Omar Farooq

Sharbesh Adhikari

CS 484 Project

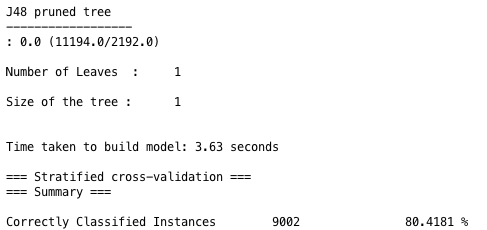
**Predicting Flight Delays Using J48 and Naïve Bayes**

**• Abstract**: Predicting flight delays can be accurately predicted given very little amount of information about the flight. Using data mining, we can find ways to make improvements on flights scheduling so that people can expect their flight to be on time, and make it to their destination with as little hassle as possible. Making flights on time as accurately as possible could reduce costs of flight rescheduling in the long run and save airliners a lot more money.

**• Introduction**: We wanted to predict flight delays as accurately as we could. We thought this would be a good way to help people get to where they are going with as much ease and enjoyment as possible. We wanted to predict this using data that is given to a passenger before they head onto the flight. Such attributes we would include would be pre flight data like Departure time, arrival time, origin, airline, etc. We also wanted to compare our data with different algorithms that we would implement from scratch, and compare our algorithms with other algorithms in Weka.

Originally, we predicted our flights to be delayed if the distance of the flight was large. However, we found that distance was not a good measure of predicting flight delays. We found that just the airline, original airport you flew out of, and the departure and arrival times are our most important attributes we would need to predict flights with a relatively high accuracy. This paper, will describe our findings, and what our best approach would be.

**• Related work:** This problem is similar to what a lot of people have been trying to solve. We were surprised that we even found a relatively high accuracy of flight delays. We were wondering why other companies have not implemented this approach. So we did some research. After running our dataset on Weka, we found that based on no rules, and no decision tress at all, Weka automatically assigned each flight to test its data on as ‘On time’. Using this approach, almost 80% of all the flights were on time. Which means that any set of flight data we would download from the Department of Transportation would have 80% of all of its flights on time. The figure below shows this finding.



Even if the flight is delayed, almost half of the flights are delayed because of weather related reasons.

We found that many people have also seen similar findings as ours. KnowDaily.com also predicts flights, however they use attributes such as weather (they ask their users to submit their flight within 3 days before their flight to return back an accurate finding). Our approach will give people accurate result days/months before your flight. Their approach would give an even higher accuracy if their flight would be delayed or not if they submitted their flight within 3 days before their flight.

Airbnb does a lot of their data analysis using RandomForest, which is what we used for our data. We could not find out more on what Airbnb was doing with RandomForest because they don’t give out a lot of their own information, but we found it interesting that they also liked what we were using as well.

**• Solution:**

We were able to get up to 87% accuracy just using where the original flight departed from, the airline, the departure time, and arrival time to determine if a flight would be delayed or not. We did a lot of data preprocessing which we describe in the Experimental section below, implemented Naïve Bayes and the J48 in python, and ran our data on Weka’s algorithms. We found that Random Forest which is an Ensemble Method with J48 decision tree classifier which gave us the 87% accuracy below. We describe more on these results in the experimental section. did you solve the problem? Describe the technical approach. Tell us what method/algorithm did you use, develop or extend and how did you implement it.

**• Experimental**

**• Experimental results:** Describe your experimental results. Structure your experiments around particular aspects of your method. For example, you could structure the experiments as follows: (1) a table showing results of your method using different types of features; (2) table comparing the performance of your method to the baselines; (3) a graph plotting the size of the training dataset vs. the time it takes to train the model; (4) Investigation of the learned model (what are the important features, etc.).

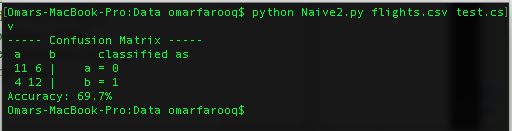
**• Data:** We got our data from the Department Of Transportation (DOT). They give out every single domestic flight for the past decade. We decided to get the past 5 years’ worth of flights. We downloaded them one month at a time, saved it as a csv file, and used python scripting to combine all of the months into one document named ‘data.csv’. From there, we wrote another python script to just print out every 100th line, so that our data would be small enough to fit into our computer’s memory. This outputted file was named ‘flights.csv’ and had about 30,000 flights, and was a size of 300kb.

**• Experimental setup:** We used a lot of software to get an accurate and readable amount of data. We used Excel to format our data into csv. With our formatted data, we got it to fit into Weka. In Weka, we found a lot of our data had to be normalized and modified. We used a combination of Weka, and python scripting to preprocess our data.

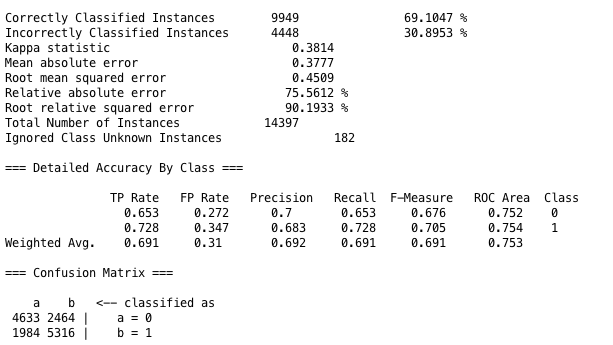
On Weka, we used SMOTE to balance the total number of delayed flights, and on time flights to be equal, so that we didn’t have one class dominating another. We also originally had up to 109 attributes that we had to reduce. We decided which attributes to reduce by having a few factors, such as any post-flight data, Weka’s “Select Attribute” feature to detect which attributes to remove using BestFirst CfsSubsetEval algorithm, calculating the information gain with our J48 algorithm to find the attributes with the lowest gain and removing them.

After that we resampled our data, just to make sure our data was accurate (random). We got had to remove any floating point numbers Weka automatically calculated for us that we had to covert into integers. After all of this, we had our data to be accurate as we possibly could.

**• Experimental results:** To solve our problem, we wrote our own algorithms and compared them with Weka’s. We used Naïve Bayes and tested it with Weka’s own implementation of Naïve Bayes. For our version of Naïve Bayes, we tested our data using training data, and got consistent results with Weka’s Naïve Bayes approach. Our results are displayed below:

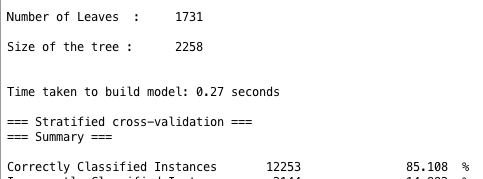


our own implementation of our Naïve Bayes using training data for testing options

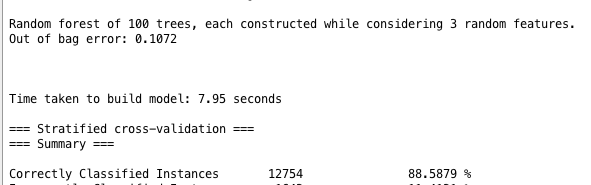


Weka’s Implementation of Naïve Bayes using training data for testing options

\*We also got similar results using cross validation



Weka’s J48 Algorithm



Weka’s Random Forest

\*The J48 may have a more complicated model, which would have a high variance of findings

**• Brief conclusion**

Using simple data that is available online, we were able to find accurate findings for predicting flight delays. This can not only help the airline industry, but also reduce time, energy, and resources to help move airline industries towards the future to better improvements, technologies, and over better flight experiences.

Partner contribution