Weight Prediction: Flight Delays

Yoon, Duk Jin
Information Systems
Hanyang University

Hyun Joong Kim
Information Systems
Hanyang University

Byeong Gon Lee
Hanyang University
Information Systems

Gyu Hyuk Kwon
Hanyang University
Information Systems

***Abstract*—Flight delays are an inconvenience to the passengers and to the airlines. A delayed flight can cause passengers to miss all of their flights if they were on a multi-plane trip or even in their scheduled tour trip. Ultimately, this will lead to frustration and complaints that causes negative action towards their corresponding airline. Furthermore, the results of delayed flights also cause a great impact to the airline companies. According to the Federal Aviation Administration, the estimated cost of flight delays is around 22 billion dollars. Airlines are not only forced to pay the federal authorities, but also are forced to compensate the passengers. There are many causes of a flight delay, statistics shows that most of the delayed flights are due to weather issues. However, there are also other causes such as: security delays, airspace system delay, and maintenance delays etc. Our proposal is to predict these kinds of delays and find out which factors impact the total delay the most.**

Keywords- flight delays; passengers; algorithm; machine learning;

I. Introduction

In this experiment, we are going to use the published data from the Bureau of Transportation Statistics to analyze and predict flight departure delays for a subset of commercial flights in the United States. The dataset that we are going to use has a size of 10 million values from 2008. However, he have decided to use 3 million of the entries to train our models. We will be using the historical on-time performance and the possibilities causes of a delay. Furthermore, we are planning to use 15 categories from the dataset: the distance, security delay, national airspace system delay, carrier delay, weather delay and late aircraft delay etc. Our initial idea is to use the several machine learning libraries and make representation of models to find the weights of each delay or factor. Furthermore, Tensorflow will be a tool that will help us get the results we want. This software is mostly used for dataflow programming used across a range of tasks. It is a symbolic math library used for machine learning applications such as our proposed project. Our main proposal is to find models that will accurately detect and predict the overall delay of a flight plan based on the factors mentioned above. Furthermore, we will also use different types of algorithms(in Python and R) on the same model to see if the outcome is accurate across various models. Lastly, if the model is trained in a significant way, we will see if the results are compatible with plane schedules and check if the delay is calculated with similar weights represented in out model. We have tried to crawl data from various airlines and different countries to see if the model is also compatible to different environments but such data are classified to be confidential.

II. Related Works

A. A Review on Flight Delay Prediction

[1]This paper proposed a machine learning model that explores the development of algorithms that can learn from data and provide predictions based on it. It deals with works that study flight systems are increasing the usage of machine learning methods. The methods commonly used include k-Nearest Neighbor, neural networks, SVM, fuzzy logic, and random forests. They were mainly used for classification and prediction.

B. Binary Classification: Flight Delay Prediction

[2]To predict flight delays this paper proposes a binary classification task with two classes, whether the flight will be delayed, or whether it will be on time. They built this experiment using Azure ML Studio. In the experiment, the model is trained using a large number of examples from historic flight data, along with an outcome measure that indicates the appropriate category or class for each example. The two classes are labeled 1 if a flight was delayed, and labeled 0 if the flight was on time.

C. Predicting Flights Delay Using Supervised Machine Learning

[3]This paper proposes a supervised machine learning technique called logistic regression to predict delayed flights. Logistic regression provides a probability of belonging to one or the two cases (delayed or on-time). Since probability ranges from 0 to 1, they used the 0.5 cutoff to determine which bucket to put in their probability estimates in. If the probability estimate from the logistic regression is equal to or greater than 0.5 then they assign it to be on-time else its delayed.

D. Optimizing Arrival Flight Delay Scheduling Based on Simulated Annealing Algorithm

[4]This paper presents a model based on the simulated annealing algorithm for optimizing arrival flight delays to reduce serious air traffic flight delays. The authors of this paper used simulated annealing algorithm to presents a queue and attempter model for flight delay to minimal delay cost. Compared with the traditional flight delay sequence method, this model is effective and easy to implement. It also can reduce the cost and the influence of the delay as much as possible.

E. *Predicting Flight Delays*

[5]Dieterich Lawson and William Castillo used SVMs, Nave Bayes, and Random Forests for attempting to predict whether or not flights would be delayed. In general, they tried to choose algorithms that parallelize well, so that they could run them on top of Apache Hadoop and take full advantage of the large size of their dataset. In result, they believe that improvements in these areas would only come with different data, i.e.) more features, further data processing, or better domain knowledge, because most of their approaches yielded similar precisions and recalls.

III. Datasets

In order to accurately predict the flight delays we are going to use the datasets that is provided by the Bureau of Transportation Statistics for the training set. Some of the features will be excluded from our model to increase validity(as some factors are not relevant to calculating the total delay) This dataset will provide us with the basic information that is required in testing our model. The variables and the description is as the following.

TABLE I

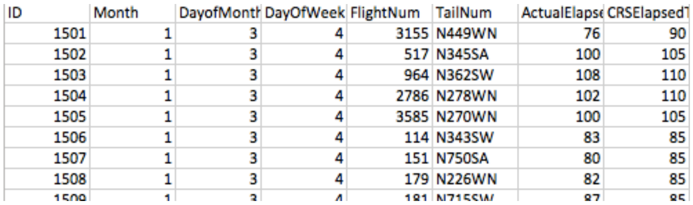
Variables	Description
ID	Primary Key
Year	1987-2008
Name	Name
Month	1-12
DayofMonth	1-31
DayOfWeek	1(Monday) 7(Sunday)
DepTime	Actual departure time (local)
CRSDepTime	Scheduled departure time (local)
ArrTime	Actual arrival time (local)
DepDelay	Departure delay, in minutes
Origin	Origin IATA airport code
Dest	Destination IATA airport code
Distance	In miles
TaxiIn	Taxi in time, in minutes
TaxiOut	Taxi out time in minutes
Cancelled	Was the flight cancelled? (T/F)
CancellationCode	Reason for cancellation (A = carrier, B = weather, C = NAS, D = security)
Diverted	1 = yes, 0 = no
CarrierDelay	In minutes
WeatherDelay	In minutes
NASDelay	In minutes
SecurityDelay	In minutes
LateAircraftDelay	In minutes

Each of these variables are important attributes in our machine learning model. Hence, it is critical that we use as many variables as possible in order to accurately predict the result. It is to say that the destination, origin and tail number were not added to the parameters in our model. In order to see if the assumption we made is accurate, we will do a second training consisting of the top 6 features that had the most weight on our decision tree classification. TaxiIn and TaxiOut variables show the time that the aircraft was on wheels before and after takeoff. The TotalDelay, which is the main factor that our group has analysed, has been calculated based on the sum of the ArrDelay and DepDelay. Factors such as Diverted, CancellationCode, Cancelled, and Name were excluded due to irrelevance in data to our result. Variables that indicate time and date were included becuse we have made an assumption that time of the week and date can also affect the delay. For example, the delay for weekends and weekdays might differ due to a difference in the number of people who use the airport. NASDelay refers to the national air system delay.

TABLE II:
Top 6 variables that have the highest weight

Variables	Weight
FlightNum	0.13866885595854561
DayofMonth	0.10923164044593994
Distance	0.10494025612935766
TaxiOut	0.095221819727173793
AirTime	0.089389198728014416
TaxiIn	0.084910841450804964

The above table shows the 6 highest variables that had the most significant weight in out decision tree algorithm. In order to see accuracy and change in the prediction of delays in the model, we have trained the data twice: once with most of the variables included; and once with the above 6 variables as parameters. We have tried this is both Python and R.



ID	Month	DayofMontl	DayOfWeek	FlightNum	TailNum	ActualElaps	CRSElapsedT
1501	1	3	4	3155	N449WN	76	90
1502	1	3	4	517	N345SA	100	105
1503	1	3	4	964	N362SW	108	110
1504	1	3	4	2786	N278WN	102	110
1505	1	3	4	3585	N270WN	100	105
1506	1	3	4	114	N343SW	83	85
1507	1	3	4	151	N750SA	80	85
1508	1	3	4	179	N226WN	82	85
1509	1	3	4	181	N715SW	87	85

Fig. 1: Dataset

The above chart shows how our dataset looks like after we have done pre-processing. For values that had data missing, we have used the median value of what the other values had in the column. Furthermore, for the blank space in the 5 delays, we have interpreted that there was no delay. In other words, the blanks were converted to zeros.

IV. Classification Model Description

A. *Decision Tree Classifier*

Python: sklearn.tree.DecisionTreeClassifier
R: library(rpart)
Decision tree analysis is the most popular decision support technique in data mining field. From the root, input training set is divided recursively at split point of input features(attributes) making a tree. The benefit of using decision tree is simple understanding of classification process. It uses white box model so it is easy to interpret and can be visualized.

B. *G k-NN(k-nearest neighbors)*

Python: python: sklearn.neighbors.KNeighborsClassifier
R: library(kknn)
kNN is a prediction method for classification as well as regression type prediction problems. Its one of the simplest

machine-learning algorithm, that test case is simply assigned to the class that most k nearest training cases are found. The result get different depending on k value you set.

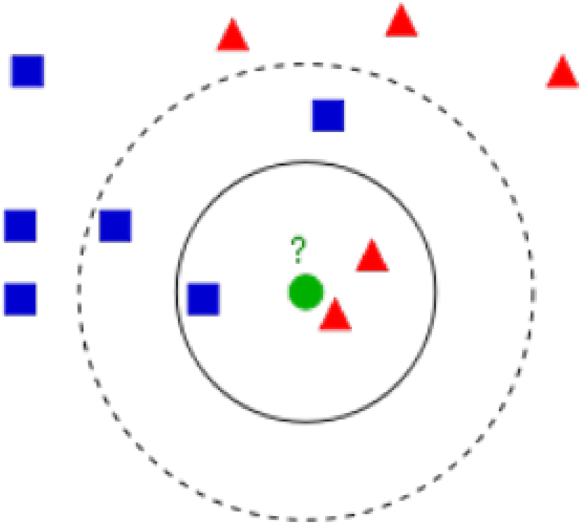


Fig. 2: K-NN

C. Random Forest Classifier

Python: sklearn.ensemble.RandomForestClassifier
R: library(randomForest)
A random forest is a meta classifier that has a number of decision tree classifiers on various sub-samples of the dataset. It uses mode of the classes for classification and mean prediction for regression to improve the predictive accuracy and control over-fitting.

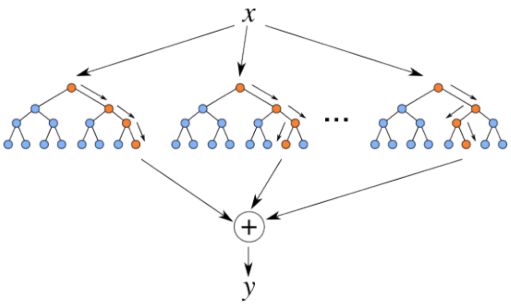


Fig. 3: Random Forest

D. Naive Bayes Net Classifer

Python: sklearn.naive bayes.GaussianNB
Naive Bayes Classification is a simple probabilistic classification based on applying Bayes theorem using the naive independence assumptions. In this algorithm, there is an assumption that every features are independent between each other. On that assumption, probability distributions of each feature multiplied. This algorithm is widely used because it requires a small amount of training data to estimate the necessary parameters and fast comparison due to its extremely simplified assumption.

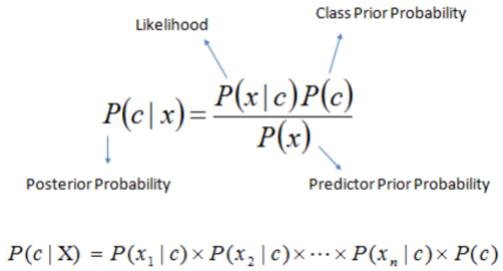

Likelihood
Class Prior Probability
$P(c \mid x) = \frac{P(x \mid c) P(c)}{P(x)}$
Posterior Probability
Predictor Prior Probability
$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$

Fig. 4: Bayes Theorem

E. Support Vector Machine

R: library(kernlab)
SVMs are supervised learning algorithms that are mostly used for classification and regression. SVMs produce linear classifiers called hyper plane that separate the data into multiple subsections. Between the input and output vectors, either a classification function or a regression function is used to transform input data to a high dimensional feature space in which the input data becomes linearly separable. SVMs are effective in high dimensional spaces due to Kernel functions and memory efficient due to support vectors which are samples of training points

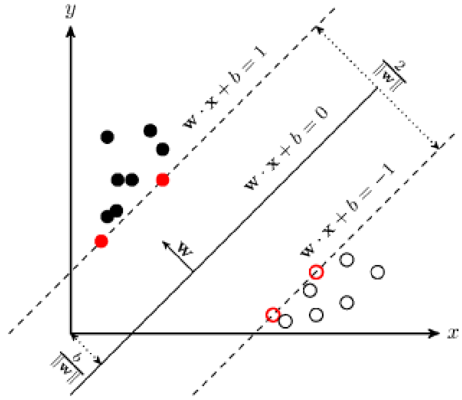

y
x
$\mathbf{w} \cdot \mathbf{x} + b = 1$
$\mathbf{w} \cdot \mathbf{x} + b = 0$
$\mathbf{w} \cdot \mathbf{x} + b = -1$
$\frac{2}{\|\mathbf{w}\|}$
$\frac{b}{\|\mathbf{w}\|}$
$\mathbf{w}$

Fig. 5: Support Vector Machine

V. Environment

A. Python3.5.2

Python is a easy-to-use and extensible object-oriented programming language. It can execute a lot of complex functions with ease, thanks to its large standard library. There are a lot of machine learning libraries for python and we mainly used scikit-learn libraries. We tried Tensorflow as well for Neural Network.

B. Jupyter Notebook

Notebook documents are documents produced by the Jupyter Notebook App, which contain both computer code in python and rich text element. Notebook documents are both human-readable documents containing the analysis description and the results as well as executable documents which can be run to perform data analysis. It was crucial to use this tool because we had to show the decision tree and classification diagram visually.

C. R

R is a popular statistical software which provides statistical computation and graphics. R has a lot of statistical modeling, classification and clustering models, and visualization techniques. It is highly extensible and easy-to-use.

VI. Evaluation and Analysis

TABLE III:
Percentage of Each Classification Mode

Classification Model	Python	R
Decision Tree Classifier	79.57	73.81
Random Forest Classifier	73.99	86.86
K-NN Classifier	55.89	34.16
SVM	NA	86.68
Naive Bayes Net	45.94	NA

We had some difficulties in using the SVM algorithm in Python and the Naive Bayes Net algorithm in R. Table 3 shows the accuracy of the Total Delay compared to the Original Delay that is presented in 300 rows of data. There was a higher precision in accuracy for the decision tree classifier in python and knn whereas random forest and support vector machine in R had the highest precision of all. The above table is the precision of when we used most of the variables for training.

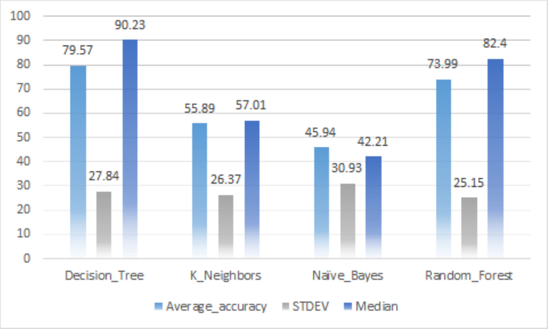


Fig. 6: Python

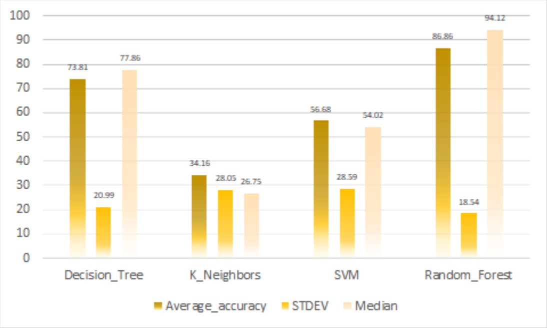


Fig. 7: R Programming

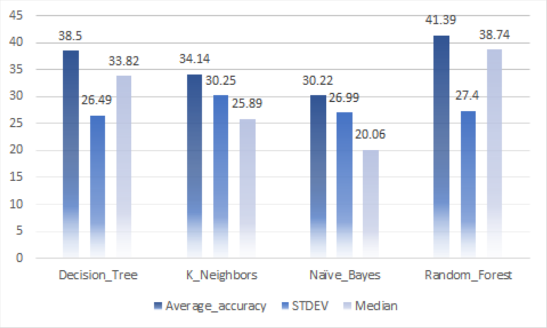


Fig. 8: Python Top 6

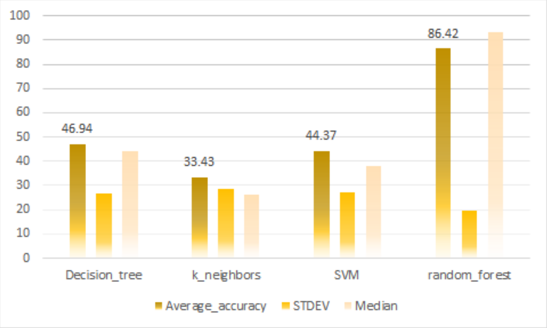


Fig. 9: R Programming Top 6

The above graphs show the change in accuracy of our models after we have take out the parameters except the top six: FlightNum, DayOfMonth, Distance, TaxiOut, Airtime and TaxiIn. This process did not boost the accuracy of our models; in contrast, there was a drastic decrease in the precision of all of the models.

TABLE IV:
Algorithm Rankings Based on Precision

Rank	Python	R Programming
1	Decision Tree Classifier	Random Forest Classifier
2	Random Forest Classifier	Decision Tree Classifier
3	K-NN Classifier	SVM
4	Naive Bayes	K-NN Classifier

VII. Conclusion

Flight delays are an important factor to consider for both the airlines and the passengers. It is crucial that we must mitigate the possible of flight delays that will occur in the near future. The impact of this cause can be devastating to such parties. Moreover, our goal in this paper will likely reduce the number of delays by analyzing the algorithms that would show the highest level of precision in predicting the total delay of planes. In addition, we have excluded the weather factor as it was the biggest reason that would impact the results of this study the most. The results of the project showed that the decision tree classifier was the most accurate when using Python and random forest classifier in case of using R. Regardless of programming language, the support vector machine algorithm shows the highest accuracy in predicting the total delay. Furthermore, reducing the number of parameters in our models did not help in increasing the prediction. Although our original goal was to see if factors other than weather would lead to a significant result in the total delay of flight, the dataset itself and the methods we have used have resulted to be ineffective. Even when we have specified the variables that had the most weight based on some of our models, the accuracy did not increase. Thus, we can conclude that our total delay is most effected by weather and that other factors have weak correlation to total delay.

References

[1] Alice Sternberg, Jorge Soares, Diego Carvalho, Eduardo Ogasawara, *A Review on Flight Delay Prediction*, Nov 6, 2017, https://arxiv.org/pdf/1703.06118.pdf

[2] AzurML Team, *Binary Classification: Flight Delay Prediction*, Sept 2, 2014, https://gallery.cortanaintelligence.com/Experiment/Binary-Classification-Flight-delay-prediction-3

[3] Peter Chen, *Predicting Flights Delay Using Supervised Machine Learning*, Mar 30, 2015, http://dataillumination.blogspot.kr/2015/03/predicting-flights-delay-using.html

[4] Tian Jungai, Xu Hongjun, *Optimizing Arrival Flight Delay Scheduling Based on Simulated Annealing Alogrithm*, Mar 30, 2015,https://www.sciencedirect.com/science/article/pii/S1875389212013867

[5] Dieterich Lawson, William Castillo, *Predicting Flight Delays*, http://cs229.stanford.edu/proj2012/CastilloLawson-PredictingFlightDelays.pdf