Planespotting

Using machine learning algorithms to predict flight delays at time of ticket purchase

Samuel Delaney
CSE 773C
University of Nevada Reno
Reno, NV USA

***Abstract*—Flight delays plague all travelers, and most people assume that it comes with the territory, but what if we could predict whether a flight was going to be delayed at time of ticket purchase? If we could accurately predict within fifteen minutes of actual flight delay would we be able to purchase flights with confidence knowing that we would not be troubled by missing lay overs, staying at airports for hours on end, and other travel woes. By searching through many travel information sources we are able to remove data about flights and come up with a list of features that will aid us in creating a flight delay predictor. To create this predictor Weka will be used as the driving force for its ease of use and ability to access a wide variety of techniques to allow us to quickly identify which learning algorithms will be best suited for the task at hand. While other research has been done to predict whether flight delays will occur or not, it has been shown that this is relatively easy to predict so this paper will focus on the ability to predict flight delays in 10 min intervals. The second aspect of this challenge is to be able to accurately predict delays with information only available to a passenger at any time of ticket purchase. This second constraint makes predictability even more difficult as mush of the information we wish to use is not available to use as it is either not available this early or kept from purchasers for some other reason. Early theories are that simpler learning algorithms such as J48 trees would provide the best results while more complicated learners would provide negligible improvement, as this seemed to hold true for just predicting delay and not delay time intervals. As it turned out however some of the best accuracy was achieved with more complicated classifiers such as Kstar achieving 66% accuracy and Naïve Bayes which was able to predict 10 minute delay intervals with a near 70% accuracy. It was with these results that we can see that the threshold of accuracy of flight delay is indeed a problem we can solve with machine learning as accuracy only improves with more information or by increasing the threshold of our delay intervals.**

I. INTRODUCTION

- Flight delay is something that most of us are all too familiar with. We book travel assuming that things are going to follow a set schedule and plan accordingly, but while a delay of a mere 10 minutes might not set us back, what about an hour? Two hours? This is when travel plans start to become nightmares as travelers rush to adjust plans at the last minute; Running to catch flights that if missed will ruin the weekend or cause a missed business meeting. While no one claims to know the future, this topic of predicting flight delay has not gone passed by the scientific community at large. Many researchers have delved into the idea that given enough information we may be able to predict how long it will take for the wheels to leave tarmac. Previous work has been done in the field regarding whether or not it is possible simply to predict whether a flight will be delayed at all, regardless of how long the delay is fairly easy to deduce, as most flights aren't delayed at all by the FAA's standard of flight only being labeled as delays if fifteen minutes has passed since the scheduled departure time.

- In addition to the research aspect this research is also big business for companies that wish to sell apps that predict flight delays. Also, the use of accurate flight delay predictions could help many travel planning sites give better flight options to users, such as the ability to foresee missed flights due to delays, or providing the user potential flight delay times and allow the user to decide if the delay prediction would merit a change in flight purchase.

- Most of these apps boast a delay prediction with high ninety percent accuracy which is far superior to most research results and while they keep their prediction methods secret it is not surprising that these prediction are only available a short time before a flight, some with only six hours of advance knowledge. Without knowing the exact details of their prediction software it isn't too hard to replicate their results by just plugging in more information as it is more readily available such as tail number, whether the flight is currently delayed at another airport etc.

-As it seems that predicting both whether a flight will be delayed at all, or given a small window before takeoff have been covered by both research and industry. This paper seeks to see how close we can predict time delay with the most basic information, and if these results will prove to be reliable enough to use in a meaningful way.

- The methods I will be using will all be facilitated by the Weka toolset. Weka is the work horse of this project due to its ease of use, and the ability to try many

different algorithms in a short time to allow us to concentrate efforts on the most promising learners. Weka has also been used by many researchers and appears to be the standard in data classification. Another appealing aspect of Weka is its ability to use Comma Separated Values allowing us to use a simple format to store data. However there were some limitations encountered to do the use of Weka, such as the limited amount of memory the Java virtual machine was able to use without terminating the program in error. This forced that data set to be significantly smaller than the data available, but the ability to overcome this obstacle will be discussed later in the paper.

- The final results of this project were both confusing and interesting. When research was first started the early hypothesis was that a J48 tree would have the best success as it seems a simple decision tree would be able to perform with a high degree of accuracy. With one hour delay intervals this did indeed prove true with a ~95% accuracy rate, but the tree proved trivial, all flights were labeled as delayed between 0 and 60 minutes, which obviously is not very relevant to travelers. However, on ten minute delay intervals the J48 performed much more poorly with only ~52% accuracy. With much trial and error it was with AdaBoost using a decision stump that the highest accuracy of 69.9% was achieved, implying that while ten minute intervals may be too hard of a target to produce results to act on, that perhaps fifteen or twenty minute intervals will provide the accuracy we need to give the traveler and perhaps even the FAA the tool they need to streamline air travel even farther.

- The rest of the paper will go into further detail on the data collected, such as how and where it was collected. How it was cleaned for weka use, and ultimately how it was binned for use by the algorithms. The next section will go into detailed analysis of what algorithms were used to predict flight delays as well as their rough results. Finally this paper will present the formal findings found by each algorithm in addition to the accuracy produced by each algorithm.

II. DATA

- The data collection was made simple due to the fact that air travel is a government controlled system, and as such there is a government database devoted to maintaining all flight data in the United States. By using this database we have access to all aspects of a flight ranging from tail number to destination airport to flight time. The data was available to download into a CSV file which could then, after moderate cleaning, be loaded into Weka. The other benefit from using the Department of Transportation as my data source was the sheer volume of data I had access to. But this turned out to be a double edged sword.

-To get the most accurate prediction possible and to make the project attainable in a semester I choose to focus on one airport, Seattle Tacoma International airport located in Seattle Washington. This airport was chosen simply as a well-known airport which also had a considerable amount of flight data logged with the Department of Transportation. After choosing an airport a time frame of one year was choosen to be the time frame of flight data gathered. Although far more than one year's worth of data was available the sheer volume of one year's worth of flight data overwhelmed the Weka virtual machine so the benefit of grabbing more data became negligible. One year's worth of flights provided roughly 350,000 flight entries to be processed through Weka, but as this amount of data overwhelmed all of the classifiers the data was forced to be cleaned and resampled.

-The process for the resampling was made simple by the same tool that forced the issue. Weka. By using Weka's built in filters the flight data was able to be cleaned to remove and flight that had no data, or missing data. Once the corrupted entries were removed filters were once again reapplied to randomize the data to resample. The resampling sizes were then chosen based on the classifier being implemented as some could handle more data than others. The sample size of all non-boosted classifiers were using a 10% sample size while all boosted classifiers could only handle a sample size of 5% of the years' worth of data. After all resampling was done, a J48 classifier was run to ensure that similar trends were maintained throughout the resampling. Classifiers that required a 1% resample size were not used as at that level of reduction the sample data no longer conformed to that of the original size and as a consequence those results become suspect of a biased or unrepresentative data set.

-While the database does provide almost every possible aspect of a flight, the goal of this paper is to provide accurate results at time of ticket purchase, so we will be reducing the feature list considerably as most of the information pulled from the database is unknown to the user until the day of the flight or if at all. With that in mind the features we will be using will be

- Airline Provider
- Date/Time of Flight
- Destination Airport
- Length of Flight
- Day of Week
- Airport of Origin

An additional reason to limit ourselves to this data is that hindsight is often 20/20 and by using all known flight attributes prediction becomes rather easy with a J48 decision tree providing an accuracy of ~95%, but again this makes the problem to easy and the application of the tool irrelevant. So by removing all the features save the above mentioned we see that the accuracy rates drop to roughly ~56%, a significant decrease. I would be remiss however to ignore the fact that while the other flight data features will not be used in the final aspect of this research, testing was indeed done with these features included to see if any features were so useful that delay prediction would be pointless without them. Oddly enough, only one feature seemed to dominate the others. That feature was the previous delay of a flight, and upon review it is obvious why this feature should feature so prominently. If an flight is already delayed upon arrival it is quite simple to accurately say that the flight will maintain within roughly 15 minutes the same delay for its departure flight and the rules and trees produced will this feature was included in the data set support this as well.

```
J48 pruned tree
------------------

ARR_DELAY = '(-inf-20.1]': '(-inf-44]' (17096.78/8.0)
ARR_DELAY = '(20.1-97.2]': '(-inf-44]' (1594.85/675.0)
ARR_DELAY = '(97.2-174.3]': '(114-184]' (238.28/81.28)
ARR_DELAY = '(174.3-251.4]': '(184-254]' (54.06/15.06)
ARR_DELAY = '(251.4-328.5]': '(254-324]' (14.02/4.02)
ARR_DELAY = '(328.5-405.6]': '(394-464]' (3.0/1.0)
ARR_DELAY = '(405.6-482.7]': '(394-464]' (7.01/2.01)
ARR_DELAY = '(482.7-559.8]': '(534-604]' (2.0/0.0)
ARR_DELAY = '(559.8-636.9]': '(464-534]' (1.0/0.0)
ARR_DELAY = '(636.9-inf)': '(604-inf)' (2.0/0.0)

Number of Leaves  :     10

Size of the tree :      11
```

Figure 1. J48 tree with Arrival delay feature in data set

- It becomes aberrant that the arrival delay feature is so dominating that all other features are not considered but we are left with a small clean decision tree, however, if we remove that one feature from our data set the decision tree, even with an arbitrary ten binning system becomes a complicated rule set as seen in the tree below.

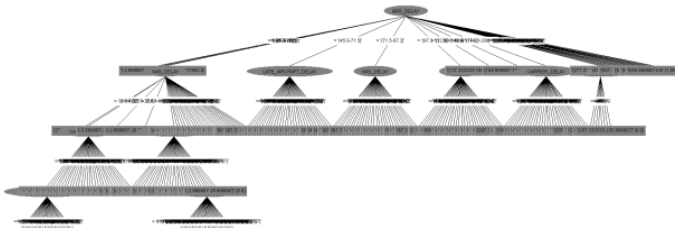

Figure 2: J48 tree without ARR_DELAY feature

We can see that the tree is so large that it cannot be presented in a readable size! However this was exactly what we wanted, this showed that our problem was not as trivial as we initially saw and that there indeed was some meaningful research that could be done on this data.

- With the resampled data in hand we now had the tools with to begin our research. The next section will detail the many classifiers used to try and classify the flight data and interpretations on why some classifiers performed better than others.

III. RESULTS

. -The basis of all the results and how the determination of success versus failure for a particular classification method was decided to run against a ZeroR classifier. The ZeroR classifier simply classifies every instance as the one most common of the training data set. This provided us with a baseline classification accuracy of 45.65%. This accuracy was achieved on a ten minute interval bin where there were 50 bins available for classification. The bin that the ZeroR classifier choose as the most common bin was the No Delay Bin which consisted of the time delay of 5 minutes early to 5 minutes delay.

-The first classifiers that were used to classify the data were all non-boosted classifiers that ran on the 10% resampling. The first of which was the J48 decision tree which produced a tremendous tree with over 1,100 leaves. While the tree was indeed large it did contain specific characteristics. The tree seemed to be broken into carrier subtrees that would then assign a delay based on destination and date of travel, hinting at an association that delay could be heavily dependent on service provider. Though, even with the great amount of leaves the classification accuracy of this tree was only 52.82% providing considerable improvement over ZeroR but still was below a usable threshold. The next classifier that was used was the Random Forest classifier. For the random forest 10 trees were created using 6 random attributes. The trees were allowed to have unlimited depth. The accuracy results proved disheartening with an accuracy of only 48.17%, only a few points better than a simple ZeroR classification. With this last result it did seem to suggest that while the J48 tree was large and not entirely accurate, that rules would perhaps win the day over random classification algorithms.

-This lead to the use of the JRIP rule classifier to see whether the assumption of rules prevailing held true. The JRIP was expected to produce a large rule set as the J48 tree was quite large but surprisingly it produced a mere 12 rules as seen below.

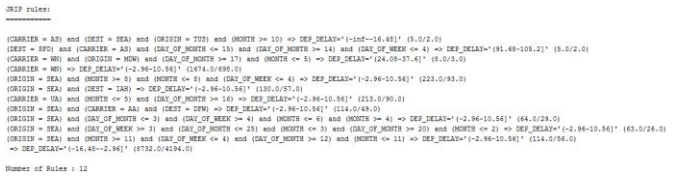

Figure 3: JRIP classification rules

And while it only generated 12 rules it did perform with 51.22% classification accuracy on the tenfold validation. Similar to the J48 tree, the airline carrier seemed to be a main contributing factor on delay prediction, but with the JRIP certain airlines were singled out while others were ignored as opposed to the J48 creating a sub tree for each carrier.

-Continuing to pursue the simple set rules to classify the data the next approach was to use a Decision Table. The decision table performed better than the other rule based classifiers with an accuracy of 54.48% but it wasn't enough of an increase to justify a breakthrough. Again however the decision table generated two distinct rules in order to predict delay, the first rule again was the airline carrier, once again reinforcing the notion that the provider is one of the most important features available. The second feature in the decision table was the destination of the flight. The flight destination had indeed shown up in both the J48 tree and in some degree the JRIP rules, but it become more obvious as the decision table was the best performing classifier so far the perhaps destination played more of a role than previously thought.

-While it was easy to assume that rules based classifiers had won the day, it was with disappointment to see that even the best performing was only providing a rough ten percent increase in performance over a simple ZeroR classifier. It was at this point it became obvious that boosting would become the next step, but before we put the boosted rules to the test it seems worthwhile to ensure that some of the other classifiers that we had discussed in class had been given a fair shake at the attempt to classify this data set.

-The first classifier chosen was to try a Naïve Bayes classifier but the expectation of performance was to be close to that of the random forest. However the performance was a shocking 67.58% accuracy rate. This was the performance breakthrough that we were looking for even though it seemed to break with the current logic that a rules based classifier was essential to get a real accuracy performance. With this new breakthrough another classifier K* was used and produced an accuracy of 66.8%. Again the current hypothesis was turned on its head. However, it was time to give boosting a try to see if anymore performance could be found.

-As previously stated by moving to the boosting methods the current dataset of a 10% random sample proved too much data for Weka to classify. It was at this point that the data was reduced further to a 5% sampling of the original dataset of 350,000 flights providing the boosted classifiers a data set of ~17,500 flights.

-The first boosting method that was used was to use the J48 trees. This classifier consisted of 3 subcommittees, run at 10 iterations and each tree having roughly 1,200 leaves. This configuration provided an accuracy of 64.49% rivaling that of both the Naïve Bayes and K8 classifiers. It would seem at this point that rules were making a comeback. The next boosted system used was AdaBoost using a decision stump or one leaved trees. The accuracy of this model proved to be the best with 69.9% accuracy, a performance that could not be bested by any other classifier run on the dataset. The following chart displays the accuracy of each classifier comparing the results.

Classifier	Accuracy
J48	52.82%
JRIP	51.22%
Naïve Bayes	67.58%
Decision Table	54.48%
K*	66.80%
Random Forest	48.71%
ZeroR	45.65%
AdaBoost	**69.90%**
J48 Boost	64.49%

Figure 4: Accuracy Summary

-Further attempts to use boosted methods failed as Weka was not able to run these specific classifiers on a data set so large. In order to run the additional classifiers Weka required the dataset be less than 1,000 instances which created a dataset that did not well represent the overall data. The J48 performance on this small set dropped to a meaningless 28% and all other attempts to classify had equally poor results.

IV. Conclusion and Future Work

- The results gathered from the data were interesting indeed. While the initial theory of a rule based system being the predominate factor was proved to be mostly true in the end, it did have to rely on boosting were as other methods provided equally compelling results in their own right. It also leaves one to wonder that had

boosting been possible would they have created even better outcomes. It is at this point that I would like to make mention of the fact that while we were concerned with 10 minute bins, most of the classifiers discussed here did not miss by much. Each misclassified class was ever only one bin away from being correctly classified. This again is in step with the previous testing showing that with large 1 hour bins it is trivial even with a ZeroR classifier to achieve upwards to 95% accuracy. The important aspect of this discovery shows that with a little more time and access to more advanced classifiers, would the performance been able to be enhanced even further for ten minute bins. Due to software and time constraints however these questions will have to remain for another day and we will have to be satisfied with the high sixty percentile range for accuracy, though I remain positive that with more time and the ability to process larger data sets accuracy could well be pushed into the high seventy if not low eighty percentile range for accuracy.

- Beyond the initial scope of classifying the delay times, it was noted late in the assignment that what if the delayed flight were isolated from the flights that had no delay or left on time. Then using this new data set what would we see? I decided to act on this and using my same data set from the previous results set about to strip all flights that had little to no delay time, namely all flights that had delays of fifteen minutes or more to see what features if any changed from having all the flights included.

- The results extracted from this approach were not as compelling as one might have hoped. It would still seem that the same rule sets were being applied labeling both Carrier and Destination as the largest factors in predicting flight delay. This is not unexpected as when looking at the data directly one can see that certain airlines have significant more delays than others, but as this study was only concerned with SeaTac airport it is not fair to say that this is a true statement for the same carrier in another city.

- Future work for this problem would most certainly include an improvement in the ability to process more data. This dataset was indeed large, but it only represented one years' worth of data. Who knows what other features might have emerged had a decade's worth of data been available to process in the classifier. Would holidays' make an appearance? What about trends over time as people adjust to changes in the industry? I feel that without this improvement it is too early to make a definitive statement on whether or not delay prediction is indeed predictable on a large scale time basis. Another factor would be as our ability to predict weather improves will that help delay predictions? It is indeed common knowledge that weather plays a factor in airline travel, but the ability to know the exact weather patterns months ahead of time is currently unavailable. However, as current physics models become more accurate perhaps a weather feature could be added to the above dataset to help improve accuracy. Another key aspect of airline prediction would be the widening of the net so to say by including other airports in the dataset. While certain trends did seem to occur it is impossible to say that they would hold true in any other airport without conducting similar tests on other hubs.

ACKNOWLEDGMENT

I would like to acknowledge Dr. Sushil Louis, the University of Nevada Reno, and the Weka developments team, who without their support and efforts this project would not have been possible.

REFERENCES

[1] Lu, Zonglei "Alarming Large Scale of Flight Delays: an Application of Machine Learning"

[2] Deshpande, Vinayak Arikan, Mazhar "The Impact of Airline Flight Schedules on Flight Delays" Manufactoring & Serice Operations Management Volume 14 Issue 3, Summer 2012 pp 423-440

[3] V Sud, M Tanino, J Wetherly, M. Brennan, M. Lehky, K. Howard, R. Oiesen "Reducing Flight Delays Through better Traffic Management" Interfaces Colume 39 Issue 1, january 2009 pp 35-45

[4] Stefanski, Tim "Predicting Flight Delays Through Data Mining" cs-people.bu.edu/dgs/courses/cs105/hall_of_fame/timoteo.html