Application of Machine Learning Algorithms to Predict Flight Arrival Delays

Nathalie Kuhn[*] and Navaneeth Jamadagni[†]
Email: [*]nk1105@stanford.edu, [†]njamadag@stanford.edu

Abstract—Growth in aviation industry has resulted in air-traffic congestion causing flight delays. Flight delays not only have economic impact but also harmful environmental effects. Air-traffic management is becoming increasingly challenging. In this project we apply machine learning algorithms like decision tree, logistic regression and neural networks classifiers to predict if a given flight's arrival will be delayed or not. We show that with only three features we were able achieve a test accuracy of approximately 91% for all three classifiers.

Index Terms—Flight prediction, air-traffic management, decision tree

1 Introduction

OVER the last twenty years, air travel has been increasingly preferred among travelers, mainly because of its speed and in some cases comfort. This has led to phenomenal growth in the air-traffic and on the ground [1]. Increase in air traffic growth has also resulted in massive levels of aircraft delays on the ground and in the air [2]. These delays are responsible for large economic and environmental losses. According to [3], taxi-out operations are responsible for 4,000 tons of hydrocarbons, 8,000 tons of nitrogen oxides and 45,000 tons of carbon monoxide emissions in the United States in 2007. Moreover, the economic impact of flight delays for domestic flights in the US is estimated to be more than $19 Billion per year to the airlines and over $41 Billion per year to the national economy [4]. In response to growing concerns of fuel emissions and their negative impact on health, there is an active research in the aviation industry for finding techniques to predict flight delays accurately in order to optimize flight operations and minimize delays.

The input to our algorithm is rows of feature vector like departure date, departure delay, distance between the two airports, scheduled arrival time etc. We then use decision tree classifier to predict if the flight arrival will be delayed or not. In the US, FAA considered a flight to be delayed when difference between scheduled and actual arrival times is greater than 15 minutes. Furthermore, we compare decision tree classifier with logistic regression and a simple neural network for various figures of merit.

2 Related Work

There are several work in the literature that focus on air-traffic management and optimization. In [2], the authors show that the Ant algorithm can be applied to optimize aircraft taxi movements on the ground by reducing aircraft taxi-times. Jiang et.al in [5] developed a Genetic algorithm to optimize the runway and taxiway scheduling, and show a better taxi-time results compared to the ant-algorithm presented in [2]. The work presented in [5] and [2] approach the optimization problem differently. Nogueira et.al focus on choosing the shortest path for an aircraft with the existing data, applying their method to all the aircrafts on the ground while making corrections one-the-fly in case of an interaction with another aircraft. The objective of this study is to show that the Ant algorithm can optimize taxi paths, hence taxi-times. Jiang et.al in [5] focus on setting the taxi-time for an aircraft, and then choosing the right taxi-route to minimize interactions with other aircrafts. Therefore, the model in [5] aims at reducing the aircrafts taxi distance. The model in [5] also guarantees continuous taxiing and thus reducing the delay associated with taxiing Nogueira et.al's model fails to offer such guarantees. The work presented in [5] is already being applied in practice. Aircraft start their push-back process from the gate within a given time-slot that is based on an evaluation of all the traffic in the airport, to minimize taxi-times.

Another important area that is extensively studied is finding and measuring factors affecting aircraft delays on the ground and in the air and develop machine learning algorithms to optimize airline and airport operations based on the factor responsible for the flight delay. In [6], the authors present a method to measure the impact of the delays occurring at one airport on other airports. They developed a model that iterates two main components: a queuing model that computes delays at individual airports, and a delay propagation algorithm. In response to the local delays calculated by the queuing model, the delay propagation algorithm continuously updates flight schedules and demand-rates at all airports in the network. Such a technique is unique in the area and more research using such techniques could be very beneficial to the aviation industry in terms of practical applications. Another example is the study of factors responsible for aircraft taxi-delays that occur on the ground. In [7], the authors investigate the possibility of reducing taxi-times of a departing aircraft through a model developed using a queuing system for departing aircraft that can be optimized on-the-fly. In [3], the authors extend the work presented in [7] by using of more complete datasets and deploying more rigorous statistical tools.

In [8], the authors compare various machine learning algorithms to predict flight delays, but failed to consider simple neural networks and decision tree classifiers. Because of our recent exposure to the field of machine learning, we decided to apply simple machine learning algorithms like decision tree and simple neural networks to predict flight delays, and investigate if we can predict flight delay with fewer feature-set accurately.

3 Dataset and Features

To train and test our models, we used a publicly available Kaggle dataset for United States domestic air-traffic. The original source of our dataset is the on-line Bureau and Transportation Statistics database [9]. The data set is for the year 2015 and consists of well over 5 Million examples with 30 features categorized as follows :

- Information about flight (day, day of the week, airline, flight number, tail number)
- Information about origin and destination (origin airport, destination airport)
- Information about the departure (scheduled departure, departure time, departure delay, taxi-out, wheels-off)
- Information about the flight-journey (scheduled time, elapsed time, air time, distance)
- Information about the arrival (wheels-on, taxi-in, scheduled arrival, arrival time, arrival delay)
- Information about diversion, cancellation and reason of delay (air system delay, security delay, airline delay, late aircraft delay, weather delay)

The first step involved verification of the dataset completeness. While the dataset was mostly complete, there were some missing data. For features such as arrival delay and departure delay, it was easy to calculate the missing data when scheduled and actual departure and arrival times are known. For features like tail number and flight number, the missing values were impossible to calculate and therefore we removed examples for such missing values from our data set. Furthermore, for classification purposes, it was useful to have labels that state if this flights arrival or departure was delayed. Therefore, we added few labels like arrival and departure delayed to our existing dataset.

Figure 1 shows the fraction of flights delayed in the year 2015, grouped by airlines. The airlines are shown using IATA airline codes. For example, label AA is for Alaska Airlines and about 17% of its flights were delayed in 2015. Figure 2 shows the arrival-delay distribution during each day of every month in 2015. For example, label 1 denotes the delay distribution for the first day of every month in 2015.

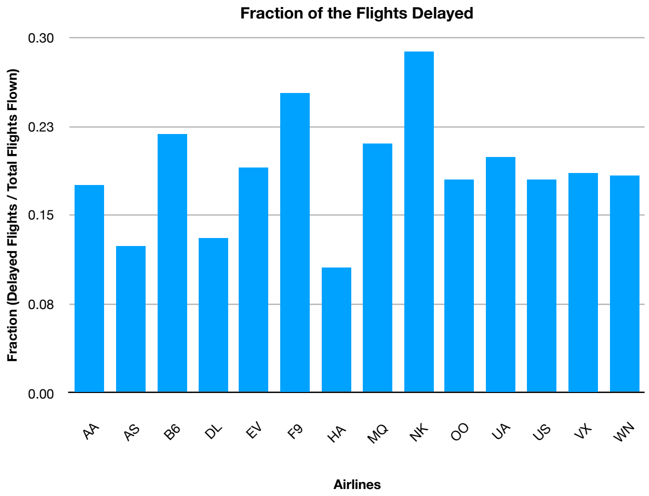

Fig. 1. Figure shows the fraction of the total flights delay at arrival, grouped by airlines.

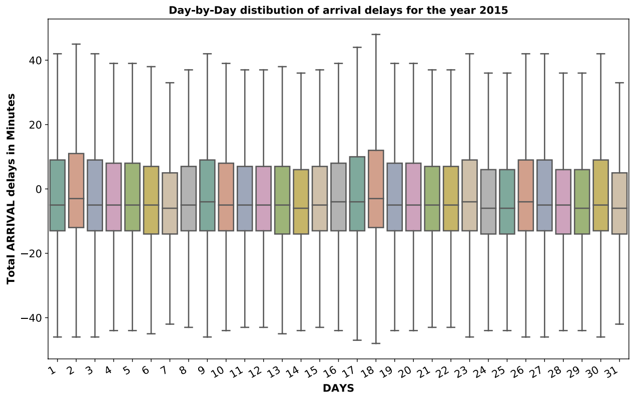

Fig. 2. Figure showing the day-by-day distribution of the arrival delays of flights for all the months in 2015

3.1 Training Data and Feature Selection

The main objective of this project is to predict if a flight will be delayed or not, hence we chose the following 13 out of 30 features which are usually known in advance: *Month, Day, Day of the week, Flight Number, Origin airport, Destination Airport, Scheduled departure, departure delay, taxi-out, distance, Scheduled Arrival*.

We decided to use our laptops for training and testing our models. Because of the computational limitations of our laptop we chose smaller subset of 100 thousand examples out of the 5 million examples. The 100 thousand samples were chosen at random such that 50 thousand of the examples had flights with arriving late and 50 thousand example with flights arriving on-time.

4 Methods

As mentioned in Section 1, we applied the following three models to predict if the flight will be delayed or not: Decision Tree, Logistic Regression and Neural Networks. In this section we describe the methodology we employed along with a brief description of the models we chose.

4.1 Methodology

We first used the training set, after 70:30 split, with 13 features to train the decision tree classifier. The decision tree classifier implementation in scikit library reports the importance score for each feature [10]. We then used the top-3 features to retrain the decision tree classifier, and train logistic regression and neural network.

4.2 Decision Tree

The main idea behind the decision tree algorithm is to build a tree-like model from root to leaf nodes. All nodes receive a list of inputs and the root node receives all the examples in the training set. Each node asks a true or false question for one of the features and in response to this question the data is partitioned in to two subsets. The subsets then become the input the child nodes where the child node asks another question for one of the other features. As the tree is built, the goal of a question at each node is to produce the purest possible labels or in other remove uncertainty associated with predicting a label label. The challenge to building such a tree is which question to ask at a node and when. To do this, decision tree algorithm uses well known indices like entropy or Gini-impurity to quantify an uncertainty or impurity associated with a certain node. Equations (1) and (2) show how entropy and Gini-impurity are calculated, respectively, for a subset of data. In the equations, C is the number of classes. More details on decision trees can be found in [11].

$$H(s) = -\sum_{c \in C} p(c) \log p(c) \tag{1}$$

$$H(s) = 1 - \sum_{c \in C} p(c)^2 \tag{2}$$

4.3 Logistic Regression

Logistic regression is a simple classification algorithm that uses the hypothesis in Equation (3)

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \tag{3}$$

where $\theta^T x = \theta_0 + \sum_{j=1}^{n} \theta_j x_j$. As described in [12], we can find parameter θ that best describes our training data using the maximum likelihood estimation and gradient ascent specified in Equations (4) and (5), respectively.

$$l(\theta) = \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)})) \tag{4}$$

$$\theta := \theta + \alpha \nabla_\theta l(\theta) \tag{5}$$

4.4 Neural Network

Neural Network is built by stacking together multiple neurons in layers to produce a final output. First layer is the input layer and the last is the output layer. All the layers in between is called hidden layers. Each neuron has an activation function. Some of the popular activation functions are *Sigmoid*, *ReLU*, *tanh* etc. The parameters of the network are the weights and biases of each layer. The goal of the neural network is to learn the network parameters such that the predicted outcome is the same as the ground truth. Back-propagation along loss-function is used to learn the network parameters.

Figure 3 shows the neural network we used in this project. As shown in the figure, our neural network consists of a single hidden layer with four neurons. We used *Sigmoid* activation function for neurons in both hidden and output layer, and a binary cross-entropy loss function described in Equation (6).

$$L = -\frac{1}{m}\sum_{n=1}^{m}(y_n \log \hat{y_n} + (1 - y_n)\log(1 - \hat{y_n})) \quad (6)$$

where $\hat{y}$ is the predicted label and y is the true label.

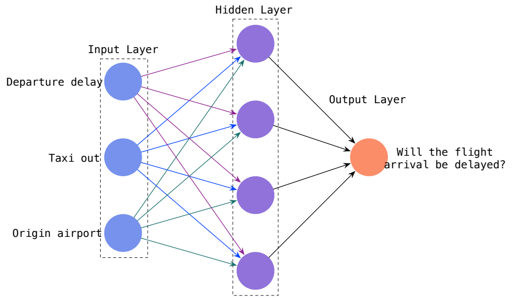

Fig. 3. Neural Network used in our project

5 Results and Discussion

As discussed in Section 3, we used 100 thousand samples to train and test the three classifiers with the recommended 70-30 split. We performed 10-fold cross-validation for decision tree and neural network classifiers and used the scikit and keras API where ever necessary. For the decision tree algorithm we parameterized the depth of the tree for better accuracy, and for logistic regression and neural network we applied L2 regularization to prevent the model from over-fitting. In addition to traditional classification figures-of-merit like AUC, precision, accuracy, recall, we also like to know tree depth and the total leaf nodes for the decision tree classifier.

Table 1 shows the top 3 features reported by the decision tree classifier along with their importance score. It is interesting to note that scheduled arrival time and destination airport does not contribute much to a flight's arrival delay.

TABLE 1
Top 3 features with importance score reported by decision tree classifier

Features	**Importance Score**
DEPARTURE_DELAY	0.8478
TAXI_OUT	0.1438
ORIGIN_AIRPORT	0.0031

We observed that the training and test accuracies for the three classifiers were approximately 91%. For the decision tree classifier, the tree depth is 7 and total number of leaf nodes is 127 which much less than 1% of the training examples. Figure 4 show the 3D scatter plot of the top-3 features reported by decision tree classifier. The blue dots indicates the flights arrived on-time and the red dots indicate the flights delayed. As we can see, the dots seem to be linearly separable in three dimensions which explains why logistic regression and a simple single-layer neural network was able classify high accuracy.

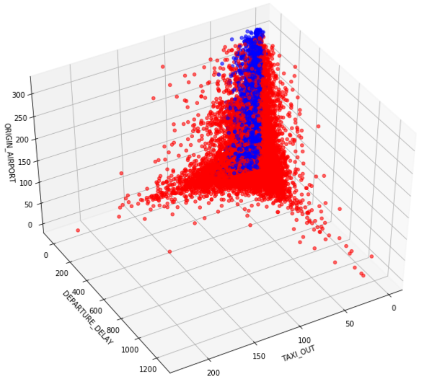

Fig. 4. Scatter plot of top-3 features. Blue dots denote flights arriving on-time and red dots denote flights delayed.

Figure 5 shows the receiver operating curves (ROC) for all three classifiers with an area under the curve of 0.96 for the three classifiers. Figure 6 shows the plot of training and dev-set loss and accuracy versus epoch. The fluctuations in the dev-set curve suggests there might an over-fitting problem. Table 2 shows the confusion matrix and classification report for all three classifiers. From the table we can observe that the decision tree classifier performs better at predicting on-time flights whereas neural network performs better at predicting delayed flights. The difference is, however, very small.

TABLE 2
Combined confusion matrix and classification report for all three classifiers. Number of test samples is 30,000 with 14,999 samples for Class 0 and 15,001 samples for Class 1.

	Predicted Class 0 (on-time)			**Predicted Class 1 (delayed)**		
	Decision Tree	Logistic Regression	Neural Network	Decision Tree	Logistic Regression	Neural Network
True Class 0 (on-time)	13,986	13,907	13,733	1,013	1,092	1,266
True Class 1 (delayed)	1,740	1,670	1,531	13,261	13,331	13,470
precision	0.89	0.89	0.90	0.93	0.92	0.91
recall	0.93	0.93	0.92	0.88	0.89	0.90
f1-score	0.91	0.91	0.91	0.91	0.91	0.91

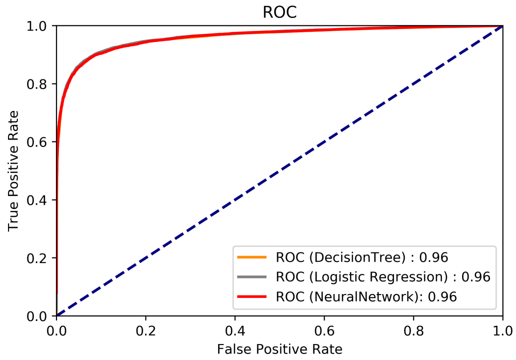

Fig. 5. Receiver Operating Curves for Decision Tree, Logistic Regression and Neural Network models

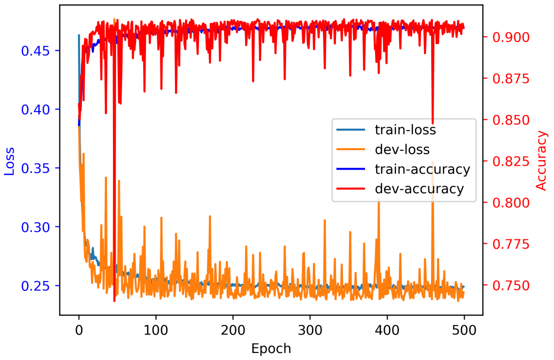

Fig. 6. Plot of training and dev-set loss and accuracy for the neural network in Figure 3

6 Conclusion and Future Work

In this project, we were able to successfully apply machine learning algorithms to predict flight arrival-delay and show simple classifiers like decision tree and logistic regression can predict if a flight's arrival will be delayed or not fairly accurately. For further work we like to further improve our models, perhaps with more training-data or deeper neural network, or both. Taxi-delay prediction is a natural progression to this work, considering amount of fuel wasted while taxiing. Accurate taxi-delay prediction requires taking airport runway and taxiway configurations in to consideration where very little work exists.

7 Contributions

Our initial group consisted three members. Unfortunately, one of the members dropped the class after the mid-term.

7.1 Navaneeth Jamadagni

- Retrieved data-set.
- Wrote software to pre-process dataset, training and testing our model, and tools to investigate model behavior.
- Wrote part of the milestone and project reports, and poster.
- Did poster presentation recording.

7.2 Nathalie Kuhn

- Retrieved data-set.
- Literature research, recommend new models implement and help investigating the results.
- Wrote part of the milestone and project reports, and poster.

References

[1] C. Cetek, E. Cinar, F. Aybek, and A. Cavcar, "Capacity and delay analysis for airport manoeuvring areas using simulation," *Aircraft Engineering and Aerospace Technology*, vol. 86, no. 1, pp. 43–55, 2013. [Online]. Available: https://doi.org/10.1108/AEAT-04-2012-0058

[2] K. B. Nogueira, P. H. Aguiar, and L. Weigang, "Using ant algorithm to arrange taxiway sequencing in airport," *International Journal of Computer Theory and Engineering*, vol. 6, no. 4, p. 357, 2014.

[3] R. R. Clewlow, I. Simaiakis, and H. Balakrishnan, "Impact of arrivals on departure taxi operations at airports," 2010.

[4] H. Balakrishnan, "Control and optimization algorithms for air transportation systems," *Annual Reviews in Control*, vol. 41, pp. 39–46, 2016.

[5] Y. Jiang, X. Xu, H. Zhang, and Y. Luo, "Taxiing route scheduling between taxiway and runway in hub airport," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[6] N. Pyrgiotis, K. M. Malone, and A. Odoni, "Modelling delay propagation within an airport network," *Transportation Research Part C: Emerging Technologies*, vol. 27, pp. 60–75, 2013.

[7] I. Simaiakis and H. Balakrishnan, "Queuing models of airport departure processes for emissions reduction," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, vol. 104, 2009.

[8] K. Gopalakrishnan and H. Balakrishnan, "A comparative analysis of models for predicting delays in air traffic networks," in *USA/Europe Air Traffic Management Seminar*, 2017.

[9] "Bureau of transportation statistics." [Online]. Available: https://www.transtats.bts.gov/ONTIME/Departures.aspx

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[11] P.-N. Tan *et al.*, *Introduction to data mining*. Pearson Education India, 2006.

[12] A. Ng, "Cs229: Machine learning lecture notes," *Standford University Lecture*, 2011.