in RMSE and MAE with longer predicted delays indicates that the model's prediction accuracy decreases as the delay

time increases. This is likely due to the higher complexity, variability, and presence of outliers in longer delays.

Summary of Table 10

0 - 50 Minutes

Performance: The model performs reasonably well with smaller errors (MAE: 3.77, RMSE: 9.23).
Inference: Short delays are predicted with relatively high accuracy.

50 - 100 Minutes

Performance: The errors start to increase (MAE: 4.10, RMSE: 10.43).
Inference: Moderate delays are slightly harder to predict accurately.

100 - 200 Minutes

Performance: Further increase in errors (MAE: 4.39, RMSE: 11.14).
Inference: Longer delays introduce more variability and prediction difficulty.

200 - 400 Minutes

Performance: Errors continue to increase (MAE: 4.75, RMSE: 12.15).
Inference: The complexity and variability in longer delays become more pronounced.

400 - 800 Minutes

Performance: Significant errors (MAE: 4.93, RMSE: 12.66).
Inference: Very long delays are challenging to predict accurately, likely due to the reasons mentioned above.

800+ Minutes

Performance: The highest errors (MAE: 5.10, RMSE: 14.07).
Inference: Extremely long delays are rare and highly variable, leading to substantial prediction errors.

8 Conclusion

The analysis underscores the proficiency of ensemble methods in predicting flight delays, with Random Forest and XGBoost outperforming in regression and classification tasks, respectively. Random Forest achieved an impressive R-squared of 0.9380, while XGBoost led in classification accuracy at 0.92. The high F1-scores for class 0.0 highlight their effectiveness. Additionally, histogram visualizations of the Random Forest regressor's predictions provided valuable insights for operational planning and resource management. These results confirm the potential of ensemble methods in flight delay prediction, suggesting further refinements and integration into comprehensive prediction frameworks for enhanced performance.