Predicting Flight Delays From Weather Patterns

Marianela Pimienta[1] and Ricardo Romero[2]

[1]Computer Science, Florida State University, Tallahassee, FL 32301, United States
[2]Computer Science, Florida State University, Tallahassee, FL 32301, United States

Abstract

In recent history there has been a trend of increased flight delays due to inclement weather. Using airline on-time data from the Bureau of Transportation Statistics and weather data from National Oceanic and Atmospheric Association we attempted to predict flight delays. To start we pre-processed and cleaned our data in order to be used by our machine learning models. Then to deal with class imbalance we performed SMOTE on the training data. The algorithms used were Support Vector Machine, Deep Neural Net, and Gradient Boosting.

The three algorithms used performed similarly when taking accuracy and precision into consideration. While SMOTE lowered accuracy and precision it increased the F1 score which is considered a better measure for imbalanced classes.

1 Introduction

The U.S Department of Transportation's Air Travel Consumer Report stated that in 2016 on-time arrival was down from 81% to 76%. Of the reported 24% of flights delayed, the Bureau of Transportation Statistics reported that 34.33% of delayed flights were due to weather, a difference of approximately 5% when compared to the Air Travel Consumer report of the previous year. This increase in weather-related delays proves how important a reliable a weather-related delay predictor could be to both passenger's and the airline industry. [4]

We gathered our data from two sources, the National Oceanic and Atmospheric Association and the Bureau of Transportation Statistics US Domestic Airline On-Time Performance. In a later section we discuss the pre-processing required to make the data usable and easily accessible. Our learning algorithms are Support Vector Machines, Deep Neural Net[2], and Gradient Boosting, in a later section we explain the process of using these algorithms, their results, and any problems that arose because of their use.

2 Literature Survey

Previous research on this topic has shown that weather data can be used to predict flight delays. *A Machine Learning Approach for Prediction of On-time Performance of Flights* did extensive research on this topic and managed to create a tool which would not only predict a delay but also the length of the delay. The researchers used multiple models: Random Forest, Gradient Boosting, AdaBoost, and Extra-Trees. Using these algorithms the researchers obtained. [3]

Along with this paper, *Prediction of Weather-induced Airline Delays Based on Machine Learning Algorithms* also used a machine learning approach to predict whether a flight would be delayed based on current weather conditions. The machine learning algorithms used were AdaBoost, K-Nearest Neighbors, and Decision Trees. [1]

Unlike the work shown above, research conducted by Mueller and Chatterji [5] uses statistical methods instead of machine learning algorithms. Unfortunately this approach leads to generalizations and approximations which could lead to incorrect results. [3]

3 Methodology

3.1 Neural Net

A Neural Net is an algorithm used for machine learning which is somewhat based on the human brain. The model is initiated by creating the user-entered amount of layers, then placing nodes within those layers, and then randomly assigning weights to the connections between nodes both within and outside the layers. With having more layers comes the benefit of being able to capture more complex features as the input of one layer is trained on the output of the previous one. As data is then passed from one layer to the next the weights between the nodes and between layers is updated to better fit the data and make the correct prediction or classification in the output layer.

3.2 Support Vector Machine

Support Vector Machines work off the idea of separating the data using a hyperplane. In the case of 2-dimensional data this split would be considered a line. In cases where the hyperplane or line is not possible then transformations, known as Kernels, can be applied to the data to make a hyperplane separation possible. When data is more complex the initialization of the model can also include a regularization parameter which allows for the hyperplane to better fit the data in order to have a better classifier, the higher the regularization the more the slope of the hyperplane is allowed to change. Gamma, another parameter used in initialization, allows for points not closest to the hyperplane to be considered when determining where the hyperplane should be. Data visualization in these cases could assist the user in determining what to set the parameters to but the case in which most flexibility is given could lead to a better classifier however it would take longer to train.

3.3 Gradient Boosting

Gradient boosting is an ensemble model that uses weak prediction models(ex: decision tress) and creates a better prediction on each iteration. On the first iteration it uses the simple model to make a prediction. It then takes the mean standard error(MSE) of that prediction and fits the model again to those data points. The fit of the new model is then added to the original data points. In essence every iteration is a combination of the model prediction and the MSE prediction. This causes the model to become more robust and better fit the data with each subsequent iteration.

4 Data

4.1 Data Acquisition

We retrieved the weather data from the "National Centers for Environmental Information". We chose data from land based stations that were directly correlated with the airports that we wanted to analyze. The weather data included information for every hour, which we decided was best to consider. The airline information was retrieved from the "Bureau of Transportation Statistics". The final airports chosen were Hartsfield-Jackson Atlanta International Airport (ATL), Dallas/Fort Worth International Airport (DFW), and John F. Kennedy International Airport (JFK). These airports were chosen because they were all in the top ten busiest airports in the U.S. Data was acquired from the January 2009 - January 2018.

4.2 Data Pre processing

Both of the data sets acquired were acquired from government data in the form of a csv file. The pre-processing stage included several steps: (1) Join data in two tables in Sqlite. We chose to first integrate the data into a database because we had to handle multiple csv files. (2) Along with integrating everything into the database, we had to assure that each value in the rows were present and aligned with the rest of the

columns. Many of the rows were empty for a certain feature.

We chose to populate those empty entries with the entries that were closest in time to it. Many entries were also incompatible with the rest of the entries for that feature. For example some entries contained strings where the other entries contained numbers. These entries were removed and treated as blanks. For one of the features (Sky conditions) it contained a string of words denoting the sky conditions. We decided to give them a rank ("FEW": 1, "SCT": 2, "BKN": 3, "OVC": 4, "VV": 5) based on the severity of the sky condition because our assumption was that the more obscured the sky was, the worst the weather conditions were. (3) We the decided that we would find the weather information for the each flight determined on how close it was to the flight departure time and the arrival time. This seemed like an appropriate estimation because the flight information was given hourly. (4) SMOTE, Synthetic Minority Over-Sampling Technique, is a method of over sampling of the minority class in order to balance training data. It works by creating synthetic samples along the feature vector of a minority classes' nearest neighbors.

5 Implementation Details

5.1 Overview

For the mode implementations we used python3 alongside sklearn.

5.2 Splitting the Data

From looking at the dataset we could see that the three airports were evenly sized. Thus, we chose to split the data for train and test sets randomly. The training set was 70% of the overall data and the test set was 30% of the overall data.

5.3 Normalizing

Following the data pre-processing stage, we were able to create an N x M matrix where N = number of feature and M = total data sample. From looking at the matrix we could see that due to the high variance between feature, we needed to normalize the data so that the data could find a better fitting model. We normalized the data using sklearn "RobustScaler".

5.4 Feature Selection

Flight Features	Weather Features
Month	Hourly Wind Speed
Day of Month	Hourly Wind Direction
Origin Airport ID	Hourly Temperature
Reporting Airline ID	Hourly Pressure
Departure Time	Hourly Humidity
Arrival Time	Hourly Sky Conditions

Due to the fact that we had so many features to work with we had to decide which ones were the most fitting. The first trial was solely based on what previous literature surveys had said were the most significant features (Month, Day of Month, Origin, Airport ID, Reporting Airline ID, Departure Time, Arrival Time, Wind Speed, Wind Direction, Temperature, Pressure and Humidity.[3] Since there were weather features for both the origin airport and the destination airport we tested the models with two different data sets. One of the data sets included only departure delay and only contained origin airport weather data.

The other data set took into account if there was any type of delay(both at departure or arrival) and included the weather features for both origin and destination airports. It should be noted that we thought

about separating them as two data sample in the array(one for departure delay and one for arrival delay) but the paper in which we modeled our methods after [3] noted that this did not work well for them. We made the assumption that it wouldn't work well because departure and arrival delays might be cause by differing features or differing weather times. This might have caused the model not to fit well to the data. The second trial in feature selection included the features mentioned above along with sky conditions. We believe that sky conditions(such as sunny, overcast, raining) would be relatively important in determining if a flight was delayed.

We then decided to use recursive feature selection on the data set. Feature selection would recursively try to fit the model with different feature and determine which feature was deemed the most significant every time. The most significant features were used to train the models again to see if this would give us more precise results.

5.5 Models

The Neural network was initialized using sklearn MLPClassifier. It was given two hidden layers because we believed that would be the best for a binary classification. lbfgs was used as the solver. lbfgs is an optimizer in the family of quasi-Newton methods.

The SVM was created using sklearn LinearSVC. A linear classifier was used in this scenario because of the amount of data made it impossible for a non-linear svm to converge. The SVM also tended not to converge if the data was not firt balanced with SMOTE and then normalized. Even with a high number of iterations, on some runs the SVM failed to converge.

The Gradient Boosting classifier was initialized using sklearn GradientBoostingClassifier. It was given 500 iterations to go through at a learning rate of 0.01. We do not believe that we are over-fitting the data because of the large amount of samples and because the learning rate was set to a low rate.

5.6 Scores

We wanted to have as much insight into the data as possible. We decided to record the accuracy, recall, precision, and f1 score of each model. Since the data classes were unbalanced, we believed that the f1 score might be the best indicator. Recall was calculated as r = TP/(TP+FN). Precision was calculated as p = TP/(TP+FP). The f1 score was then calculated as F1 = (2 * r * p) / (r + p).

6 Results

Neural Net

Neural Net				
	Accuracy	Precision	Recall	F1 Score
Regular	0.8094	0.3819	0.0217	0.0403
Feature Selection	0.8089	0.5092	0.0213	0.0398
Smote	0.497	0.2463	0.7886	0.3749
Feature Selection and Smote	0.5295	0.2525	0.7424	0.3761

Figure 1. Neural Net - Origin Airport Only

Neural Net				
	Accuracy	Precision	Recall	F1 Score
Regular	0.8084	0.5842	0.0344	0.0647
Feature Selection	0.8078	0.567	0.0319	0.0595
Smote	0.5348	0.32576	0.7421	0.3818
Feature Selection and Smote	0.5494	0.2604	0.7174	0.3814

Figure 4. Neural Net - Origin and Destination

Support Vector Machine

Support Vector Machine				
	Accuracy	Precision	Recall	F1 Score
Regular	0.8091	0.5714	0.0028	0.0056
Feature Selection	0.8089	0	0	0
Smote	0.5625	0.262	0.7097	0.3827
Feature Selection and Smote	0.5711	0.2625	0.6881	0.3801

Figure 2. Support Vector Machine - Origin Airport Only

Support Vector Machine				
	Accuracy	Precision	Recall	F1 Score
Regular	0.8066	0.5429	0.0033	0.0066
Feature Selection	0.8065	0.6667	0.0003	0.0007
Smote	0.5609	0.2647	0.7134	0.3861
Feature Selection and Smote	0.5811	0.2692	0.6791	0.3856

Figure 5. Support Vector Machine - Origin and Destination

Gradient Boosting

Gradient Boosting				
	Accuracy	Precision	Recall	F1 Score
Regular	0.8135	0.6106	0.0653	0.118
Feature Selection	0.8108	0.5516	0.0519	0.0949
Smote	0.1972	0.1917	0.9952	0.3214
Feature Selection and Smote	0.2368	0.1958	0.964	0.3255

Figure 3. Gradient Boosting - Origin Airport Only

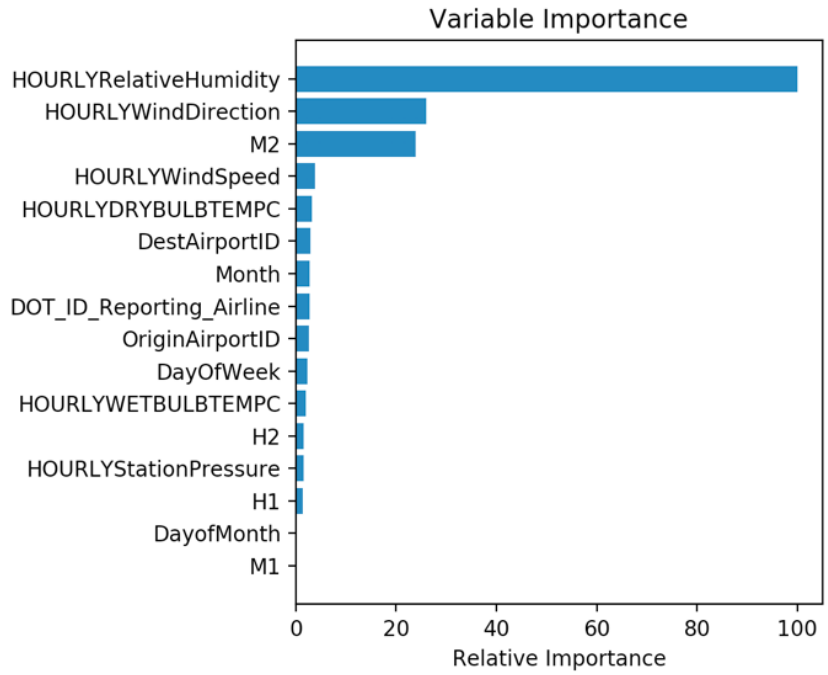
Gradient Boosting				
	Accuracy	Precision	Recall	F1 Score
Regular	0.8113	0.621	0.0648	0.1174
Feature Selection	0.8108	0.6016	0.066	0.119
Smote	0.1968	0.1939	0.9976	0.3247
Feature Selection and Smote	0.2117	0.1951	0.9828	0.3255

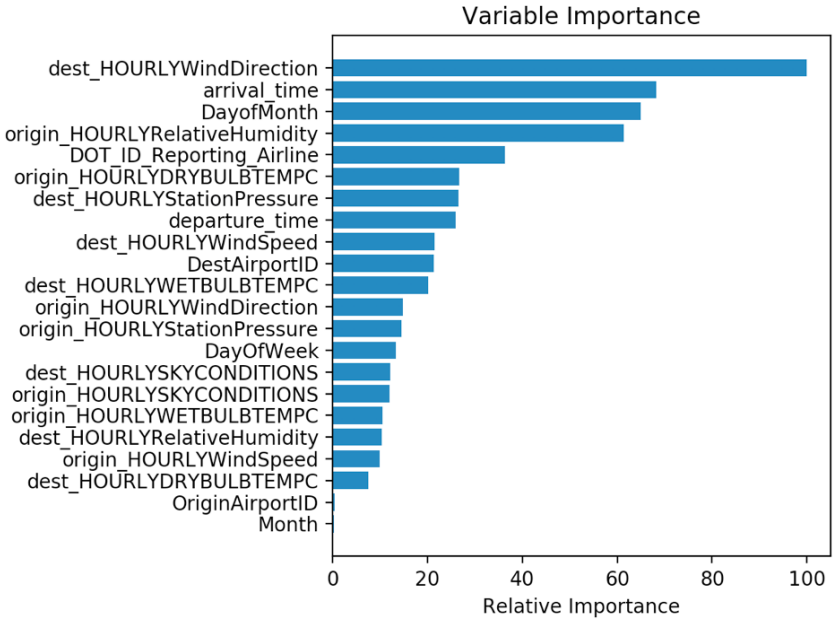
Figure 6. Gradient Boosting - Origin and Destination

6.1 Origin vs Origin and destination

Originally, we had wondered if we should include the weather information for the destination along with the weather information for the origin. Since the paper that we had referenced, included both origin and destination we experimented with both. From looking at figure 1, figure 2, and figure 3(models when just considering the origin) we can see that there does not seem to be a significant difference from the models that we considered both destination and origin weather(figure 1, figure 2, and figure 3). This leads us to believe that if we are trying to solve the problem of "Will there be a delay from the origin city?" we should just use origin data to reduce the matrix complexity.

6.2 Recursive Feature Selection





This recursive feature selection was one thing that we hoped would increase the accuracy of the models but overall it did not. Figure 4 shows the variable importance for when we considered only the origin weather in the features. The most important features seemed to be humidity and wind direction. Figure 5 shows the variable importance for when we considered the weather of both the origin city and the destination city. Wind direction and arrival time seemed to most significant features in that case.

6.3 Smote vs Non-smote

When looking at figure 1, figure 2, and figure 3, we see that the accuracy was much higher for the occurrences where we did not use the SMOTE technique on the training data. We also see that for the occurrences that we did use SMOTE on the overall f1 score was higher. The recall was also highest when using the SMOTE technique.

6.4 Models

All of the models also gave around the same scores all around for all of the different implementations. All of the models gave a consistent accuracy score of around 0.80 and an f1 score of about 0.05 for models that were not used with SMOTE. The models also gave a consistent accuracy score of around 0.55 and an f1 score of about 0.35 for models that were trained by data where the SMOTE technique was used. Precision and recall both increased throughout the models where SMOTE was performed.

7 Conclusion

From the results we can see that the recursive feature selection did not have a significant impact on the models. I believe this was due to fact that we had already defined the features we thought were best based on the papers that we had previously read. The features that were deemed as important, such as humidity, seemed like appropriate features to have a high importance because the more the weather becomes inhibiting(rain, snow, etc.) the more humid that it gets.

Overall when looking at the models where we did not perform SMOTE on the training data, the accuracy seems relatively high(0.80). This is misleading, since we have an imbalance in the classes where the number of delayed flights is much fewer than the number of flights that were on time, this means that and f1 score is a better measure. The f1 score when we did not perform SMOTE was relatively low for the different models(around 0.05). When we did perform SMOTE on the models the average f1 score was 0.35. Although not a great score, it still shows that oversampling the delayed flights in the training data did help in increasing a measure of precision and recall.
Overall it does not seem that we can accurately predict if a flight will be delayed. We believe that a big factor in this was due to the weather data that we chose. The weather information seemed not to be very consistent.

8 Future Work

For future work we would have liked to explore a different weather database and include all of the features provided. We would then like to run the recursive feature selection algorithm in order to best pick the most significant features. We would have also liked to train the model on more than three airports. Perhaps limiting the amount of year and increasing the amount of airport might have produced better results. Finally, We would have liked to implement the model on fairly recent data.

References

[1] Prediction of weather-induced airline delays based on machine learning algorithms. *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), Digital Avionics Systems Conference (DASC), 2016 IEEE/AIAA 35th*, page 1, 2016.

[2] Building usage profiles using deep neural nets. *2017 IEEE/ACM 39th International Conference on Software Engineering: New Ideas and Emerging Technologies Results Track (ICSE-NIER), Software Engineering: New Ideas and Emerging Technologies Results Track (ICSE-NIER), 2017 IEEE/ACM 39th International Conference on, ICSE-NIER*, page 43, 2017.

[3] A machine learning approach for prediction of on-time performance of flights. *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC), Digital Avionics Systems Conference (DASC), 2017 IEEE/AIAA 36th*, page 1, 2017.

[4] Air travel consumer report: April 2018 numbers. *States News Service*, 2018.

[5] G. Chatterji E. Mueller. *Analysis of Aircraft Arrival and Departure Delay Characteristics*. Technical Forum, 2002.