ON TIME FLIGHT PERFORMANCE PREDICTION

DS-502

Akshay Sadanandan

Arjun Rao

Lyzanne Erika D'souza

Sayali Shelke

Vishaal Prabhakar

Table of Contents

On-time Flight Performance Prediction

Abstract:

Commercial airlines are a backbone of the worldwide transportation system, bringing significant socioeconomic utility by enabling cheaper and easier long-distance travel. After more than half a century of mainstream adoption (especially in the US), airline operations have seen major optimizations, and today function with excellent reliability even in the face of onerous engineering challenges. Still, the modern passenger is occasionally inconvenienced by aircraft delays, disrupting an otherwise exacting system and causing significant inefficiencies at scale in 2007, 23% of US flights were more than 15 minutes late to depart (federal definition of flight delay), levying an aggregate cost of $32.9bn on the US economy. Suboptimal weather conditions were the direct cause of ~17% of those delays, suggesting that better understanding of aircraft unfriendly weather could improve airline scheduling and significantly reduce delays [1].

Our problem deals with flight data from Bureau of Transportation Statistics, USA. We are trying to predict the on- time performance of flights which is a two-stage machine learning problem involving both classification and regression methods. We have used an earlier case study from a previous project [3] as a reference point for our problem statement. The earlier case study only used weather data from the Bureau of Transportation Statistics, USA. In order to optimize and gain better results for predicting the on-time flight prediction, we have taken into consideration two data sets (weather and flight) and have merged them together.

Brief Introduction:

Airline flight delays have come under increased scrutiny lately in the popular press, with the Federal Aviation Administration data revealing that airline on-time performance was at its worst level in 13 years in 2007. Flight delays have been attributed to several causes such as weather conditions, airport congestion, airspace congestion, use of smaller aircraft by airlines, etc. We analyse empirical flight data published by the Bureau of Transportation Statistics to estimate the scheduled on-time arrival probability.

Since June 2003, the airlines that report on-time data also report the causes of delays and cancellations to the Bureau of Transportation Statistics. Reported causes of delay are available from June 2003 to the most recent month. Carriers that have 0.5 percent of total domestic scheduled-service passenger revenue report on-time data and the causes of delay. In 2019, there are 17 carriers reporting these numbers.

The airlines report the causes of delay in broad categories that were created by the Air Carrier On-Time Reporting Advisory Committee. The categories are Air Carrier, National Aviation System, Weather, Late-Arriving Aircraft and Security. The causes of cancellation are the same, except there is no late-arriving aircraft category [2].

How are these categories defined?

- **Air Carrier:** The cause of the cancellation or delay was due to circumstances within the airline's control (e.g. maintenance or crew problems, aircraft cleaning, baggage loading, fuelling, etc.).
- **Extreme Weather:** Significant meteorological conditions (actual or forecasted) that, in the judgment of the carrier, delays or prevents the operation of a flight such as tornado, blizzard or hurricane.
- **National Aviation System (NAS):** Delays and cancellations attributable to the national aviation system that refer to a broad set of conditions, such as non-extreme weather conditions, airport operations, heavy traffic volume, and air traffic control.
- **Late-arriving aircraft:** A previous flight with same aircraft arrived late, causing the present flight to depart late.
- **Security:** Delays or cancellations caused by evacuation of a terminal or concourse, re-boarding of aircraft because of security breach, inoperative screening equipment and/or long lines in excess of 29 minutes at screening areas[2].

What have the airline reports on the causes of delay shown about flight delays?

Delay Cause by Year, as a Percent of Total Delay Minutes :

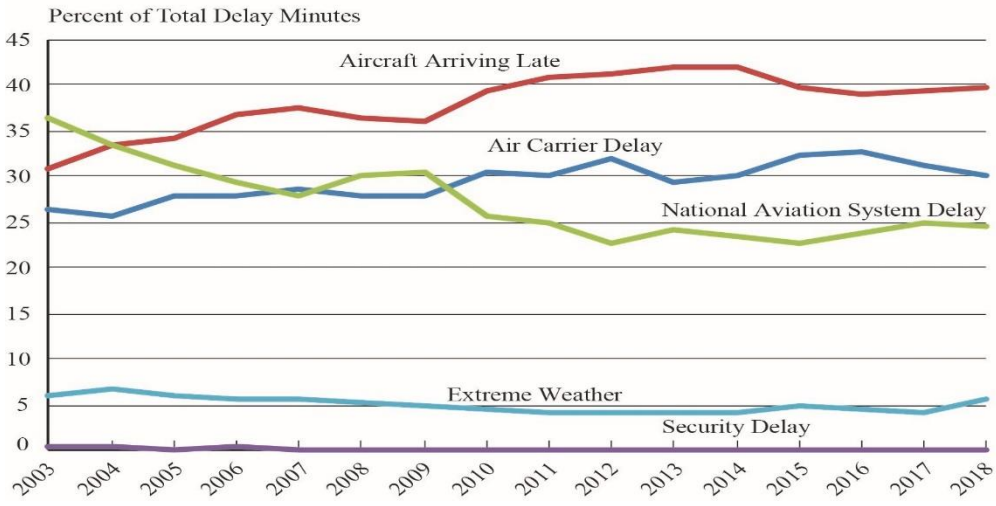

Figure 1.Delay Cause by Year, as a Percent of Total Delay Minutes

Dataset Description:

The project involves flight records from the **Bureau of Transportation Statistics, United States of America**. The number of data instances amounted to almost 200,000 and the number of predictors were 23 in number.

The initial weather dataset consisted of weather records for the five airports under consideration and had almost 45000 instances. The weather data was obtained from multiple sources

including but not limited to **National Center for Environmental Information** and **Weather Base**.

Literature Survey:

We have referred to a problem statement [3] from a previous project, for our further study and experimentation. Our main aim is to optimize and gain better results by using different machine learning and statistical methods.

Their study mainly included them using the transport data of the year 2009 from the U.S Department of transportation and predicting the features affecting the on-time performance of flights.

Data Cleaning:

To mine the weather data, they first filtered the training and test examples for all unique dates and airport locations. The GET requests and BeautifulSoupPython packages were used to download the weather data for each unique date at each airport and parse the data to CSV format. And finally, leveraging Python's shelve library and the pytz package, they matched each flight example to weather data recorded within half an hour of the flight's scheduled departure time and created the following new features: temperature (Fahrenheit) , visibility (miles),wind speed (MPH), precipitation level (binary indicating whether any precipitation presently), and weather conditions (categorical). Weather conditions (such as freezing rain, thunderstorms, and patches of fog) were ranked on a 0−9 scale depending on its impact on flight delays.

Methods Performed and Results:

1) Logistic Regression:
   For initial testing, the MATLAB glmfit library was leveraged to create a logistic regression model. A Bayesian regularized logistic regression model with Newton's method optimization was also created. Cross validation was used to optimize parameters for the regularized MAP estimate. Logistic Regression achieved the best F1 score of 0.5994.

2) Naïve Bayes:
   For classification with Naive Bayes, the supervised learning methods from MATLAB's Statistics Toolbox were used. Initially, the Naive Bayes classifier, was attempted to fit a Gaussian distribution to all the features. However, for some training samples (mainly small sample sizes), the variance of certain features was zero (which resulted in a degenerate normal fit). This signalled that some of the features may not necessarily follow a normal distribution.
   14 of the features were continuous whereas the other 89 features (created from the non-numerical data for weather conditions, airline carrier, airport origin, airport destination, and previous flight delay) were categorical. Then, in order to create a Naive Bayes model that takes into account both continuous and categorical data, they considered the following methods:(1)independently fit a Gaussian Naive Bayes model on only the continuous part of the data and fit a multivariate multinomial Naive Bayes model on

only the categorical part of the data; transform the entire dataset by taking the class assignment probabilities as new features; and fit a new Gaussian Naive Bayes model on these new feature, or (2)transform all continuous features into a categorical representation by binning. At first, they attempted the former method by using different distributions to independently fit the data. However, discovered vastly different results from the two different distributions.

From the learning curve of a Naive Bayes model using a Gaussian distribution, it can be seen that the training error is unacceptably high and there is a small gap between training and testing error, which is indicative of high bias. After performing Naïve Bayes on Gaussian Distribution (only continuous features), Multivariate Multinomial(only categorical features) and Multivariate Multinomial (all features), the results have that the multivariate multi-nomial Naive Bayes model (both subset and entire set of features) performed the best (and about equal), achieving an 83% accuracy and a F1score of ~0.89, indicating both high precision and high recall.

3) SVM:
   For SVM, Python library scikit-learn were used, which wraps liblinear and libsvm. After plotting bias-variance learning curves for both Gaussian(rbf) kernel and linear kernel, the results showed that linear kernel generally performed better.

4) Multiclass Classification:
   Late class was split into two classes: class 1 between 15 and 45 minutes late, and class 2 for > 45 minutes late. An SVM with linear kernel and one vs. all strategy almost never predicted class 1, so they used an SVM with a Gaussian kernel and one-against-one strategy (create one SVM per pair of classes, and to predict, the class which receives the most votes is selected). Overall accuracy was 0.8167; recall for class 1 was only 0.15, but recall for class 2 was higher at 0.60. This is because our classifier predicted class 2 57% more often than class 1, despite the fact that more flights are actually class 1. This suggests that class 1 flights (between 15 and 45 minutes late) do not have strong distinguishing characteristics in the dataset, compared to class 2 flights.

5) Random Forests:
   After performing parameter optimization on the number of trees in the forest, and the size of the random subsets of features considered when splitting a node, it was found that a random forest classifier with 100 trees considering 100 features had 0.8917 accuracy, 0.93 precision, and 0.44 recall.

   Overall, accuracy of all algorithms was relatively good: all algorithms were about 90% accurate. The multiclass classification (>15 min & >45 min) showed that flights between 15 and 45 minutes late do not have strong distinguishing characteristics in the dataset, so better features might improve recall.

Block Diagram:

We start off with two datasets (Flight Data and Weather Data) and merge them based on place, arrival time and destination time. After merging, we perform classification for checking if a flight is delayed and perform regression to see by how much time would the flight be delayed (arrival time).

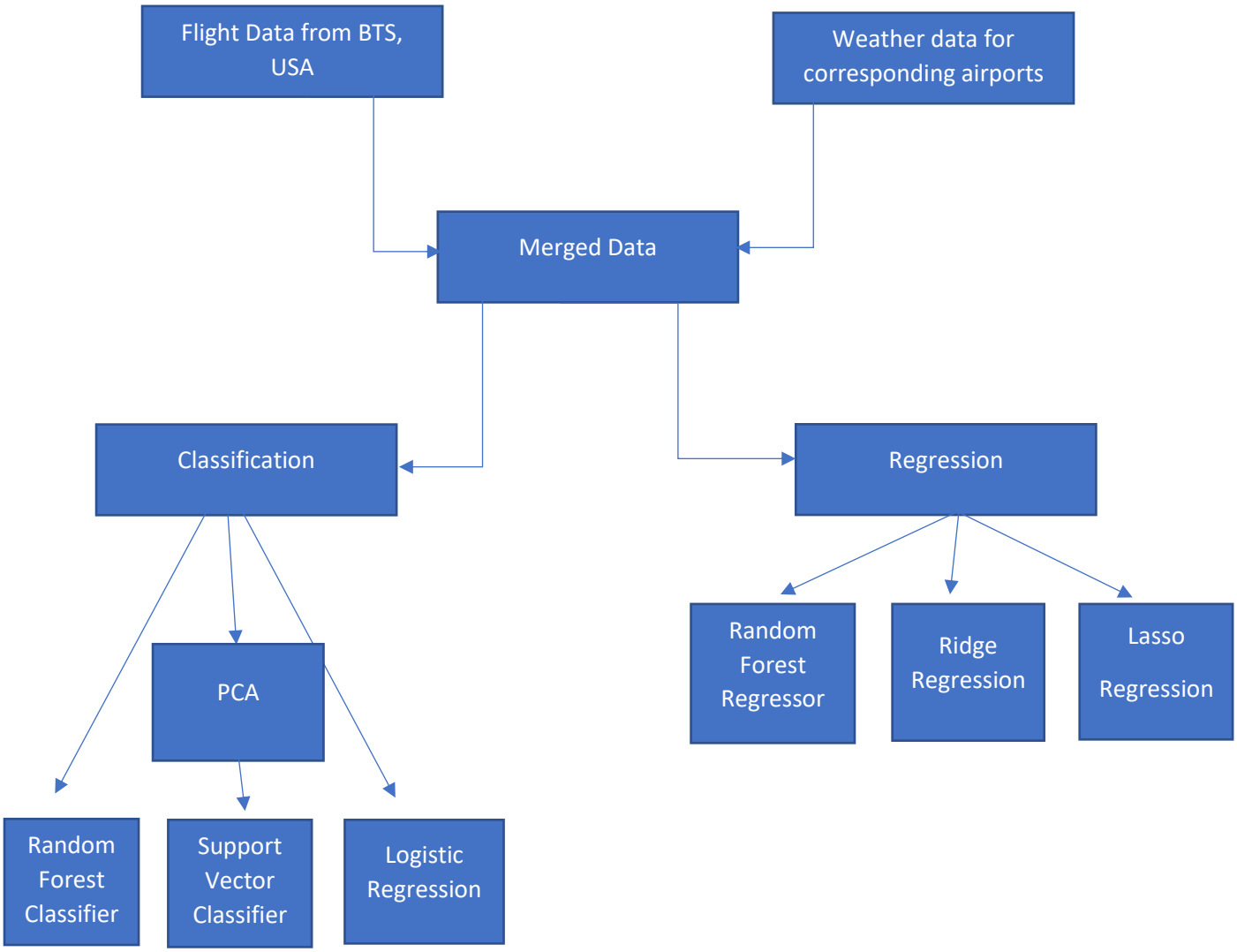

Figure 2.Flow chart of the model.

Methodology:

1) **Cleaning and preprocessing the data:**

We created the timestamp for the weather and flight data that we imported from the Bureau of Transportation Statistics, USA.Then we merged the two data frames based on the place and timestamp. Label encoding was done to convert categorical variables into numerical values. Different techniques were used to treat missing values by taking mean, median and filling them with zeros.

- The raw flight data available was filtered to obtain flight records for the year 2014.
- The weather data frame was filtered to obtain weather parameters for 2014 at the airport areas under consideration.
- Initially, a timestamp column was created on both the flight and weather data frames by a combination of date and time columns.
- Now, the two data frames are merged, based on the combination of location and timestamp columns.

2) **Target Column Creation:**

The target column is obtained by comparing the actual arrival time and scheduled arrival time.

- Classification target column
    - If the actual arrival time is greater than the scheduled arrival time, we assign the value to 1 indicating that the flight is delayed.
    - On the other hand, if the flight is on time or early, we set the target column to 0 for that instance
- Regression target column
    - The target column is obtained by subtracting the scheduled arrival time from actual arrival time giving us the time time by which the flight is delayed.
    - On the other hand, if the flight is on time or early, we set the value of the target column to 0 for that instance.

3) **Algorithms performed on the dataset:**

a) Random Forest Classifier:

A random forest classifier is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. We are predicting if the flight is delayed or not using Random forest Classifier.

Observations and results:

n_estimators	Max_leaf_node	R^2	Accuracy	F1 score	AUC (Area under curve)
20	default	0.9997	0.99979	0.9997	0.9999
50	5	0.996	0.99635	0.9955	0.9999
10	3	0.972	0.97269	0.9659	0.9999

Table 1.Random Forest Classifier Results.

In table 1, we used different n_estimators and max_leaf_nodes values and evaluated the metrics based on the models. We observed that full depth tree is giving the best results. And as we go on decreasing the number of nodes, the classification results start decreasing slightly.

Figure 3 is a tree which is pulled out from our random Forest Classifier model.

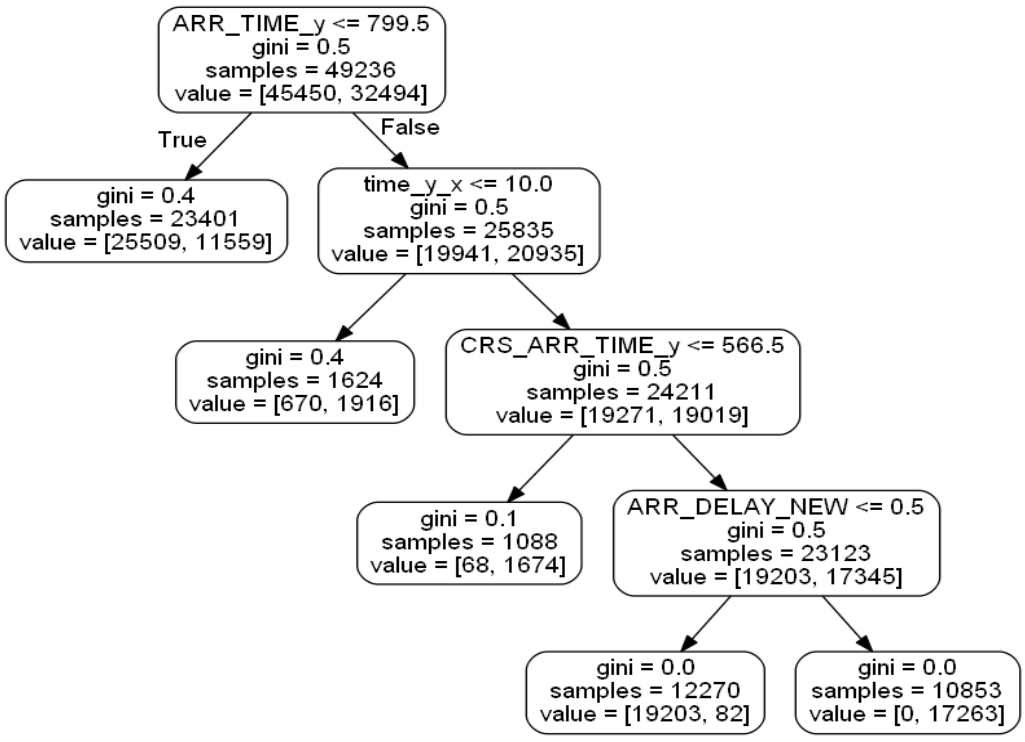

Figure 3.A tree which is pulled out from our random Forest Classifier model.

b) Random Forest Regressor:

In order to enhance the one-dimensional nature of binary classification, we attempted to extend the problem statement by not only predicting the occurrence of flight delays, but also its numerical value. We did this using the Random forest Regressor.

Observations and Results:

n_estimators	Max_leaf_node	R^2	RMSE	MAE (Mean Absolute Error)
20	default	0.9895	6.2704	0.4455
50	5	0.81807	26.16	11.87
20	3	0.6398	36.81	15

Table 2.Regression metrics

In table 2, we varied the n_estimators and the max_leaf node value and the corresponding RMSE, r^2 and adjusted r^2 values were observed. It must be noted that there was not much difference between the r^2 and adjusted r^2 values in the test cases used here. RMSE goes on decreasing as we decrease the max leaf nodes.

Figure 4 is a tree which is pulled out from our random Forest Regressor model.

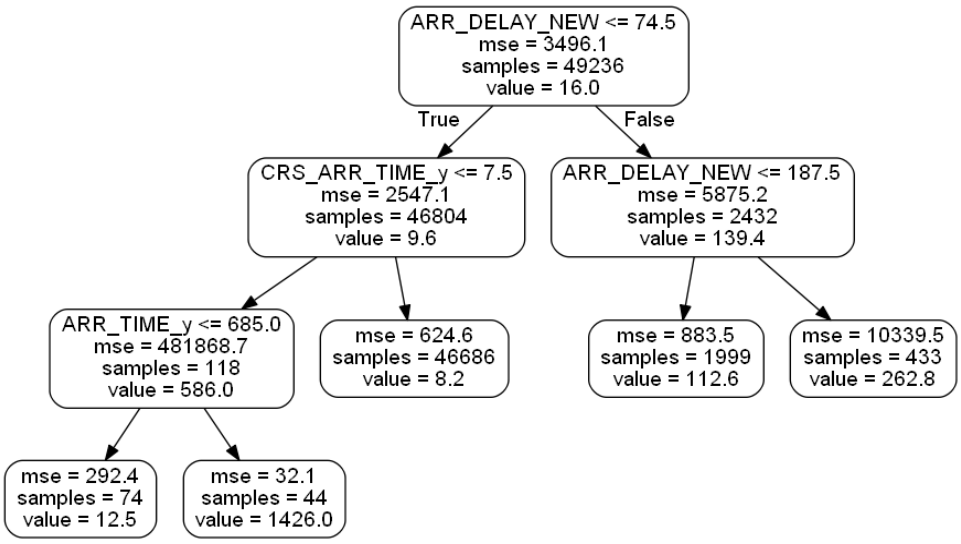

Figure 4.A tree which is pulled out from our random Forest Regressor model

We observed that the RF with full trees has a much lower error on train data than the RF with pruned trees but the error on test data is higher. Let's visualize this on the scatter plot.

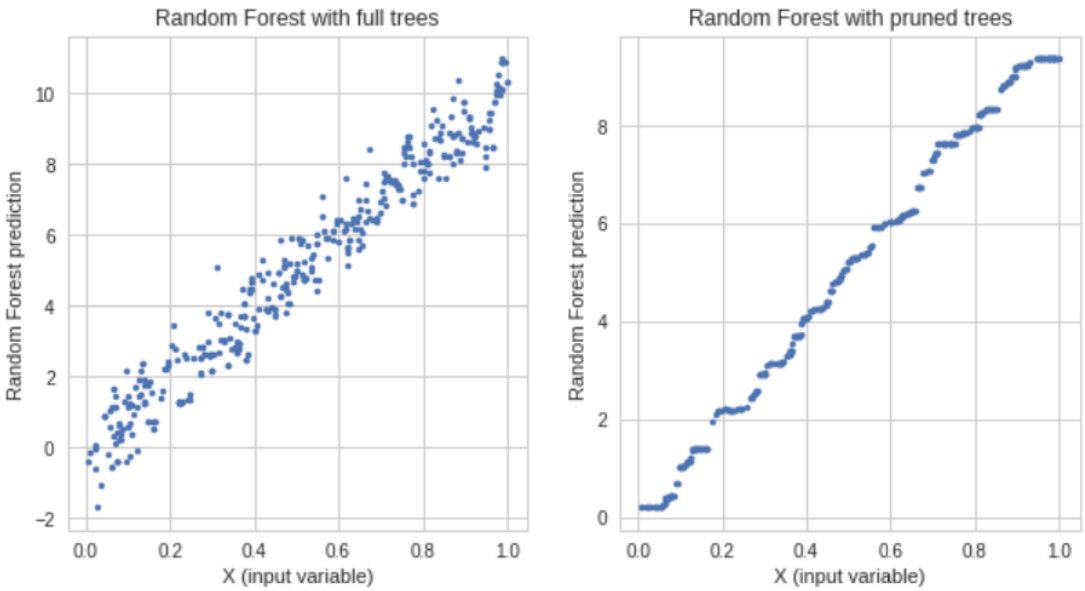

Figure 5.Comparison of Fully-grown RF trees and Pruned RF trees

In Figure 5, on the left, there is a response from overfitted Random Forest and on the right the response of the Random Forest with pruned trees. We see the RF with full trees, which overfitted, predicts a noise which it learns during the training. The response from the RF with pruned trees is much smoother. Hence, we can safely say that RF perform better with full depth trees.

c) Support Vector Classifier:

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.

Approach:
There are three factors gamma, budget and kernel that are taken into consideration. In our case, since the dataset was huge, we performed sampling in order to choose which kernel fits our model the best. After performing sampling on the dataset, RBF kernel was the best kernel picked for this model. We worked on different budgets (1, 10, 50 100).
Since the number of predictors was 84, the computation time for a support vector classifier was large. To tackle this problem, we implemented **principal Component analysis**.

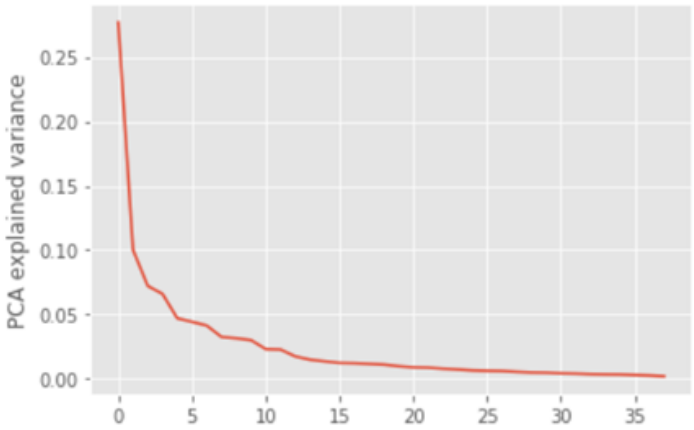

Figure 6.PCA Components curve to choose number of features.

After performing PCA we decided to take the linear combination of 13 components into consideration. The 13 components took into account 80% of the variance.
By doing this, our computation time reduced drastically to 40 minutes per training session for a support vector model.

[220]:

principal component 1	principal component 2	principal component 3	principal component 4	principal component 5	principal component 6	principal component 7	principal component 8	principal component 9	principal component 10	principal component 11	principal component 12	principl compone
3.926144	1.322982	0.400486	-4.646717	3.438677	-0.881934	-1.254468	-1.289295	0.394208	2.319808	-2.195412	-1.225915	2.1524
3.502730	2.996086	-1.814035	-1.762718	2.742628	-1.984238	-1.599268	0.898856	2.021691	1.148601	-0.977153	0.431206	0.3662
3.695668	-0.753318	0.663203	-3.267756	0.345635	-1.108897	1.724806	-0.687978	-0.257800	-0.604114	0.122871	-0.176525	2.7763
-1.634707	-0.642343	4.406584	0.098287	-1.199735	-4.439268	-3.583514	1.088084	-1.078545	-1.872381	2.643702	-0.156975	-1.0113
4.938502	-2.008015	2.379901	0.735392	-2.875588	5.796849	-1.722324	-0.533487	-0.673071	1.547884	0.901260	-2.302763	0.0525
...	...	...	...	...	...	...	...	...	...	...	...	
4.111170	5.004689	-0.966218	-0.811170	0.472301	-1.765846	-1.673306	-1.558891	1.333595	-0.586430	-0.598975	-1.830731	-1.6120
9.446037	4.842144	1.113984	-0.978427	-1.152779	-0.971068	-0.974589	-3.786928	1.338085	1.979646	0.599995	-0.015407	-0.9722
-4.164645	-2.194852	3.039465	-1.428655	0.943453	0.570286	-0.678643	1.459211	0.586510	1.446377	1.848606	0.188924	0.8919
0.822591	1.449186	-2.733065	1.171446	0.240161	1.338471	-1.847708	-1.270633	-0.339831	0.560963	-1.076834	3.440664	2.4205
-3.892253	1.222765	3.684904	-0.265187	2.549770	-0.043778	1.468240	1.076675	-0.586608	3.522247	-2.389120	-1.130551	-1.2837

Table 3.Linear combination of 13 components after PCA.

Now that the computation time is adjusted, we work on the hyperparameters of our model.

Kernel:
There are a couple of kernels when it comes to performing support vector machines, such as, linear, rbf and poly kernels. We trained our model on each kernel to check which kernel would give us the best results. The metrics taken into consideration was the F1 score.
We chose to perform on the 'rbf' kernel since the linear kernel is similar to logistic regression which we have performed later. We did this to explore the performance
Budget:
This is hyperparameter in support vector machines which allows a certain number of points in our data to be misclassified. This can be done to prevent outliers from affecting our model and also to prevent overfitting of our model.

We took into consideration budgets of 1,10,50,100 and found that as our model performed best at a budget of 100. We should keep in mind that the budget should not be too high or else all the models will misclassify our data.

With Randomised dataset	**Performing K fold for budget =100**
Budget:100, F1: 0.96 Budget:50, F1: 0.79 Budget:10, F1:0.76 Budget:1, F1:0.67	F1 score: 0.81 best cross validation score

Table 4.The comparisons of various budgets in the data set.

From the table above, we can see that our model performed best when we took a budget of 100. Performing cross validation on the entire data set we got a F1 score of 0.81.

Gamma:

The definition of gamma is 'the gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors.

A small gamma value defines a Gaussian function with a large variance. The model is too constrained and cannot capture the complexity or "shape" of the data.

A large gamma value means defines a Gaussian function with a small variance and in this case. The radius of the area of influence of the support vectors only includes the support vector itself and no amount of regularization with 'C(budget)' will be able to prevent overfitting.
We implemented for 3-4 values of gamma since this parameter was relatively new to us.

We worked with gamma values of 50,5,1, scale to check which gamma parameter would suit our model best. Four our model, it was gamma with 1 which gave us the best test result. But then we decided to add gamma to scale in order to find the best fit for itself.

With C=100

Gamma= 5 on test data	F1 = 0.00024
Gamma= 50 on test data	F1 = 0.425
Gamma=scale on test data	F1 = 0.783
Gamma= scale on training data	F1 = 0.94

Table 5. Performance of gamma on the dataset.

d) Logistic Regression:

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression is estimating the parameters of a logistic model. Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labelled "0" and "1". Logistic regression is used to describe data and to explain the relationship between one

dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

Approach:
In our model, we performed logistic regression on the entire dataset. Our results were good, in fact, better than the support vector classifier. This gives us a strong reason to believe that **our data is linear**.
We performed 5-fold cross validation on the model to obtain the following results,

F1 scores for each cross validated model	Accuracy
0.905	0.919
0.907	0.918
0.890	0.901
0.906	0.918

Table 6. F1 score for each model created during K-fold cross validation

We obtain good results on the test data too indicating that our model performed well.

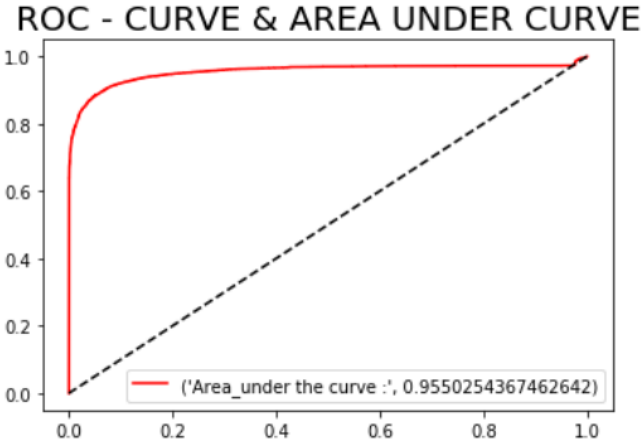

Figure 7. The ROC curve for logistic regression with area under the curve = 0.955

e) Ridge Regressor:

Ridge Regression is a technique for analysing multiple regression data that suffer from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. It is hoped that the net effect will be to give estimates that are more reliable.

Since the dataset had around 80 predictors, our idea was that a shrinkage method such as this should produce good results.

As expected, the model performed well on the dataset but we ran into an odd situation. The model gave a few negative values. This was surprising because none of the values in the target column were negative. An explanation to this might be that since our data for the target column ranges from 0 to ~1500, with a majority of these values being zero, the model is trying to predict values closer to zero and sometimes predicts negative values as a consequence.

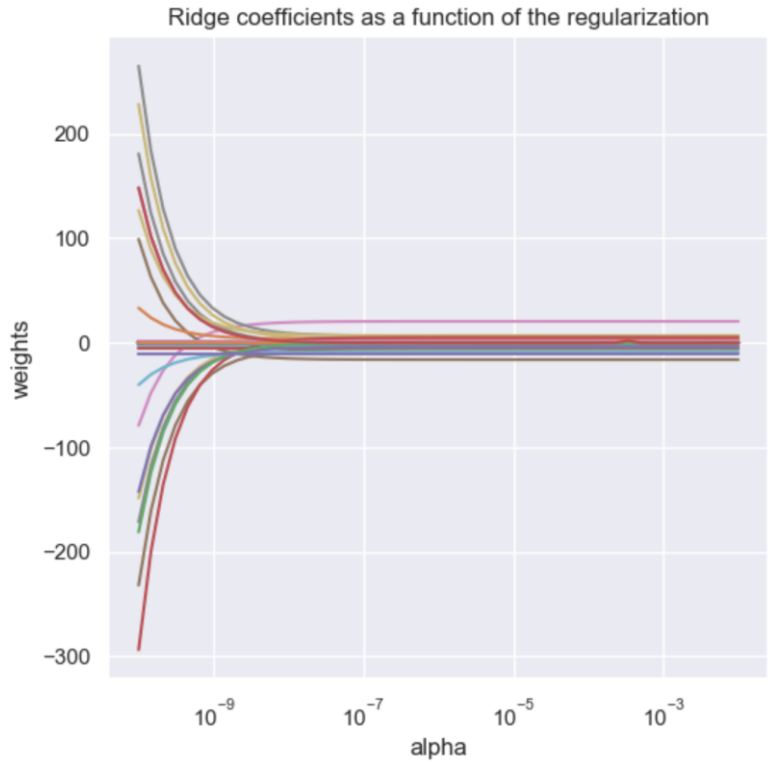

Figure 8. Ridge coefficients as a function of regularization

Figure 8 depicts the plot of Ridge coefficients as a function of regularization. Here, each colored line represents a different coefficient. When alpha is very large, the regularization effect dominates the squared loss function and the coefficients tend to zero. At the end of the path, as alpha tends toward zero and the solution tends towards the ordinary least squares, coefficients exhibit big oscillations. In practise it is necessary to tune alpha in such a way that a balance is maintained between both.

	Adjusted R^2	**MSE**	**RMSE**
Without k-fold CV	0.934896172	238.1883576	15.43335211
With k-fold CV	0.941732173	230.8236845	15.19288269

Table 7.Ridge regression metrics

f) Lasso Regressor:

Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

We expected Lasso to perform well on this dataset, which it did. However, we ran into the same problem we had with Ridge Regressor, where some of the predicted values were negative. This is due to the large disparity between number of delayed flights and flights that are on time.

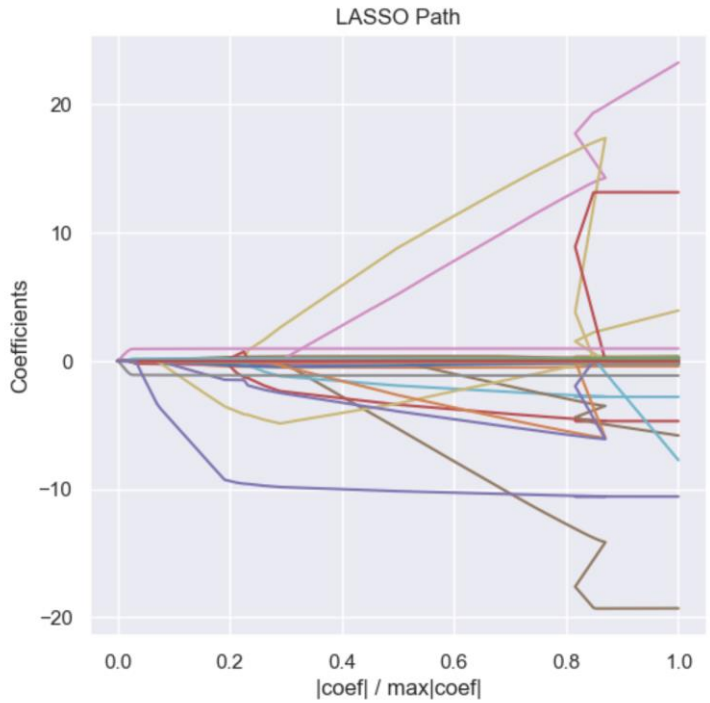

Figure 9. Lasso path

In figure 9, each colored line represents a different coefficient of the Lasso Regression model. Lambda is the weight given to the regularization term (the L1 norm), so as lambda approaches zero, the loss function of your model approaches the OLS loss function. Therefore, when lambda is very small, the LASSO solution should be very close to the OLS solution, and all of your coefficients are in the model. As lambda grows, the regularization term has greater effect and fewer features will be present in the model (because more and more coefficients will be zero valued).

	Adjusted R^2	**MSE**	**RMSE**
Without k-fold CV	0.9245667943	270.71246212	16.453341974
With k-fold CV	0.9337758274	260.65744758	16.144889209

Table 8.Lasso Metrics

4) Challenges:

- Timestamp creation was tricky because of the different date-time formats involved in the two data frames.
- Complexity of the datasets hindered the merging.
- Large number of missing values required extensive treatment.
- Encoding type selection required extensive research.
- Random Forest Classifier required a lot of parameter tuning.
- The type of kernel to be used in our SVM model required a lot of study. Computational time for the polynomial kernel and RBF kernel was large. Experimenting with different values of budget was difficult because of the large computational time involved.

5) Conclusion:

- Since logistic regression is giving better results than the rbf kernel for SVM, and lasso model performed well with alpha at a low value, we can confidently say that our data is linear.
- Ridge and lasso performed well on the dataset giving 0.94 and 0.93 adjusted R^2 respectively. However, some of the predicted values were negative indicating that there might be a large disparity between number of delayed flights and on-time flights.
- Our classification models take into account the F1 score as a metric to tackle the issue for data imbalance.
- Our regression models take into account the root mean squared error(RMSE) as a metric.
- Most of the algorithms performed really well i.e. accuracy > 90%
- From our observations, the Random Forest Classifier is the one that gives the best F1 score in case of classification.
- The random forest regressor model is the one that gives us the lowest RMSE among other regression models. Random forest models are known to deal with data in case there is an imbalance in the data

6) References:

1. http://cs229.stanford.edu/proj2016/report/MenonMovva-PredictingFlightDelays-report.pdf
2. https://www.bts.gov/topics/airlines-and-airports/understanding-reporting-causes-flight-delays-and-cancellations
3. http://cs229.stanford.edu/proj2013/MathurNagaoNg-PredictingFlightOnTimePerformance.pdf
4. https://www.wikipedia.org/
5. https://scikit-learn.org/stable/modules