Real Time Flight Delay Prediction

Abstract. A flight delay is when an airline flight takes off and/or lands later than its scheduled time. This project focuses on classifying whether a flight is delayed using different classifiers and the one that produces the best result is chosen. Then regression is used to compute the minutes by which the flight might be delayed using weather and flight data and analyze where the regressors work best.

1 Introduction

Flight enables highly perishable and valuable products to be moved fast over long distances, but the scheduled flights may deviate from the schedule due to environmental factors, the main environmental stressors are time pressure, turbulence, noise, and temperature. Flight delays not only displeases air passengers and disrupt their schedules but also cause a decrease in efficiency, reallocation of flight crews and aircraft, and additional crew expenses.

Air transport includes passenger and freight airplanes, that is, aircraft configured for transporting passengers, freight, or mail. Thus a higher demand for air transport results in more air traffic thus reducing reliability .

This project aims to predict whether a flight may delay or not depending upon environmental conditions at the time of arrival of that particular flight towards its airport and subsequently predict the total number of minutes the flight may be delayed in **minutes** using Machine Learning Algorithms, i.e., classifiers and regressors. The data used is then manipulated to study where the models perform better and the results are then analyzed.

2 Dataset

The datasets used to train the machine learning models are flight and weather dataset. The flight dataset contains the details of all the flights that flew inside the USA between 2016 and 2017. On the other hand, the weather dataset contains the weather report recorded every 1 hour across the USA .

Table 1: Features selected from flight dataset

FlightDate	Quarter	Year	Month
DayofMonth	DepTime	DepDel15	CRSDepTime
DepDelayMinutes	OriginAirportID	DestAirportID	ArrTime
CRSArrTime	ArrDelayMinutes		

Table 2: Features selected from weather dataset

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

The flights are selected from 15 specific airports as referenced in Table 3 and segregated by that basis. After segregating the features from their core dataset, each individual flight is combined with its corresponding weather data at its time of arrival at its airport.

Table 3: Airports selected

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

3 Classification

The classifiers are trained to classify whether a flight is delayed or not. Arrdel15 is used as the target variable, which holds the value 1 if the flight produces a delay of more than 15 minutes and 0 otherwise. The dataset is then split on a 7:3 train over test ratio to train the classifiers. The **classifiers** explored in this project to classify are **XGBoost**, **RandomForest**, **DecisionTree**, **LogisticRegression** and **ExtraTrees** classifiers.

3.1 Evaluation Metrics

The metrics used to evaluate the classifiers in this project are **precision**, **recall**, **f1-score**.

Precision Precision is a metric that quantifies the number of correct positive predictions made. Precision, therefore, calculates the accuracy for the minority class.

$$Precision = TP/(TP + FP) \tag{1}$$

Recall Recall is a metric that quantifies the number of correct positive predictions made out of all positive predictions that could have been made. Unlike precision that only comments on the correct positive predictions out of all positive predictions, recall provides an indication of missed positive predictions.

$$Recall = TP/(TP + FN) \tag{2}$$

F1-Score F1-Score provides a way to combine both precision and recall into a single measure that captures both properties. Alone, neither precision or recall tells the whole story. We can have excellent precision with terrible recall, or alternately, terrible precision with excellent recall. F1-Score provides a way to express both concerns with a single score.

$$F1 - Score = (2 * Precision * Recall)/(Precision + Recall) \tag{3}$$

Accuracy Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$Accuracy = TNP/NCP \tag{4}$$

TP = TruePositives
FP = FalsePositives
FN = FalseNegatives
TNP = Total no of Predictions
NCP = Number of correct Predictions

3.2 Classification Report

The following table contains the results produced by the classifiers:

Table 4: Classifier results

	Precision		Recall		F1-Score		
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	Accuracy
DecisionTree	0.90	0.60	0.89	0.63	0.90	0.62	0.84
XGBoost	0.85	0.96	1.0	0.33	0.92	0.49	0.86
ExtraTrees	0.87	0.82	0.89	0.72	0.88	0.78	0.86
RandomForest	0.90	0.85	0.97	0.73	0.94	0.79	0.89
Logistic	0.81	0.99	1.0	0.15	0.90	0.25	0.82

The best classifier is chosen in terms of its f1-score. It can be observed that some classifiers produce high f1-score for class 0 but poor score for class 1. Thus, it can be inferred that the classifier predicted majority to be no delay even if it was delayed, which defeats the purpose of the project, hence the best classifier is chosen in consideration of the f1-score of the minority class(1). However, due to heavy data imbalance between the classes, without implementing sampling techniques the best classifier cannot be chosen. The sampling done are oversampling and undersampling.

3.3 Data Imbalance

It can be seen that the dataset is largely imbalanced(see Fig. 1), class 0 takes upto 79% of the data. This may result the classifier to get used to majority class more than minority class, which can be seen in table 4, making the classifier to show bias and being not very accurate. This data imbalance might have a bigger impact on the efficiency of the classifier. Thus, methods such as **Undersampling** and **Oversampling** is used to observe the impact of the imbalance on the classifiers.

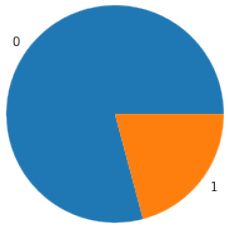


Fig. 1: Data Imbalance

UnderSampling

Undersampling is a technique to balance uneven datasets by keeping all of the data in the minority class and decreasing the size of the majority class. This makes the classifier to train with equal datapoints for each class which prevents the classifier from showing bias. RandomUnderSampler is used for undersampling the dataset. RandomUnderSampler is a fast and easy way to balance the data by randomly selecting a subset of data for the targeted classes, it randomly deletes the rows of the majority classes according to our sampling strategy. The scores that the classifiers produced are tabulated below in Table 5:

Table 5: UnderSampling results

	Precision		Recall		F1-Score		
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	Accuracy
DecisionTree	0.94	0.51	0.79	0.80	0.86	0.62	0.80
XGBoost	0.94	0.73	0.92	0.79	0.93	0.76	0.88
ExtraTrees	0.95	0.65	0.88	0.81	0.92	0.72	0.87
RandomForest	0.88	0.80	0.92	0.78	0.90	0.79	0.90
Logistic	0.92	0.74	0.93	0.76	0.93	0.75	0.88

OverSampling

Oversampling involves duplicating examples in the minority class. These new examples can be synthesized from the existing examples. This is a type of data augmentation for the minority class and is referred to as the Synthetic Minority Oversampling Technique, or SMOTE for short. In our case, Class 1 is the minority, so it is duplicated until it equalizes the majority in frequency. While, Class 0 is the Majority, so it is left as it is. The table below shows the scores that the classifiers produced:

Table 6: OverSampling results

	Precision		Recall		F1-Score		
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	Accuracy
DecisionTree	0.92	0.68	0.91	0.71	0.92	0.69	0.87
XGBoost	0.93	0.81	0.95	0.73	0.94	0.77	0.91
ExtraTrees	0.93	0.76	0.92	0.75	0.93	0.76	0.90
RandomForest	0.95	0.84	0.94	0.84	0.94	0.80	0.93
Logistic	0.94	0.75	0.93	0.77	0.94	0.76	0.90

From the scores obtained through oversampling, it can be inferred that the scores after oversampling have subsequently gotten better than the scores produced by the classifiers with No-sampling/Under-sampling. This implies that the data-imbalance did actually have an impact on the classifiers, making the classifiers to show bias towards the majority class, thus affecting their performance.

After implementing sampling techniques and observing the performance of the classifiers, **randomforest** classifier produced the best result in terms of true positive predictions and less false positives. Thus randomforest is chosen as the best classifier between the five classifiers explored.

4 Regression

Regression is a method that attempts to determine the strength and character of the relationship between one variable (usually denoted by Y) and a series of other variables. It is a supervised learning technique which helps in finding the correlation between variables and enables us to predict the continuous output variable based on one or more predictor variables. In this project **ArrDelMinutes** is Y and all the other columns mentioned in Table 1 and Table 2 are X.

4.1 Evaluation Metrics

To build and deploy a generalized model we require to Evaluate the model on different metrics which helps us to better optimize the performance, fine-tune it, and obtain a better result. The metrics used to evaluate the regressors are **Mean absolute error**, **Mean squared error** and **R Squared**.

Mean absolute error It is the difference between the actual value and predicted value that is an absolute error but we have to find the mean absolute of the complete dataset used.

$$MAE = (1/n)\sum_{i=1}^{n}\left|x_i - x_p\right| \tag{5}$$

Mean squared error It represents the squared distance between actual and predicted values. The squared is done to avoid the cancellation of negative terms and it is the benefit of MSE.

$$MSE = (1/n)\sum\left|x_i - x_p\right|^2 \tag{6}$$

R Squared R2 score is also known as Coefficient of Determination or sometimes also known as Goodness of fit. A good R2 score is when it s closer to 1.

$$RSquared = 1 - (SSr/SSm)$$

SSr = Squared sum error of regression line
SSm = Squared sum error of mean line
x_p = Predicted value
x_i = Mean value

4.2 Regression Scores

The results of all the regressors namely Linear regressor, XGB, ExtraTrees, RandomForest, are tabulated below in Table **??**.

Table 7: Core Dataset

Regressors	MSE	MAE	R2score
Linear	19.81	14.45	0.9234
ExtraTrees	16.36	10.90	0.9477
XGB	17.02	11.68	0.9435
RandomForest	16.82	11.79	0.9448

The dataset is split and each regressor is trained and tested to analyze which regressor is more accurate in terms of predicting the delay in minutes.

5 Pipeline Architecture

The regressors are trained and tested using two different datasets separately to see which yields the best result. The first dataset comprises of data which belongs to flights which were predicted to be delayed by the best classifier(RandomForest). The second dataset comprises of data which consists of only delayed flights from the main dataset. The process is depicted below in fig 2.

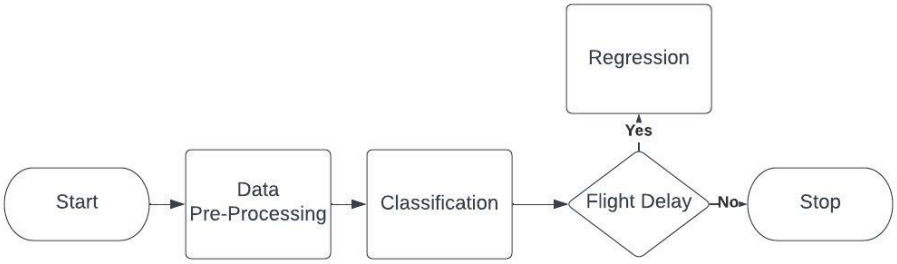


Fig. 2: Pipeline Architecture

It can be observed that both the goodness of fit as well as the error made by the regressors has increased noticeably which were trained using classifier predicted dataset. The scores are tabulated below:

Table 8: Classification Dataset

Regressors	MSE	MAE	R2score
Linear	17.95	12.77	0.9527
ExtraTrees	17.27	11.80	0.9563
XGB	17.10	11.52	0.9571
RandomForest	17.42	11.78	0.9555

From both the tables tabulated above (Table 7, Table 8) it can be observed that the ExtraTrees regressor was the most accurate regressor compared with all the other explored regressors.

6 Regression Analysis

Regression analysis helps us to understand how the value of the dependent variable is changing corresponding to an independent variable when other independent variables are held fixed. In this project, the ExtraTrees regressor is trained with different datasets consisting of datapoints of a particular range. The regressor is trained in five different ranges(i.e.,15-300, 300-600, 600-1000, 1000-1500, 1500+).

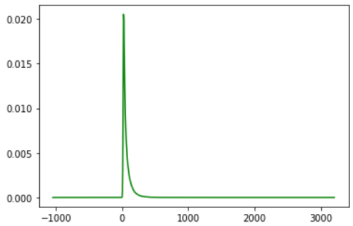
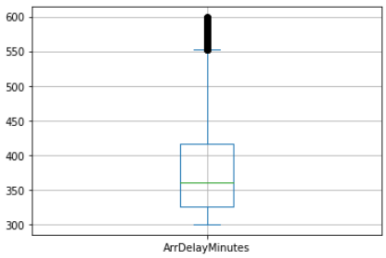


Fig. 3: Delay minutes density

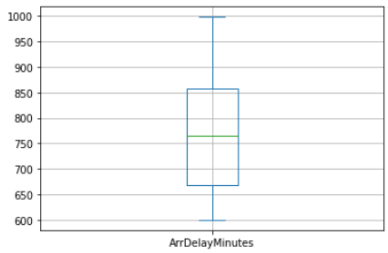
The scores which were produced by the regressors trained and tested with different intervals of the dataset are tabulated below:

Table 9: Regression Analysis

Intervals	MSE	MAE	R2score	Datapoints
15-300	16.60	11.76	0.8893	94842
300-600	31.93	18.96	0.7904	1052
600-1000	24.61	17.16	0.9464	176
1000-1500	20.23	16.92	0.9719	38
1500+	33.25	15.87	0.8161	3



(a) 600-1000



(b) 1000-1500

Fig. 4: Intervals

It can be inferred from the above table that the regressor performed best in the interval 1000-1500 producing low errors and good fit. The prediction here is supposed to be even tougher than its preceding ranges due to the wide range and diversity of the datapoints. It can also be seen that the regressor produced poor result in the interval 300-600 even though it is supposed to be easier to predict than the successive ranges. After thorough analysis it was found that every datapoint(1052) in this range was split to the testing dataset, due to which the regressor had no data to train and learn from, this performing poor. The interval 15-300 had the highest datapoints to predict and achieved low errors and

a overall good fit. The interval 1500+ had significantly low datapoints to learn from than its previous interval, but, still, it produced a good fit but a noticeable error increase.

From the above scores it can be concluded that the model made the least error and performed well in the intervals 1000-1500 and 600-1000. However, it produced a lot of errors in the interval 300-600 due to lack of datapoints to learn from.

7 Conclusion

The RandomForest classifier was able to predict whether a flight is delayed or not with satisfactory accuracy and gave a f1-score of 0.94 for the majority class and 0.79 for the minority class. Among the explored regressors, the ExtraTrees regressor produced a decent result in terms of prediction and gave an average error of 11 minutes off to the actual arrival delay. The MSE on the other hand produced an error close to 16 minutes off the actual arrival delay. The predicted time were good when compared to the diversity of the datapoints. Overall, this project was successful in achieving the desired objectives. However, further development of this project can yield better scores and prove to be more accurate .