Predicting Flight Delays with LAMP and a Sliding Window

Adrian Lim A0123432W
Ng Hui Xian Lynnette A0119646X
Toh ZiJing A0123506R

Abstract

Have you ever been stuck in an airport because your flight was delayed or cancelled and wondered if you could have predicted it if you'd had more data? In this project, we used 7.5 million entries worth of commercial flight data within USA to answer the following question: Given a current flight's data and the past few days of flight data, can you predict if the flight will be delayed? The 1.04 GB of flight data is obtained from RITA, and combined with meteorological data from NOAA as well as latitude and longitude data. As per the RITA definition of delay, a flight is considered to be delayed ifs it is more than 15 minutes late. This project explores using different window sizes as the number of previous days of training data to predict a query flight. It extracts k-nearest-neighbours from the training data, where k is a fraction of n records of training data, before performing various classifiers on the new dataframe. Using the map-reduce architecture, this project parallizes computation with multiple mappers computing the classification label of IsDelayed and one reducer aggregating the results to formulate the precision and recall score for the parameters n, k and the classifier. It reports a precision of 90% at best with a gradient boosted regressor and $k = n/3$

Introduction

Have you ever been stuck in an airport because your flight was delayed or cancelled and wondered if you could have predicted it if you'd had more data? ASA Statistical Computing hosted a Data Expo competition in 2009 (asa 2009) on Airline on-time performance. The challenge of the Data Expo is to provide a graphical summary of the flight data consisting of flight arrival and departure details from all the commercial flights within America from October 1987 to April 2008. The dataset contains about 120 million records in total and is 12GB large. A total of nine entires were received, and many made use of tools such as R, Hadoop and Spark.

In this project, we make use of the flight data for Year 2007 and try to answer the following question: Given a current flight's data and the past few days of flight data, can you predict if the flight will be delayed? We use the RITA definition of delay, where a flight is considered to be delayed if it is more than 15 minutes late. This project explores different number of days of past data used for training (henceforth referred to as window size), and its effect on the precision and recall of classification.

Literature Review

Xu (Xu et al.) used Naive Bayes to construct a Dirichlet prior distribution in order to estimate the delay in the American national aviation system. The accuracy of their method is around 81%. (Rebollo and Balakrishnan 2012) used Random Forest algorithms built upon temporal and spatial variables over a 2 hour prediction horizon, with an error rate of 19%. However, they classified delays as 60 minutes and not 15 minutes as per the RITA definition. (Smith and Sherry) had approximately 83% success rate in using Support Vector Machines (SVM) to predict the aircraft arrival rates with flight data augmented with meteorological data.

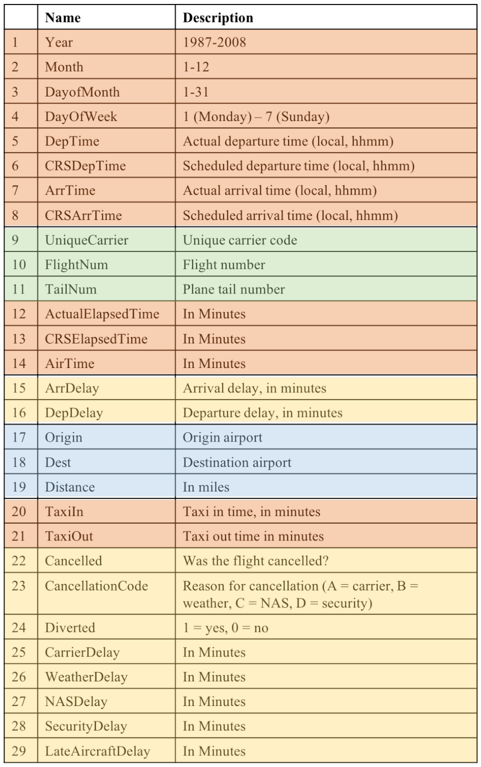
Some student prediction projects using this dataset include Predicting Flight Delays through Data Mining(Stefanski 2009), where data from Year 2008 was used for prediction. No additional data added to the dataset, and a variety of classifiers such as the NB Tree, J48 Decision Tree, PART, Decision Table and OneR classifiers from WEKA were used on the dataset with an average prediction accuracy of 62.6%. Stanford students Lawson and Castillo (Lawson and Castillo 2012) combined the dataset with NOAA weather data and performed binary classification on the dataset using Naive Bayes, Random Forest and SVM. Their best precision was 86% with Random Forest on a selected test set from Newark Liberty International Airport. With random query flights, their average prediction accuracy is 85%.

Commercial Sites

There are commercial sites that are built to benefit passengers by predicting flight delays. For commercial flights within USA, there are two main such sites. One of them is Flightcaster, which claims a precision of 85% and a recall of 61%.(Flightcaster 2015) Another site is KnowDelay (KnowDelay 2015), but its precision-recall rate is not reported

The Data

The flight dataset is taken from RITA, the United States Department of Transportation. For this project, we used only the data from Year 2007, which contains over 7.5 million flight records and is 1.04 GB uncompressed. Figure 1 shows a table of the values in the data and has been colour coded with into four categories: spatial data (blue), carrier data (green), temporal data (orange) and delay statistics (yellow).



	Name	Description
1	Year	1987-2008
2	Month	1-12
3	DayofMonth	1-31
4	DayOfWeek	1 (Monday) – 7 (Sunday)
5	DepTime	Actual departure time (local, hhmm)
6	CRSDepTime	Scheduled departure time (local, hhmm)
7	ArrTime	Actual arrival time (local, hhmm)
8	CRSArrTime	Scheduled arrival time (local, hhmm)
9	UniqueCarrier	Unique carrier code
10	FlightNum	Flight number
11	TailNum	Plane tail number
12	ActualElapsedTime	In Minutes
13	CRSElapsedTime	In Minutes
14	AirTime	In Minutes
15	ArrDelay	Arrival delay, in minutes
16	DepDelay	Departure delay, in minutes
17	Origin	Origin airport
18	Dest	Destination airport
19	Distance	In miles
20	TaxiIn	Taxi in time, in minutes
21	TaxiOut	Taxi out time in minutes
22	Cancelled	Was the flight cancelled?
23	CancellationCode	Reason for cancellation (A = carrier, B = weather, C = NAS, D = security)
24	Diverted	1 = yes, 0 = no
25	CarrierDelay	In Minutes
26	WeatherDelay	In Minutes
27	NASDelay	In Minutes
28	SecurityDelay	In Minutes
29	LateAircraftDelay	In Minutes

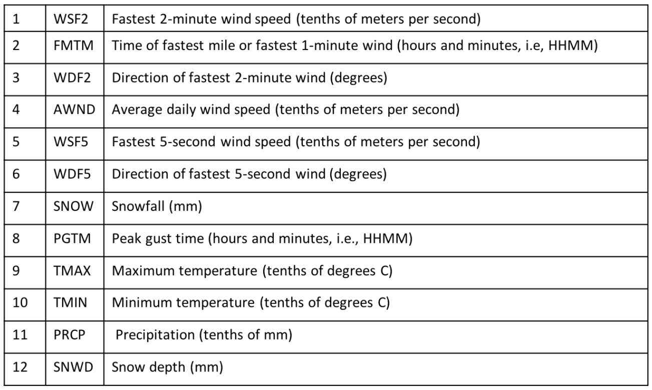
Figure 1: Fields in the Flight Data

We combine the data with weather data from NOAA (National Oceanic and Atmospheric Administration), which provides temperature, precipitation and wind speed data collected from the various airport weather stations. Figure 2 shows a table of the fields in the weather data set.

We also add data about the distances between the origin and destination airports, calculated from the latitude and longitude of the airports.

Exploratory Data Analysis

In this section, we present certain findings and graphs generated from the raw data itself. This helps us with our feature selection in the preprocessing step. In year 2007, 2 244 514, or 30% of the flights were considered delayed. On average, there are about 20, 407 commercial flights per day within USA. There are 5000 unique flight routes, or origin-destination pairs, a fraction of which is represented in Figure 4. We also chart the percentage of delays of the airport when they are the Origin airport or Destination airports. Figure 5 show the 20 airports with the most number of delays, by airport code. Most are municipal airports, hence it is acceptable for them to experience high delays as they have a small land space and lesser resources yet have to still cope with the volume of inbound and outbound flights. International airports that are in this top 20 list include those with much passenger traffic, such as Atlantic City International Airport and John F Kennedy International Airport. A similar chart in Figure 3 depicts delays experienced by outbound flights and the cause of delays.



1	WSF2	Fastest 2-minute wind speed (tenths of meters per second)
2	FMTM	Time of fastest mile or fastest 1-minute wind (hours and minutes, i.e, HHMM)
3	WDF2	Direction of fastest 2-minute wind (degrees)
4	AWND	Average daily wind speed (tenths of meters per second)
5	WSF5	Fastest 5-second wind speed (tenths of meters per second)
6	WDF5	Direction of fastest 5-second wind (degrees)
7	SNOW	Snowfall (mm)
8	PGTM	Peak gust time (hours and minutes, i.e., HHMM)
9	TMAX	Maximum temperature (tenths of degrees C)
10	TMIN	Minimum temperature (tenths of degrees C)
11	PRCP	Precipitation (tenths of mm)
12	SNWD	Snow depth (mm)

Figure 2: Fields in the Weather Data

Number of delays by Day of Week as depicted by 6 do not have much deviations, as the average number of delays on each day is 32 000 flights. Figure 7 presents a star chart on the number of delays from the origin to destination airports. The lines which are most strong are generally major international airports. Some airports are Chicago O'Hare International Airport, JFK International Airport and San Francisco International Airport.

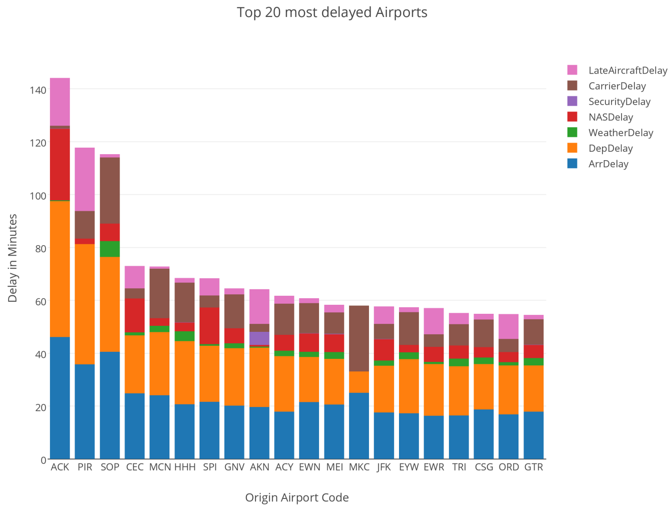


Figure 3: Chart of Delays by Origin Airports (i.e. represents outbound flights)

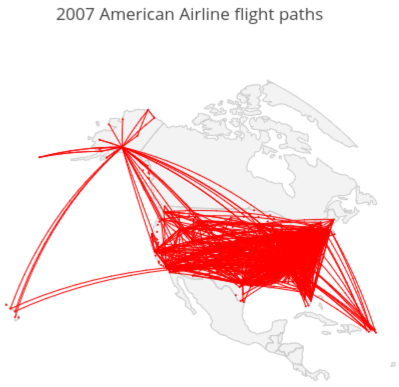


Figure 4: Chart of Delays by Airport

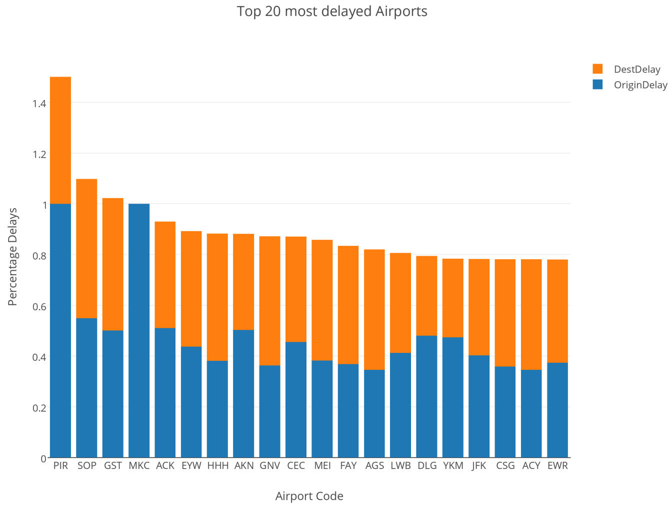


Figure 5: Chart of Delays by Airport

Predicting Flight Delays

This section describes the workflow we used to predict flight delays. The flight delays were predicted via a sliding window. This sliding window is suitable for such a time-series data to simulate real-time prediction, in which we have a certain number of previous day's worth of records and have to predict the flight delay on the current query flight.

We used python and scikit-learn for classification and prediction. For visualisation, we used Plot.ly and matplotlib libraries. In order to process the huge dataset, we used Hadoop's map-reduce framework with multiple mappers and one reducer on the school's Tembusu Compute Cluster.

Overview

Figure 8 shows a graphical flow of our data mining pipeline.

The raw flight data is first combined with weather data obtained from NOAA and flight distances calculated from the latitude and longitude of the origin and destination airports. The merged dataset is then preprocessed for feature agglomeration for better representation of the data. In our pipeline, we make use of the map-reduce architecture. The mappers will perform find k-Nearest-Neighbours to the query flight among the n training examples they receive. Then, they will train a classifier on the k-nearest-neighbours. This project explored various classifiers. The mapper outputs the actual and predicted classification of delay. One reducer collects all the classification labels and calculates the result statistics for the dataset.

We perform a sliding window classification for this time-series data. 3 tests are run. First iterate through the window size n from 20 000 data records, which represent the number of flight data from one day, to 5 million records, or 80% of the dataset. Each increment consists of 1 day's worth of flight data, around 20 000 records. From this we can find out the effect of window sizes.

Secondly we fix the window size n to 60 000 and vary nearest neighbours k from 100 to 10000 to find the effect of k.

Lastly we fix n and k, but drop weather and distance data, to find out if the additional information helps prediction.

Preprocessing

Combining weather and distance data The first step of the preprocessing pipeline is to combine the raw flight data with weather data. The weather data is obtained from NOAA in a summary of meteorological data obtained from the airport weather station. The average meteorological data of each day over the Years 2007 and 2008 is obtained and merged with the flight data. There are two main problems in these datasets which require some data clean-up: missing data and inaccurate data. Some airports do not have weather stations or may not report a certain data field, which results in missing data. In the case where data from other airports in the state are present, we use the mean of the state. Where the mean of the state is not available, we use the mean of all the airports in USA. In cases where we can intelligently guess, we will fill in the data fields with our external knowledge. For example, McCarran International Airport (LAS) does not report precipitation data. As we know LAS is situated in Las Vegas which is a very dry state, we fill in the precipitation as 0 mm, since the place is generally known to be desert-like. On inaccurate data, we will remove the data field and attempt to fill it up as per missing data, i.e. mean of state. Santa Barbara Airport (SBA) has no temperature readings but reported maximum and minimum temperatures as -9999.0. This messes up the dataset and is therefore removed.

After this, we combine the flight data with the data on the distances between the origin and destination airports from `http://www.openflights.org`. We obtain the latitude and longitude coordinates of the two airports and estimate the flight distance using the haversine formula, factoring in the radius of the Earth (est. 6373km).

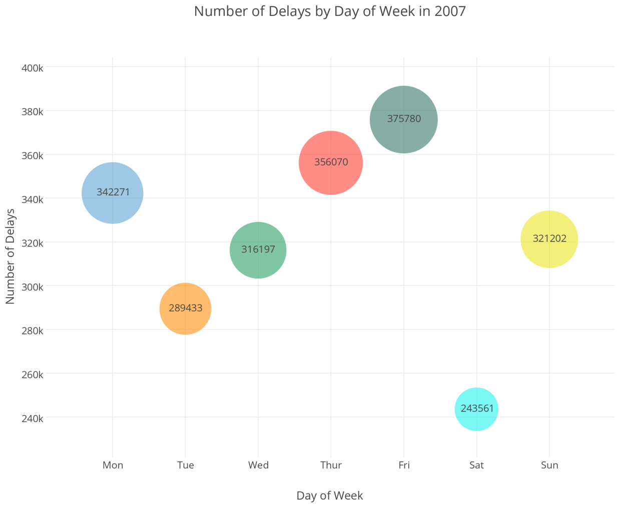


Figure 6: Chart of Delays by Day of Week

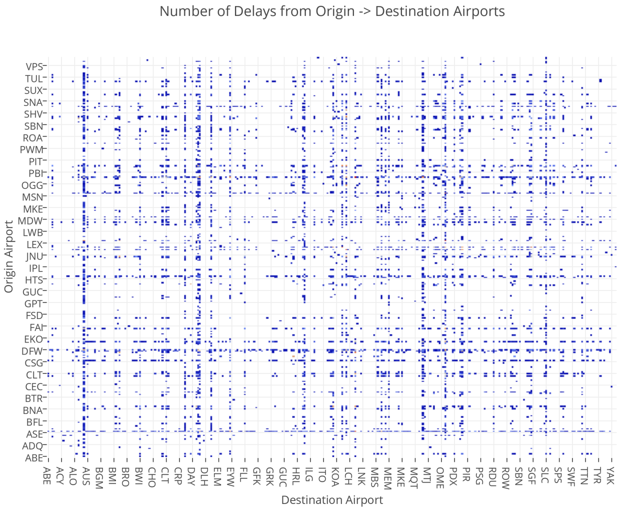


Figure 7: Star chart of delays from origin to destination airports

$$\text{haversin}(\theta)=\sin^2\left(\frac{\theta}{2}\right)=\frac{1-\cos(\theta)}{2},$$

$$d=2r\sin^{-1}\left(\sqrt{\text{haversin}(\phi_2-\phi_1)+\cos(\phi_1)\cos(\phi_2)\text{haversin}(\lambda_2-\lambda_1)}\right) \tag{1}$$

Where ϕ_1, ϕ_2 is the latitude of points 1 and 2, λ_1, λ_2 for the longitude of points 1 and 2, r radius of the sphere and d the distance that we want to find.

We believe flight distance do affect the length of flight delays, because in general, insurance companies give greater financial compensation for long-haul flights than shorter flights for the same amount of delay.

Feature selection After the data is combined, we merge certain columns to better represent the data. We first merge the columns of Arrival Delay, Departure Delay, Carrier Delay, Security Delay, Weather Delay, Security Delay and Late Aircraft Delay to a column called Total Delay. Then we formulate a column IsDelayed as a boolean which is true of the Total Delay is greater than 15 minutes and false otherwise. We merge the Year, Month, Day, Departure Time into a column called DateTime, then sort the data by the DateTime. We also calculate the days from nearest American public holiday such as Christmas or New Year. This is because there tend to be more flights nearer such dates and the airports will get more congested which could result in more delays. Lastly, we convert all the nominal data columns to numeric ones to facilitate classification.

kNN and Classifers

The parameter k in k-Nearest-Neighbours is defined as a proportion of the window size. We define $k = n/2$, $k = n/3$ and so forth, so that each run of the dataset has the same proportion of neighbours. Each window would be used to predict a flight at least 2-4 hours away. The flights from the result of the k-Nearest-Neighbours are put through various classifiers: Gradient Boosted Regressor, Ada Boost and Decision Tree Regressor, Support Vector Machines, Decision Trees and Bernoulli Neural Networks. These classifiers are obtained from the scikit-learn package and their default settings are used, as the focus of the project is to explore the effect of window size rather than classifier parameters.

Validating the Results

We validate the results by calculating the precision, recall and fscore for each predicted set as compared to the actual dataset. This is done using the scikit-learn API. In our situation, we prefer a lower recall rate to a higher precision. False negatives is favoured over false positives. From a consumer point of view, it is better to anticipate a flight delay than to be delayed. When a flight delay is anticipated, alternative transport routes or plans can be made. The passenger can prepare to take a later connecting flight, and it is in his favour that the flight lands on time. However, if the flight is delayed when our prediction algorithm reports otherwise, the passenger miss the connecting flight or appointments.

Map-Reduce

We made use of Hadoop on the Tembusu Computer Cluster, which is run on Linux, to parallize the classification process in order to speed up computation. With Hadoop, iterating through the entire 7.5 million dataset with a window size of 20 000 takes roughly 2 minutes for the Gradient Boosted Regressor. A full iteration from 1 day worth of training data record to 5 million training records took over 8 hours. Other classifiers, however, took different amounts of time. Our map-reduce stack is implemented with multiple mappers and one reducer.

Mapper As we are performing a sliding window over the time-series data, each mapper only needs to know the n training data and the one query flight. The training data

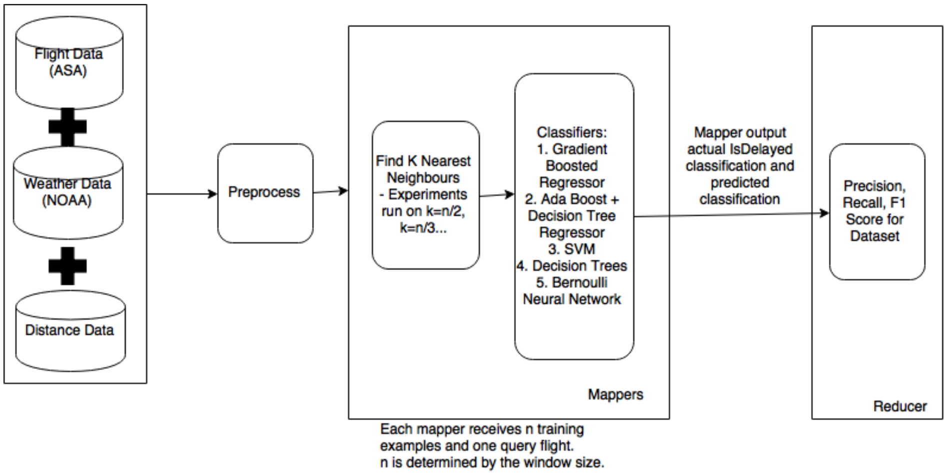

Flight Data (ASA)
Weather Data (NOAA)
Distance Data
Preprocess
Find K Nearest Neighbours - Experiments run on k=n/2, k=n/3...
Classifiers: 1. Gradient Boosted Regressor 2. Ada Boost + Decision Tree Regressor 3. SVM 4. Decision Trees 5. Bernoulli Neural Network
Mapper output actual IsDelayed classification and predicted classification
Precision, Recall, F1 Score for Dataset
Mappers
Reducer
Each mapper receives n training examples and one query flight. n is determined by the window size.

Figure 8: Overview of prediction pipeline

contains of the past n records of flight data, which ranges from one day worth of records (20 000) to 5 million, with an increment of a day at each iteration. The mapper will find k-nearest-neighbours to the query flight from the training data, where k is a proportion of n. Then it will classify the k-neighbours using one of the classifiers and output the actual classification label and the predicted classification label.

Reducer There is only one reducer in each run of the dataset. The reducer takes in the actual classification label and predicted classification label and outputs the precision, recall and F-score for the n and k parameters.

Results

We ran a binary classification prediction problem. Given a window size n training flights, we want to classify whether the query flight will be delayed or not. We ran this over several classifiers and regressors. The regressors will output an estimated number of minutes of delay for the query flight, in which we classified the flight to be delayed if the estimation were above 15 minutes, and not delayed otherwise.

For the algorithms, we used Gradient Boosted Regressor (GBR), Ada Boost with Decision Tree Regressor, (ADR) Support Vector Machines (SVM), Decision Trees (Tree) and Bernoulli Neural Networks (BNN), with their default parameters.

We did 3 tests, one by varying window size n, with the number of nearest neighbours k as a fraction of n. For each k, we ran the algorithms from $n = 20000$ to $n = 5000000$, each time increasing n by 20 000. These numbers are chosen because 20 000 is the average number of commercial flights per day and 5 000 000 comprises of 80% of the flights. Hence, we create a window consisting of the flight for a single day to 80% of the flight data given, each time increasing the number of training data by the number of flights per day. We term one full run as running through $n = 20000$ to $n = 5000000$ for one k. From this we would be able to find out the effect of different window sizes from the dataset.

	GBR	ADR	SVM	Tree	BNN
Precision	0.82	0.24	0.59	0.83	0.45
Recall	0.66	0.29	0.59	0.81	0.50
F-Score	0.69	0.20	0.57	0.80	0.45
Approx Single build time	0.38s	0.68s	9.2s	0.3s	5.5s
Approx Total run time	5h	9h	5days	4h	3days
Query Time	<1s	<1s	<1s	<1s	<1s

Table 1: Average scores for $k = n/2$

The Tables 1 and 2 shows the average precision, recall and f-score for each algorithm ran. The approximate single build time for building one model using one window size worth of data is recorded, and the total time taken to run one full iteration through the dataset. Despite expensive build times, the query time for each model are extremely fast. These timings are measured using the Tembusu Compute Cluster. This shows that the average optimal window sizes are around 60 000.

For the second test we fixed n at 60 000 which is the average optimal window size with GBR, while varying nearest neighbours k to find out the effect of k on the results. Figure 10 shows the results, showing k around 4600 being the most optimal, with precision, recall and fscore scores of 0.88, 0.81 and 0.81 respectively.

Lastly we show that the additional data improves prediction - Table 3 shows that adding weather data and/or distance significantly improves prediction by around 20%.

Effect of Different Algorithms

Tree-based classifiers such as Decision Tree Classifier, Ada Boosted Regressor and Gradient Boosted Regressor classified the data very quickly. Running for one window size through the entire dataset took around 30 minutes. For one full run of varying window size with a fixed proportion of k neighbours, these classifiers took 4 hours to four days to classify the data. Classifiers such as SVM and Neural Net-

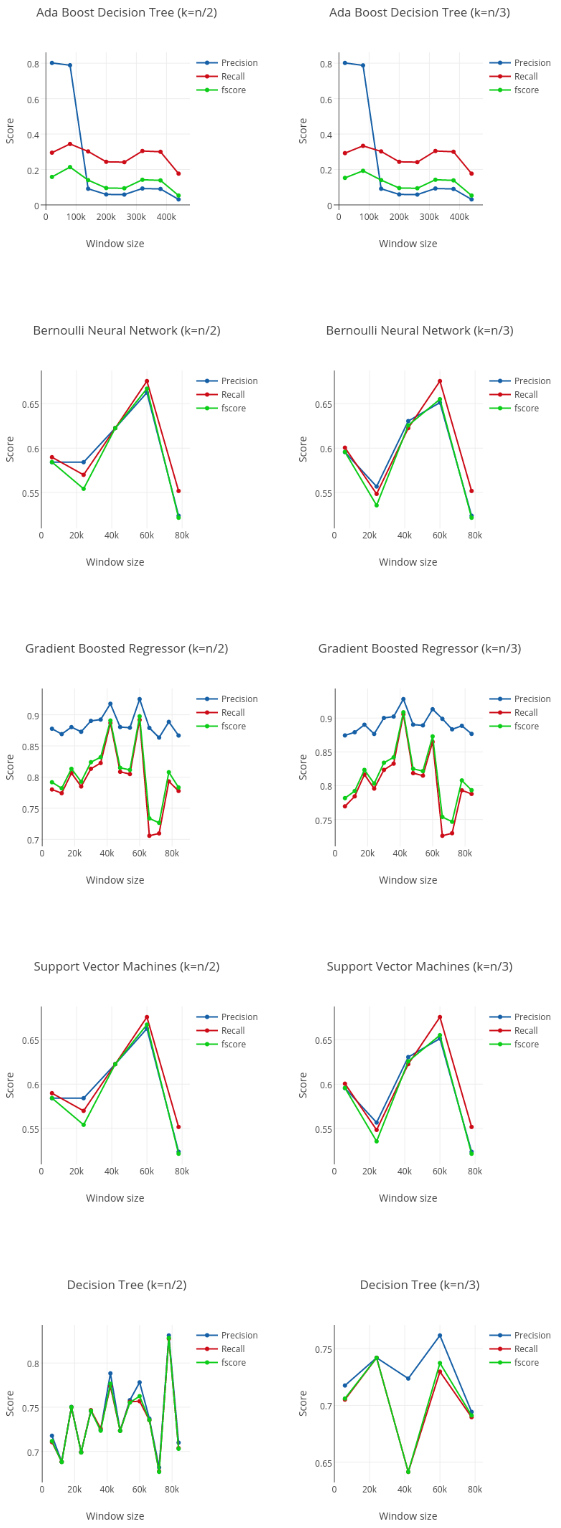


Figure 9: Scores for regressors with different window sizes.

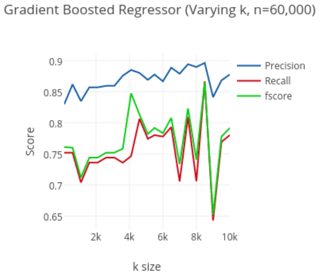


Figure 10: Scores from varying k.

	GBR	ADR	SVM	Tree	BNN
Precision	0.90	0.25	0.59	0.78	0.62
Recall	0.79	0.27	0.60	0.75	0.59
F-Score	0.81	0.16	0.59	0.75	0.58
Approx Single build time	0.3s	0.6s	7.3s	0.3s	5.5s
Approx Total run time	4h	8h	4days	4h	3days
Query Time	<1s	<1s	<1s	<1s	<1s

Table 2: Average scores for $k = n/3$

	Original data	+Weather	+Distance	+Both
Precision	0.62	0.88	0.84	0.88
Recall	0.65	0.82	0.78	0.80
F-Score	0.61	0.83	0.79	0.81

Table 3: Effect of additional data on scores for fixed $n = 60000$ and $k = 4600$

works have been known not to scale very well to this large dataset, and they took approximately days for one full iteration.

A gradient boosted regressor performed best when k is $1/3$ the window size with an average precision of 0.90, followed by a decision tree classifier. The Ada Boost performed worst with a precision of 24% and the neural network also fell below 24%.

Effect of Window Sizes

In general, the smaller the window size yielded better results. This shows that the most recent flight data is most useful in predicting whether the query flight will be delayed. Hence, programs and sites which do flight delay prediction may reduce space complexity by not needing to store too many flights.

Effect of k

From our experiments, decreasing the window size used stabilised and improved the precision recall score. This points to the effectiveness of kNN in reducing the set of items within a large dataset to its most similiar ones.

Effect of Additional Data

By adding correlated data such as weather and flight distance improved the predictions by around 20%.

Feature Selection

At best, the classifiers reported results of about 90% precision. This shows that the algorithm chosen has lesser importance as compared to the feature selection. To achieve good prediction results with this dataset, we need to pick out the salient features for classification. We tested a dataset without distance data with the Gradient Boosted Regressor, and the dataset had slightly lower precision of 80% as compared to 82% with distance as a feature, for $k = n/2$. This shows that the distances of flights do make a difference in prediction of delays.

As compared to the literature where no additional data is merged into the flight data, our method achieved a performance of 10% better precision. The additional features of weather and distances are useful in determining the possibility of delay of the flight. Our results are on par with commercial flights delay sites and literature.

Further Work

Much can be done to explore the effect of varying window size with the accuracy of predicting flight delays: different sizes of k, constant k with varying window size, different classifiers, different classifier parameters, and even different window size and window increment sizes. One might want to look into kernel methods that scale for using kernel-based classifiers such as SVM for classifying such a huge dataset.

One can also explore which features are most useful in flight delay prediction. This would be done using logistic regression of each feature versus the delay vector. However, logistic regression on such a scale is very slow. An experiment that we ran involved a logistic regression of maximum temperature vs number of minutes of delay, which fitting the data to the logistic regressor took more than a day, and the reported correlation value was only about 0.2.

One can also explore using the full dataset of flights from 1987 to 2008 with weather data. However, NOAA only provides weather data from 1900 onwards, so one has to disregard weather data from Years 1987 to 1900. To do such an experiment, one needs to come up with a more efficient way of scaling the experiment, with a variation of our map-reduce architecture.

Conclusion

This project explored a binary classification prediction problem with data from commercial flights within America in Year 2007, consisting of 7.5 million records. From the results we find that the average optimal window size is about 3 days (60 000 records), with *some k*.

Adding additional data like weather and distance also helps in improving predictions.

We find k nearest neighbours prunes large datasets down to the most relevant records required for prediction, therefore making slow classifiers fast to run, while being accurate.

With several regressors and classifiers on a map-reduce architecture on the Tembusu Compute Cluster, our method reported an average precision of 90% at best with a gradient boosted regressor and $k = n/3$ which is on par with commercial sites and better than some literature.

References

2009. Airline on-time performance. `http://stat-computing.org/dataexpo/2009/`.

Flightcaster. 2015. Flightcaster.

KnowDelay. 2015. Knowdelay. `http://www.knowdelay.com/`.

Lawson, D., and Castillo, W. 2012. Predicting flight delays. `http://cs229.stanford.edu/proj2012/CastilloLawson-PredictingFlightDelays.pdf`.

Rebollo, J. J., and Balakrishnan, H. 2012. A network-based model for predicting air traffic delays.

Smith, D. A., and Sherry, L. Decision support tool for predicting aircraft arrival rates, ground delay programs and airport delays from weather forecasts.

Stefanski, T. 2009. Predicting flight delays through data mining. `http://cs-people.bu.edu/dgs/courses/cs105/hall_of_fame/timoteo.html`.

Xu, N.; Donohue, G.; Laskey, K. B.; and hung Chen, C. 1 estimation of delay propagation in aviation system using bayesian network.