Flight Delay Prediction

Gundepudi V Surya Sashank

gundepudi18047@mech.ssn.edu.in

Abstract. Flight scheduling has been a problem since the dawn of air travel and is something that airline companies wish to tackle. For an airport to be able to schedule the flights such that they reach on time, they must be able to tell if the flight will arrive on time or not. A flight is said to be delayed if the flight either takes off or arrives later than the scheduled time. This Project predicts whether if the flight will arrive delayed or not, after the flight's departure, and if the flight is classified as arriving late, then the arrival delay in minutes is predicted.

Keywords: Machine Learning, Prediction, Data Analysis

1 Introduction

1.1 Scope

Since the inception of commercial air travel, the number of people travelling by air has increased drastically, with an increase of 42 % in the last decade alone. This means that there will be even more air traffic than usual at a given point of time and hence scheduling flights will be a colossal problem for the Aviation Department.
When a flight is delayed it will cause issues for the customers in the form of loss of money and time. Not only does it disturb the lives of the customers travelling by air commercially but it also destroys the integrity of the airline company.
Flights can be delayed due to various reasons, one of them being, extreme weather conditions. Since it is possible for the Aviation Department to estimate the weather conditions after the flight departs it may help them schedule flights better and hence reduce air traffic and also make commercial air travel smooth.
Hence it is critical to be able to predict if a flight will be delayed or not and if delayed by how long.

1.2 About the Project

This project examines the impact of various weather conditions on the arrival delay for 15 domestic flights in the United States. It uses a two stage machine learning model to classify and predict the arrival delays of various flights in 15 different airports during the years 2016 - 2017. The machine learning engine's Classification and Regression algorithms are then evaluated with standard metrics and hence compared.

2 Data Pre-Processing

2.1 Flight On-time Performance Data

This data set contains the On-time performance for various flights over the years 2016 and 2017. The airports and flight attributes taken into consideration are given in Table 1 and Table 2 respectively,

Table 1 Airports taken into consideration

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

Table 2 Flight attributes taken into consideration

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

2.2 Weather Data

Weather data for the airports in interest was collected hourly. The weather features under consideration are shown in table 3

Table 3 Weather Features taken into consideration

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

From the data collected only the years 2016 and 2017 were taken into consideration because we are examining the performance of the flights for those years alone.

2.3 Merging the data

Finally, after the required data was collected the flight and weather data were merged on the - Airport the flight is departing from, the date the flight is departing and the time at which the flight is departing.

3 Classification

Within this section we will look at - **classification**, where the classifier must predict if the flight will be arrive late or on time.

3.1 What is Classification?

Classification is an instance of supervised learning. Within classification we aim to predict a class under which an object will fall into.
With respect to the problem statement at hand, **ArrDel15** is a binary categorical variable that holds a value of 0 for flights that arrived on time and a value of 1 for flights that arrived late. The classifier will need to predict if the flight will fall into class 0 (On-time) or class 1 (Delayed).

3.2 Algorithms Used

The following algorithms have been used and evaluated.

1. Logistic Regression
2. Random Forest
3. Extra Trees
4. Decision Trees
5. XGBoost

3.3 Splitting the Data into Train and Test Data

ArrDel15 (Which tells us if the flight is delayed or not) and **ArrDelayMinutes** (Which gives us the number of minutes by which the flight is delayed) were removed for our independent variable, because these two are considered ground truth features which we will not know beforehand. The data was split into test and train in a 70:30 ratio.

3.4 Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.
The following are some terminology related to a confusion matrix
TP - True Positive, which means the number instances that were classified correctly. In the current use case it refers to the number of flights that were classified correctly as Delayed.
FP - False Positive, which refers to the number instances that incorrectly indicates the occurrence of an instance. In the current use case it refers to the number of flights that were classified as Delayed but were actually On-time.
TN - True Negative, which is the number instances that were classified correctly for the non-occurrence of an instance. In the current use case it refers to

the number of flights that were correctly classified as On-time.
FN - False Negative, which refers to the number instances that were classified incorrectly for the non-occurrence of an event. In the current use case it refers to the number of flights that were classified as on-time for flights that were Delayed.

A confusion matrix is drawn with each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa). A representation of one can be seen in Figure 1.

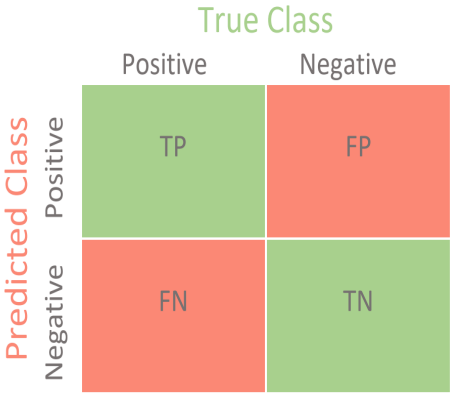

Fig. 1: Confusion Matrix

3.5 Metrics

Precision Precision quantifies the number of positive class predictions that actually belong to the positive class. Therefore it tells us how many of the classified items are relevant.
With respect to our problem at hand it gives us the proportion of the flights which have been classified correctly, either as delayed or not delayed, with respect to the total number of classified flights.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Recall Recall quantifies the number of positive class predictions made out of all positive examples in the data-set.
With respect to our problem at hand it gives us the proportion of flights it has classified as delayed with respect to the total number of delayed Flights.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

F1 Score or F- Measure F1 Score or F-Measure provides a single score that balances both the concerns of precision and recall in one number. It is evaluated as the harmonic mean of Precision and Recall.

$$F1Score = (\frac{Precision^{-1} + Recall^{-1}}{2})^{-1} = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3)$$

3.6 Results

The results of the given classification algorithms have been evaluated with the metrics refereed to in Section 3.5. Class 0 refers to the flight arriving on time and class 1 refers to the flight arriving late.

Table 3 Results for classification without sampling

MODEL	Precision		Recall		F1Score		Accuracy
	0	1	0	1	0	1	
Logistic Regression	0.92	0.89	0.98	0.68	0.95	0.77	91.58
Random Forest	0.92	0.89	0.98	0.70	0.95	0.78	91.81
Extra Trees	0.93	0.83	0.96	0.74	0.95	0.78	91.19
Decision Trees	0.92	0.68	0.91	0.70	0.92	0.69	86.77
XGBoost	0.92	0.90	0.98	0.73	0.95	0.79	91.93

3.7 Class imbalance and Sampling methods

The performance of the classifier on Class 1 is weaker than the performance of Class 0 and one of the major reasons for this could be attributed to the class imbalance between the two of them.

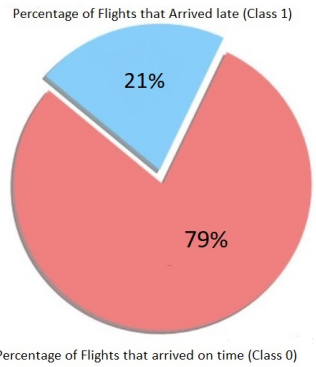

Fig. 2: Class difference before Sampling

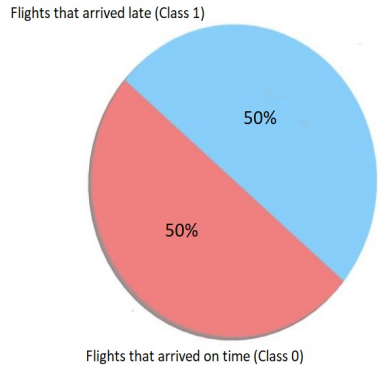

Fig. 3: Class difference after Sampling

To tackle the problem of Class imbalance there are standard sampling methods. Some of which used here are

1. **SMOTE**
   - SMOTE (Synthetic Minority Oversampling Technique) is an oversampling technique that generates synthetic samples from the minority class. Rather than replicating the minority observations, SMOTE works by creating synthetic observations based upon the existing minority observations .
2. **Random Undersampling (RUS)**
   - Random under-sampling involves randomly selecting examples from the majority class and deleting them from the training data-set

Let us look at the performances after sampling

Table 5 Results for classification after SMOTE

MODEL	**Precision**		**Recall**		**F1Score**		**Accuracy**
	0	1	0	1	0	1	
Logistic Regression	0.94	0.74	0.93	0.78	0.93	0.76	89.65
Random Forest	0.93	0.84	0.96	0.74	0.95	0.78	89.27
Extra Trees	0.94	0.79	0.95	0.76	0.94	0.77	88.66
Decision Trees	0.92	0.66	0.90	0.71	0.91	0.68	79.10
XGBoost	0.92	0.90	0.98	0.70	0.95	0.78	89.59

Table 6 Results for classification after RUS

MODEL	**Precision**		**Recall**		**F1Score**		**Accuracy**
	0	1	0	1	0	1	
Logistic Regression	0.94	0.74	0.93	0.78	0.93	0.76	89.65
Random Forest	0.95	0.72	0.91	0.81	0.93	0.76	89.27
Extra Trees	0.95	0.70	0.90	0.82	0.93	0.75	88.66
Decision Trees	0.94	0.50	0.79	0.80	0.86	0.62	79.10
XGBoost	0.95	0.73	0.92	0.80	0.93	0.76	89.59

3.8 Inference

Considering both Precision and Recall is important in this use case, therefore considering F1 score as a defining metric is better because it provides a single score that balances both the concerns of precision and recall and gives us a good weighted average of the two. The XGBoost Classifier outperforms the other classification algorithms.

4 Regression

Within this section we will look through -**regression**. The regressor must predict the number of minutes by which a flight is delayed given that the flight arrives late.
In statistical modeling, regression analysis is a set of statistical processes for estimating the relationships between a dependent variable and one or more independent variables.
With respect to the problem at hand we will be predicting, for flights that have been predicted to arrive late, the number of minutes by which the flight arrived late given other features will be predicted.

4.1 Algorithms Used

The algorithms used in this section are

1. Linear Regression
2. Extra trees Regression
3. XGBoost Regression
4. Random Forest Regression

4.2 Metrics

To evaluate the performance of our machine learning model we will use the following metrics.

1. **Mean Absolute Error (MAE):**
   - Mean absolute error is a measure of the absolute errors of our predicted value with respect to the true value.

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

2. **Root Mean Squared Error (RMSE):**
   - RMSE measure of the differences between values predicted by a model or an estimator and the values observed.

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

3. **R-Squared (R^2):**
   - -R^2 is defined as the ratio of the sum of squares explained by a regression model and the "total" sum of squares around the mean.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

Where,
y_i : Actual Value/Ground Truth Value
$\hat{y}$: Predicted Value
$\overline{y}$: Mean of all instances of y_i
n : The total number of instances of y_i

4.3 Results

The results for our regression model have been evaluated using the metrics mentioned in Section 4.2.

Table 7 Results for Regression Models

MODEL	MAE	RMSE	R^2
Linear Regression	12.2689	17.6820	0.94135
Extra Trees	11.9296	17.1652	0.94472
Random Forest	11.8462	17.0423	0.94551
XGBoost	11.3099	16.4655	0.94914

5 Regression Testing

5.1 What is Regression testing

Regression testing is evaluating the performance of our regression model in various ranges of the predicted variable. This gives us insight as to check what ranges our model predicts the delay correctly.

5.2 Results

The model chosen was XGBoost because it had performed better than the other regressors and it's performance within various ranges of Arrival Delay Minutes. The performance of the model was evaluated using the metrics defined in Section 4.2. The Results are as shown below

Table 8 Results for Regression Testing

Range	MAE	RMSE
15 < Minutes < 100	10.103	13.431
100 < Minutes < 200	17.247	26.035
200 < Minutes < 500	17.006	27.2796
500 < Minutes < 1000	17.4596	24.9994
1000 < Minutes	44.8225	103.45

Inference The results are varying with the class $15 < d < 100$ having the lowest values of errors. This can be attributed to the fact that there are many data points within that range. Looking at table 8, we see that our regressor performs well even in the range of 200 and 1000 minutes. Therefore having an RMSE in the range of 16-18 and an MAE in the range of 11-13 is accepted because the values are acceptable compared to the real arrival delay minutes.
From the results in Table 7 we can see that The XGBoost regressor has performed the best with respect to the other regressors, because it has the least MAE and RMSE value and also the highest R^2 value.

6 Pipe-lining

6.1 What is Pipe-lining

In Pipelining we try to feed the outputs of the classifier as an input to the regressor, i.e we predict which of the flights will be delayed and for those flights we predict the number of minutes by which the flight is delayed. To visualize pipe-lining for the problem at hand refer to the figure 2.

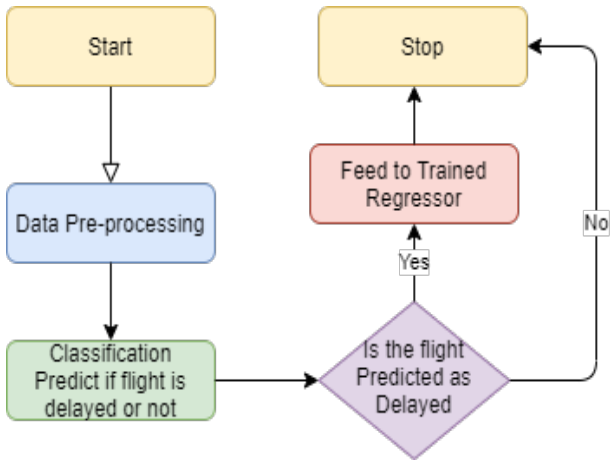

Fig. 4: Flowchart of the Pipe-Line Process .

We use the XGboost Classifier and the XGBoost regressor for the pipeline model, because they have the best performance with respect to the other algorithms.

6.2 Results

The pipeline results will be evaluated by viewing the performance of the final regressor using the metrics defined in Section 4.2.

Table 9 Results for Pipe-lined Model

Metrics	Value
Mean Absolute Error	12.9925
Root Mean Squared Error	17.7564
(R^2)	0.95046

6.3 Inference

The regression model returns a higher Mean Absolute Error and also a higher Root Mean Squared Error with respect to the regression results from Section 4, Table 7. This can be due to the fact that the classifier may have incorrectly classified some of the flights as delayed or not delayed.

7 Conclusion

The data for Flight attributes for the selected airports and also the weather features pertaining to these airports was collected. Both these datasets were processed, then merged to a single data-set that contains the features in interest, for further analysis.
Using the XGBoost classifier the flights were classified as arrived late or on time. The XGBoost Classifier has performed with a Precision of 0.90, Recall of 0.73 and an f1-Score of 0.79 for the flights belonging to class 1 (flights that Arrived late).
A pipeline was performed with the best performing classifier, XGBoost, classifying if the flights would arrive late or on time. For those flights that were predicted to have arrived late, the XGBoost regressor, the comparatively best performing regressor, predicted the number of minutes by which the flight will arrive late with an MAE of 12.99, RMSE of 17.75 and R^2 of 0.95.
Hence a two stage, classification and regression, machine learning engine was designed and built to classify whether if a flight will arrive late or on time and predict the number of minutes by which a flight arrives late.