Machine Learning Engineer Nanodegree Capstone Project Predicting flight delays using supervised learning

Sirichandana Sambatur

Introduction:

Air-travel is very popular and many people choose air-travel because it is the fastest form of travel and people don't want to spend more amount of time traveling. But many airports and airlines are frequently prone to flight delays. These delays could be because of several reasons but predicting such delays ahead of time could help people plan their travel in an efficient manner. As there are several reasons that could cause a flight delay, manually assessing these reasons to predict a flight delay will be a tedious process. Machine Learning and historical data can be leveraged to build predictive models that predict flight delays[1]. One of the previous works in this domain analyzed the flight delays in Pittsburgh.[1]. In this analysis, some of the main results obtained were that more delays were occurred on Fridays and Saturdays. It was also found that after the year 2006, the delays in flights decreased and the main reason behind this was that the number of flights flown to Pittsburgh were reduced. In this project, it is expected to produce such similar results related to different carriers and airlines.

Problem Statement:

The objective of this project is to build an efficient and generalized model that predicts flight delays. Machine Learning algorithms have the ability of understanding and detecting patterns in the historical data. Some of the machine learning algorithms such as Support Vector Machines, Logistic Regression, Decision Trees etc. can be used to learn the patterns of flight delays and build an effective predictive model. The project considers this problem as a classification problem and builds an identifier that predicts whether the departure time of the flight will be 'on-time' or 'delayed'. The input for these models will be a data instance that represents the information about a flight. This information includes airline carrier, origin airport, scheduled departure time, destination airport, scheduled arrival time etc. Provided an unlabelled instance which represents a flight, the trained models will identify if the flight will depart on-time or not.

Evaluation Metrics:

The dataset will be split into training and testing datasets. Various models will be trained on the training dataset and will be tested on the testing dataset. The dataset is very imbalanced and

contains a greater number of instances which represent the flight being on-time. Because of this imbalance in the class distribution, accuracy is not a good evaluation metric and different evaluation metrics such as precision, recall, and area under ROC will be computed to compare the performance of different models. Precision is defined by the exactness of the results and quality of results that are obtained. On the other hand, recall measures the proportion of positives that are correctly identified. When both the precision and recall are high this indicates that the model is performing well. Also, for good models the area under the ROC tends to be greater than 0.5. So, by using precision, recall and area under ROC, the quality and performance of the model can be assessed.

Analysis

Data Exploration:

The dataset that will be used for this project is gathered by The U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics[2]. The dataset contains summary information of flights for the year 2015. The dataset contains 5,819,079 instances and the dataset doesn't contain any class labels by default. This data set consists of 31 features that represent the data. As there are many features, only the relevant ones will be considered for the model. The dataset represents information such as the airline carrier, scheduled departure time, actual departure time, origin airport, destination airport etc. This data will be provided as input for training the machine learning algorithms. Flights can be categorized as "on time" or "delayed" based on the "departure_delay" feature. The following figures give us some insights on how the dataset that is being used for model training exists-

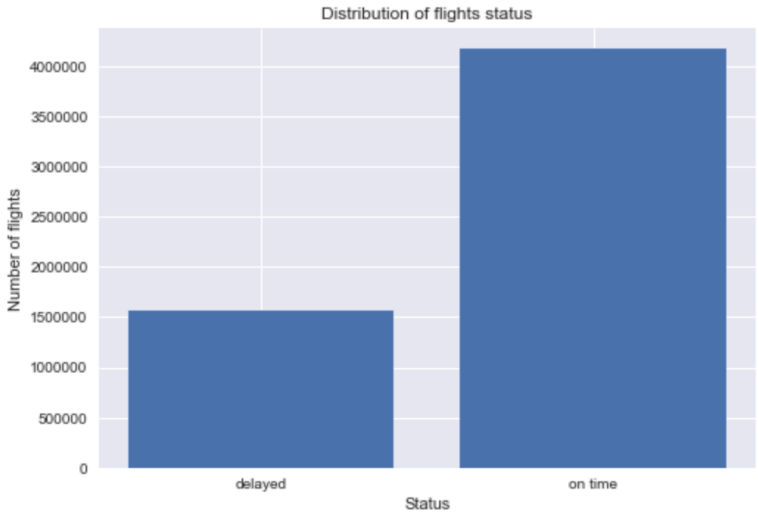

Figure 1- Distribution of flight status.

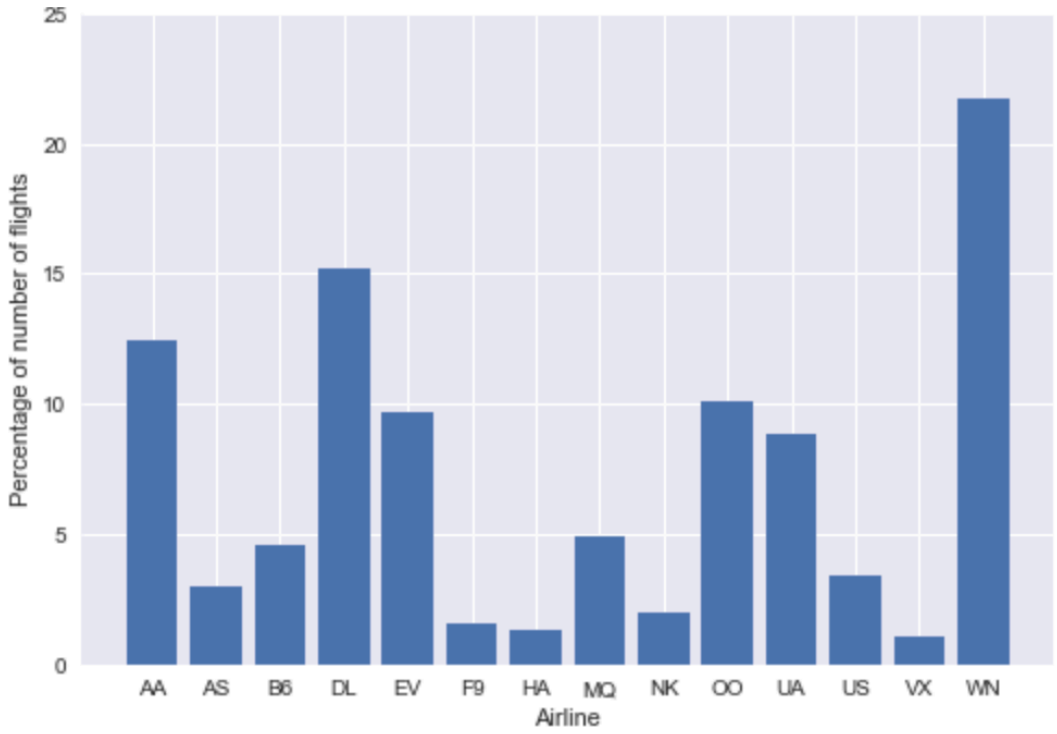

Figure 2- Percentage of number of flights across different airlines.

This visualization shows that there is an imbalance in the flights that are on-time and delayed. Due to this imbalance, a downsampling preprocessing technique will be applied so that the model that is developed will not be biased.

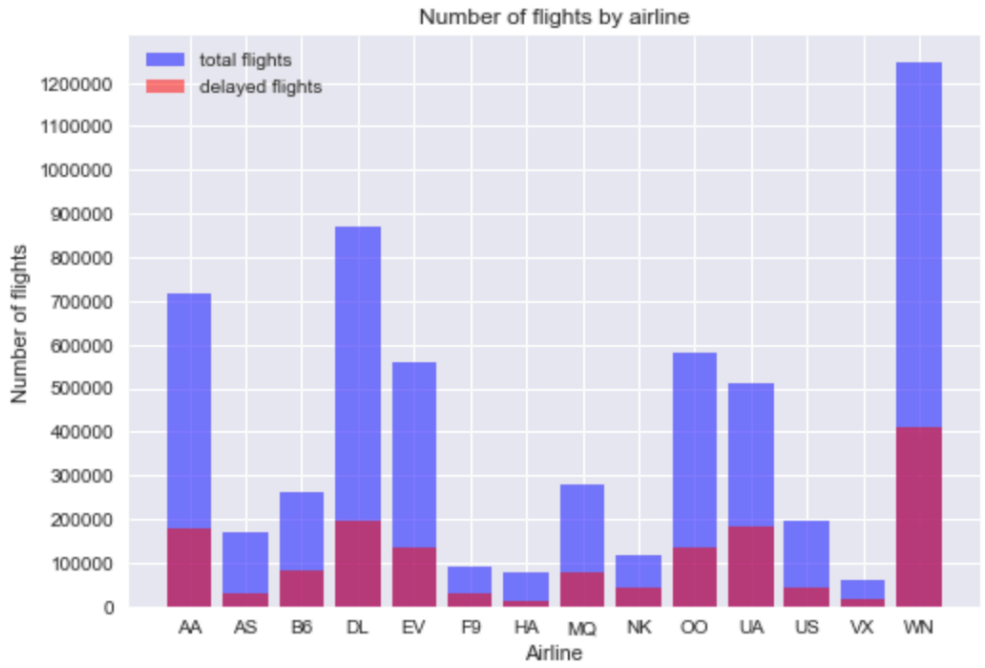

Figure 3- Status of flights across the different airlines.

From the above graph, it can be seen that among the 14 different airlines present in the data set, 23% of the flights belong to Southwest Air Lines, 15% to Delta Air Lines and 12.5% belong to American Air Lines.

The above visualization helps us understand the delays in flight across the various airlines in the data set. The percentage of delays varies across different airlines, it ranges from 15%-45%.

	DAY_OF_WEEK	COUNT
0	1	845287
1	2	829990
2	3	845583
3	4	860780
4	5	853884
5	6	692252
6	7	805144

Figure 4- Flights operated based on the days of a week.

From this visualization, the flight operations can be analyzed on different days of the week. It will help us understand if particular days of the week are busier compared to others and these days would lead to more flight delays. So, from the table it can be said that most days operate the same number of flights, however, on Saturdays the number of flights is less when compared to the other days.

Figure 5- Status of flights across the days of a week.

From this above visualization, it can be seen that the days of the week don't seem to be making a much different on the delays of the flights. The delays in flights are mostly uniform across the different days of the week.

Feature	Datatype
Year	Numeric
Month	Numeric
Day	Numeric
Day_Of_Week	Numeric
Airline	Categorical
Origin_Airport	Categorical
Destination_Airport	Categorical
Scheduled_Departure	Numeric
Departure_Time	Numeric
Departure_Delay	Numeric
Status	Categorical

Figure 6- Features of the data set

The table above lists the various features that are available in the dataset and will be used by the model. Also, some preprocessing techniques will be applied to these features so that they will suitable for the model.

Algorithms and Techniques:

As the data does not contain any class labels by default, the class labels will be added during the data pre-processing stage. Exploratory Data Analysis, Feature Selection or Principal Component Analysis will be used to determine the most important features. By using feature selection methods like PCA or SelectKBest, the features that accurately determine the delays in flights can be used and unimportant or unnecessary features could be discarded. Multiple classification algorithms such as Gaussian Naive Bayes, Decision Trees, Random Forests, Logistic Regression etc. will be trained and tested to identify the best model. Both Decision trees and Random Forests are tree like models that decide on class labels based on the feature values. As our feature set will be limited after feature selection, both these Machine learning techniques could yield good results. These tree based classifiers work well on categorical data and by tuning some of

their parameters such as ‘min_samples_split’, ‘criterion’, ‘num_estimators’, ‘max_depth’ etc efficient models can be trained. ‘Min_samples_split’ specifies the minimum number of samples required to split an internal node and the ‘criterion’ is the measure for accessing the quality of split. The different criterions are ‘gini’ and ‘entropy’. Also, ‘max-depth’ defines the maximum depth of the tree. One of the most important parameters for the Random Forest Classifier is the ‘num_estimators’. This parameter indicates the number of decision trees to be used in the forest. By tuning these different parameters, a desired model for our dataset could be obtained. Tuning of parameters is important as this will help in preventing over fitting or under fitting of the data set. under fitting means that the model doesn’t correctly analyze the data set and it is very simple. However, overfitting means that the model is too complicated for the model and over generalizes it.

Logistic Regression works for categorical target labels like in our dataset. It does so by using a Logistic function, which is a sigmoid curve and given by the following equation-

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

where

- e = the natural logarithm base (also known as Euler's number),
- x_0 = the x-value of the sigmoid's midpoint,
- L = the curve's maximum value, and
- k = the steepness of the curve.[1]

Some of the parameters that can be tuned for a Logistic Regression classifier are the kind of solver to be used, the maximum number of iterations to be considered by the solver to converge and the method of penalization. The different types of solvers are ‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, ‘saga’. On the other hand, Gaussian Naive Bayes is based on the Bayes theorem which assumes the independence of the features. Also, in our data set the features are independent, so, Gaussian Naive Bayes could yield desired results. These different ML techniques have different approaches to predicting the target label and they are suitable for the data set and hence, it would be important to compare the results of these techniques and find the one that suits the data set best.

Benchmark Model:

A ZeroR Rule can be used as a benchmark model for this project. A ZeroR Rule does not consider any attributes or features. It uses prior probability and predicts the class label with the higher probability for all the instances. The accuracy of the ZeroR rule is 50.015%. Also, basic

Gaussian Naive Bayes and Decision Trees without any parameter tuning gives the results as follows-

Technique	Accuracy	Precision	Recall	Area under curve
Gaussian Naive Bayes	58.64%	0.509	0.26	0.5809
Decision Tree	52.833%	0.458	0.706	0.5535

Methodology

Data Preprocessing:

Different preprocessing techniques are applied to the dataset before it is used to build the model. From the list of features (31) in the model, the features which are not related to the model like 'FLIGHT_NUMBER', 'TAIL_NUMBER', 'TAXI_OUT', 'WHEELS_OFF', 'AIR_TIME', ‘WHEELS_ON', ‘TAXI_IN', 'AIR_SYSTEM_DELAY', ‘SECURITY_DELAY', 'AIRLINE_DELAY', ‘LATE_AIRCRAFT_DELAY', 'WEATHER_DELAY', 'DIVERTED', 'CANCELLED', ‘CANCELLATION_REASON','ELAPSED_TIME', ‘ARRIVAL_TIME', ‘ARRIVAL_DELAY' are being removed. Most of these features indicate the reason for our delay and are not suitable for building the model. Also, the features ‘FLIGHT_NUMBER', and ‘TAIL_NUMBER' are unique identifiers which will not be useful in building the model.

The next preprocessing step is to remove rows which have “DEPARTURE_DELAY”, “DEPARTURE_TIME” and “SCHEDULE_TIME” as null. There 86000 which are removed from the dataset. Apart from this, the ‘SCHEDULED_DEPARTURE' and ‘DEPARTURE_TIME’ contains values of time which is not properly formatted. When this value is given to the model it will not be able to process it. So, the time is formatted by converting the floating point to integer and then it is formatted as hour blocks.

The dataset lacks the target variable and hence, a target variable called “STATUS” has to be created which takes two values “on time” and “delayed”. These values are assigned to the flights based on the “DEPARTURE_DELAY” feature, if the delay is greater than 5 mins then the flight is “delayed”, otherwise the flight is “on time”.

Since, the data consists of a large number of flights, for the model sampling of the dataset would be an ideal technique. If we wear to sample data randomly then it would consider different flights from different months and this could reduce the precision. So, for the purpose of the model, the flights in the months January and February are being considered and the model is built on this data. Also, as observed in the data exploration section, the count of on-time flights seems to be greater than delayed flights and this could affect our model. So, to remove this bias,

downsampling techniques have been used on the chosen data from the dataset. When downsampling technique is applied, the sampling rate of the majority class is reduced. After applying the down sampling technique using the resample module from sklearn.utils package, the total number of flights are 527836. Out of which 264000 are on time flights and 263836 are delayed flights.

The final preprocessing step applied to the data is to convert the categorical data features. The features "DAY_OF_WEEK","AIRLINE","ORIGIN_AIRPORT","DESTINATION_AIRPORT" had to be converted and for this hot encoding technique is used. By hot encoding, the features are converted into a binary format and there is no ordinality among the features. After applying the one hot encoding technique on the data set, the number of features increased from 11 to 658.

After applying the above preprocessing techniques, the data is ready to build the model/ implement various machine learning techniques on the data.

Implementation:

The dataset contains 5,819,079 instances and 31 features. Some of these features like 'FLIGHT_NUMBER', 'TAIL_NUMBER', 'TAXI_OUT', 'AIR_TIME' etc represented unique identifiers for the flight and some irrelevant information. All such features have been removed from the dataset. Predicting flight delays is a binary classification problem and the data contains many categorical variables. For such categorical data, classifiers such as Decision Trees, Random Forest and Logistic Regression work well.

As the dataset is very big all the classifiers have been trained only on a subset of data which represent instances for the months of January and February. This subset of the data had imbalanced class distribution with 'on-time' flights representing the majority class. To handle this, the majority class of the dataset has been downsampled to make the class distribution balanced. The dataset contains many categorical variables and to allow the machine learning models to learn relevant information, all the categorical variables have been transformed to boolean binary variables.

The subset of the data has been split into training and testing sets to train and test the classifiers. To avoid look-ahead bias in the time series data, the dataset has not been split randomly. To keep the time-series relevant the information of the flights from January 1, 2015, to February 15, 2015, have been included in the training set and the instances which represent the flights' information from February 16, 2015 to February 28, 2015, have been included in the testing set.

To build a benchmark and baseline models, initially, a ZeroR Rule classifier has been applied on the dataset which had an accuracy of 50% and this is equivalent to guessing. Further,

simple Gaussian Naive Bayes and Decision Tree Classifier have also been trained on the data. The Gaussian Naive Bayes classifier had an accuracy of 58.6% and the area under the ROC curve is 0.58. The Decision Tree classifier trained with '*splitting criteria'* as '*gini'* and a *'min_samples_split'* of '2' had an accuracy of 52.9% and the area under the ROC curve is 0.55.

To improve the performance of the models, feature selection has been used to identify the most important features and also to reduce the size of the feature space. The *'SelectKBest'* module available in scikit-learn has been used for this purpose. To identify the best value for 'k' the Gaussian Naive Bayes classifier and the Decision Tree classifier with a '*min_samples_split*' of 20 and *'gini'* criterion have been trained and tested for multiple values of k. Analyzing the performance of the classifiers, the best value for k has been selected as 50. The feature space has then been reduced to 50 for further modeling.

On this reduced feature space, Random Forest classifiers with multiple parameter settings have been trained and tested. A model with 200 estimators, *'entropy'* splitting criterion and min_samples_split of 40 has provided the best results with an accuracy of 58.85% and an area under ROC of 0.67. A Logistic Regression model with 'l2' penalty and a 'liblinear' solver has also been built which had an accuracy of 46.9% and an area under ROC of 0.64. The maximum number of iterations used by the solver to converge is set at 100. The process of refining the classifiers and identifying the best parameters has been explained in detail in the next section.

Refinement:

The simple Gaussian Naive Bayes and the Decision Tree classifier have been trained using the entire feature set that consisted of 658 features. Feature Selection technique has then been applied to reduce the feature space and to narrow it down to the most important features. The 'SelectKBest' module available in scikit-learn has been used to perform feature selection. To identify the best value for the number of features, the performance of the Gaussian Naive Bayes classifier and Decision Tree classifier has been recorded for different values of k. These results for the Gaussian Naive Bayes classifier can be seen in the tables below-

k-value	Accuracy	Precision	Recall	Area under ROC
10	51.60	0.45	0.80	0.60
20	55.84	0.48	0.69	0.59
30	56.54	0.48	0.68	0.60
40	56.86	0.48	0.67	0.60
50	58.94	0.50	0.57	0.60

60	59.36	0.51	0.53	0.60
70	58.60	0.50	0.39	0.60
100	59.01	0.51	0.33	0.59
150	59.16	0.52	0.25	0.59
200	59.34	0.52	0.23	0.59
250	59.21	0.52	0.22	0.59

Similarly, the results for Decision Tree classier have been listed below

k-value	Accuracy	Precision	Recall	Area under ROC
10	50.1	0.44	0.83	0.59
20	52.53	0.45	0.72	0.57
30	53.24	0.46	0.71	0.58
40	53.15	0.46	0.71	0.58
50	53.08	0.46	0.71	0.58
60	52.94	0.45	0.70	0.58
70	52.95	0.45	0.71	0.58

The above results suggest that for a k value of 50 both the classifiers have slightly better performance. Using these results and also to keep the models simple the number of features have been reduced to 50.

The Decision Tree classifier has then been tuned by changing its parameter values. The value for the parameter ‘min_samples_spit’ has been set to 20 because it achieved better performance. A Random Forest Classifier has then been trained on the dataset with the reduced features. The classifier has been trained for different values of ’n_estimators’ which represent the number of trees in the forest. The performance of the classifier for different values of ’n_estimators’ can be seen below

n_estimators	Accuracy	Precision	Recall	Area under ROC
30	54.34	0.47	0.85	0.66
60	54.63	0.47	0.85	0.66
90	54.78	0.47	0.85	0.66

120	54.95	0.47	0.85	0.66
150	54.86	0.47	0.85	0.66
200	54.84	0.47	0.85	0.67
250	54.84	0.47	0.85	0.67
300	54.83	0.47	0.85	0.67
350	54.82	0.47	0.86	0.67
400	54.77	0.47	0.86	0.67

Considering the above observations, the model with 200 estimators and a min_samples_split value of 40 has been chosen as the final model. A Logistic Regression classifier has also been trained on the data, this classifier had an accuracy of 46.9 % and an area under ROC of 0.64.

Results

Model Evaluation:

After data preprocessing, feature selection and parameter tuning the performance of the classifiers is recorded below

Classifier	Accuracy	Precision	Recall	Area under ROC
Gaussian Naive Bayes	58.94	0.50	0.57	0.60
Decision Tree	53.11	0.46	0.71	0.58
Random Forest	54.84	0.47	0.85	0.67
Logistic Regression	46.91	0.43	0.95	0.64

The data for the months of January and February have been selected for modeling. All the classifiers have been trained on about 80% of the data which represent the flights' information until February 15, 2015. The trained classifiers have then been tested on the remaining data which represents the information about flights after February 15th, 2015 and constitute about 20% of the data chosen for modeling. The performance measures listed in the table above have been recorded by testing the classifiers on the unseen testing data.

Predicting flight delays is a binary classification problem and the most suitable metric for evaluating binary classifiers is the area under ROC. The final Decision Tree model trained by

using ‘gini index’ as the splitting criteria and a minimum samples split of 20 had an accuracy of 53.11 and an area under ROC of 0.58. The splitter method used for this model is ‘best’ i.e. the attribute resulting in best split is always chosen rather than a random approach.

After several iterations of training the Random Forest classifier with different parameter settings, the final model used 200 estimators and a minimum samples split of 40. The criteria used for determining the best split is ‘entropy’ which calculates the information gain. The Random Forest classifier did not employ any pruning techniques for the trees and every tree is expanded until its maximum depth. This condition is facilitated by setting the value of the parameter ‘max_depth’ to ‘None’. The number of features considered while determining the best split is the square root of the number of features supplied. This condition is facilitated by setting the value of the parameter ‘max_features’ to ‘auto. This model had an accuracy of 55% and an area under the ROC of 0.67 which is better than that of the other classifiers.

As the data set is time series, and suffers from look-ahead bias, which means when the data is trained on future data and tested on past data, it would lead to inaccurate results. Hence, for data sets like this cross validation techniques cannot be applied. So, to check for the robustness of our final Random Forest classifier, it has been applied on instances from March 1, 2015, to March 10, 2015 which is considered as our validation set. This dataset is completely new and independent of the training and testing sets. The features of this validation set have also been transformed by using the previously fitted SelectKBest transformer. The final Random Forest model preformed very well in predicting the flight status on the validation set. The final model had an accuracy of 64.16%, a precision score of 0.7, a recall score of 0.75 and an area under the ROC of 0.65. These metrics indicate that the performance of the final model on this independent dataset is efficient and also beats the benchmark model and other classifiers. Considering this the Random Forest classifier has been chosen as the final model as it is robust on unseen data.

Conclusion

Free-form visualization:

One of the most important aspects of building an efficient model is to identify the best features that represent the information about data. For this project, Feature Selection has been used to reduce the size of the feature space and to identify the best features. The following figure displays the best 50 features.

It can be noticed that the most important feature is ‘DAY_OF_WEEK_Sat’ which indicates of the day is a Saturday. This observation is interesting because the analysis during the exploratory visualization phase did not indicate any strong relationship between the Day of the week and the

```
'DAY_OF_WEEK_Sat',          'AIRLINE_NK',                'AIRLINE_MQ',
'DESTINATION_AIRPORT_DFW', 'MONTH',                     'AIRLINE_HA',
'ORIGIN_AIRPORT_MSY',       'DESTINATION_AIRPORT_SLC', 'AIRLINE_F9',
'DESTINATION_AIRPORT_OGG', 'ORIGIN_AIRPORT_EWR',       'AIRLINE_EV',
'ORIGIN_AIRPORT_ANC',       'ORIGIN_AIRPORT_ITH',      'AIRLINE_DL',
'ORIGIN_AIRPORT_BOS',       'ORIGIN_AIRPORT_LGA',      'AIRLINE_B6',
'ORIGIN_AIRPORT_HNL',       'ORIGIN_AIRPORT_MDW',      'AIRLINE_AS',
'ORIGIN_AIRPORT_BWI',       'ORIGIN_AIRPORT_ORD',      'DESTINATION_AIRPORT_CLT',
'ORIGIN_AIRPORT_CLT',       'DESTINATION_AIRPORT_SFO', 'DAY_OF_WEEK_Wed',
'ORIGIN_AIRPORT_PDX',       'ORIGIN_AIRPORT_SLC',      'DAY_OF_WEEK_Tue',
'ORIGIN_AIRPORT_DCA',       'ORIGIN_AIRPORT_BGR',      'DAY_OF_WEEK_Sun',
'DESTINATION_AIRPORT_ORD', 'ORIGIN_AIRPORT_ATL',      'DESTINATION_AIRPORT_ITH',
'ORIGIN_AIRPORT_JFK',       'DAY',                     'DAY_OF_WEEK_Mon',
'DESTINATION_AIRPORT_BOS', 'AIRLINE_WN',               'SCHEDULED_ARRIVAL',
'DESTINATION_AIRPORT_LGA', 'AIRLINE_US',               'DISTANCE',
                            'AIRLINE_UA',              'SCHEDULED_TIME',
                            'DESTINATION_AIRPORT_ATL', 'SCHEDULED_DEPARTURE',
                                                       'ORIGIN_AIRPORT_DEN']
```

flight status. From the other best

features most of the features represent information about the origin airport. In the real world this aspect of flights is very important because some of the major airports in the United States have a lot of air traffic and this could be a reason for delayed flights.

Reflection:

The dataset used for this problem contained more than 1 million instances representing information about flights during the year 2015. The dataset included enough relevant information in multiple columns to implement supervised learning. Some of these columns represent the Airline, Origin Airport, Scheduled Departure, Day of the week etc. Initially, during the data cleaning phase, the columns which represented irrelevant information have been removed and the instances with null values have also been deleted. During the data exploration phase, the distribution of the status of the flights has been explored. This distribution was imbalanced as expected because the number of flights that get delayed is often less. During the modeling phase, the imbalanced class distribution has been handled by downsampling the majority class. Further analysis was done on the airlines to determine the percentage of the number of flights by each airline. Southwest Airlines and Delta Airlines have the most number of flights. To determine how the day of the week affects the flights, an exploratory visualization has been generated. Surprisingly, this visualization presented the information that the day of the week does not have much influence on the status of the flight and the number of flights delayed on the weekends is very slightly less than the other days. But this is because of the fact that the total number of flights on weekends is less than the number of flights on weekdays.

After the analysis phase, to prepare the data for modeling some of the features required preprocessing. The features that represent the time such as SCHEDULED_DEPARTURE and DEPARTURE_TIME have been preprocessed to represent the time as hour blocks. A challenge that was encountered during the project was that the dataset contained many categorical features. To enable modeling these features have been transformed to boolean binary values. As the dataset is very huge, only a subset of the data that contains information about January and February months has been used for modeling. This subset has been split into training and testing sets to validate the models. Feature Selection has been used to reduce the size of the feature space to build simpler and efficient models.

Multiple classifiers such as Gaussian Naive Bayes, Decision Tree, Random Forest and Logistic Regression have been trained on the data. These classifiers have been tuned to improve their performance on the testing set. Considering the area under the ROC curve as the primary evaluation metric, the Random Forest Classifier has been selected as the final model. This model also performs better than the benchmark model considered.

Improvement:

For this project, the models have been built only on a subset of the data that represent the first two months of 2015. This approach has been selected to build simpler models in a timely manner. One improvement that could be done is to build a more general model that is trained on the entire dataset. Another major improvement that could be made is to use other important and relevant features which represent the weather or any special occasions. These features could be used to identify some outliers or anomalies that improve the models.

References:

1. http://rstudio-pubs-static.s3.amazonaws.com/179496_64f977e866ee430484ec7c262f7583e4.html
2. https://www.kaggle.com/usdot/flight-delays/data
3. https://www.kaggle.com/fabiendaniel/predicting-flight-delays-tutorial/notebook
4. https://www.datasciencecentral.com/profiles/blogs/predicting-flights-delayusing-supervised-learning