Flight Delay Prediction

Karthik Chandranna
chandranna.k@husky.neu.edu
Northeastern University

Naveen Aswathanarayana
aswathanarayana.n@husky.neu.edu
Northeastern University

1. Introduction

Air travel has become an affordable and convenient mode of transportation. For both domestic and international travel, air is the preferred mode. Air travel is no longer a luxury, with number of airlines increasing every day. The competitiveness in providing economical and faster mode of transportation has made air travel cheaper. With increase in number of customers for air travel, flight delays and cancellations become inevitable part. Every year around 20 to 40 billion dollar is at loss for customers, airlines and society which are related to delays [5]. Although flight patterns are extremely difficult to analyze, there are many factors which affect the delay. The main goal of this project is to use large historical dataset of United States domestic airlines to predict whether a future flight will be delayed without the use of weather data.

The past research on flight delay prediction emphasize more on combining weather data with the flight details [3], [4]. Although weather plays an important factor in flight delay, we will be analyzing other predictors such as air traffic at the airport, date and time of the scheduled flight and check whether any customized predictors will increase our accuracy. Since the data is in the order of millions of records, we will be using MapReduce to parallel process the huge dataset. We will be using two learning algorithms i.e. k Nearest Neighbors and Naive Bayes for prediction.

2. Problem Statement

The main approach taken by the past researchers to determine the flight delay prediction was to emphasize more on the weather data [3], [4], and [5]. The importance given to the flight data has been minimal and the factors related to flight itself was considered to have no significant impact. We wanted to determine if the flight details by itself can be used to make prediction without any weather data. Weather data in most cases are reliable utmost a week in advance. Hence prediction delays for future flights in advance is not reliable and might change unpredictably.

For example if we consider flight originating from New York, which is one of the busiest airports. The air traffic at such airports are complex to manage, this might impact the flight arrival and departure, which leads to delays. These factors do not depend on the weather data, hence combining both the data might lead to state where one data has significant impact over the other. We have tried to analyze the impact of different factors on flight delay and how these predictors impact the flight delay.

3. Data

The historical data has been taken from United States Department of Transportation, Bureau of Transportation Statistics [1]. This dataset contains details such as time, airline, origin, destination, departure performance, arrival performance and flight summaries. The entire dataset contains 12 million records. We noticed inconsistencies in the obtained dataset, hence it required cleansing. The data cleansing consisted of removal of malformed data and removal of data which do not conform to sanity conditions which are listed below [1].

CRSArrTime and CRSDepTime should not be zero
timeZone = CRSArrTime - CRSDepTime - CRSElapsedTime;
timeZone % 60 should be 0
AirportID, AirportSeqID, CityMarketID should be larger than 0
Origin, Destination, CityName, State, StateName should not be empty
For flights that is not Cancelled:
ArrTime - DepTime - ActualElapsedTime - timeZone should be zero
if ArrDelay > 0 then ArrDelay should equal to ArrDelayMinutes
if ArrDelay < 0 then ArrDelayMinutes should be zero
if ArrDelayMinutes >= 15 then ArrDel15 should be true

3.1 Features Affecting Delay

The dataset contains large amount of features but we have to make sure to take the right set of features which make significant impact on the delay prediction. From the available historical dataset we tried to bring out the factors which will impact the flight delay as displayed in Figure 1. From Figure 1(a) we can see that delays are more frequent from 6PM to midnight, hence hour of day could make good factor for predicting delays. Delays are more frequent for flight with shorter distances, around 500 miles 1(b). We have taken three cities with different variation in flight delays based on either it is the origin or destination. City Detroit (DTW) has more delays when it is the originating city and lesser delays when it is the destination. From 1(c) and 1(f) we can conclude that airport or city also has significant impact on flight delay. Monday and Friday can be related to have more travel since people using air travel for work and holidays are high on these days, thus days of the week is also one of the important factor 1(e). Although day of the month has almost linear projection of delay, one can clearly say 12th of every month has higher chance of flight being delayed than any other day. We might not related this factor to any possible explanation, but this brings out hidden impact on flight delay as shown in 1(d).

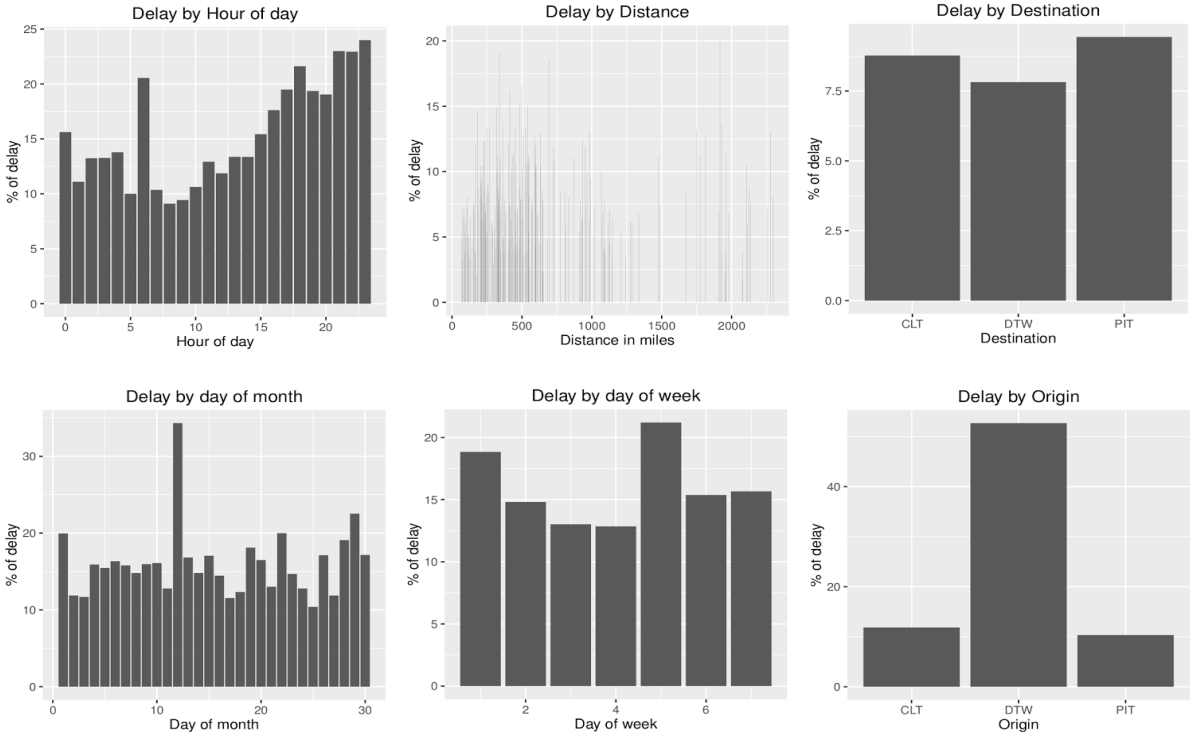

Figure 1

4. Methodology

We have implemented two learning algorithms namely Naive Bayes and k-Nearest Neighbors for the flight delay prediction.

4.1 Naive Bayes

In machine learning, Naive Bayes classifiers are a family of simple probabilistic classifiers based on Bayes' theorem with strong (naive) independence assumptions between the features [8].

The fundamental concept of the Naive Bayes classifier is the Bayes' theorem given by –

$$p(C_k|\mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x}|C_k)}{p(\mathbf{x})}$$

- P(C|x) is the posterior probability of class(target) given predictor (feature)
- P(C) is the prior probability of the class
- P(x|C) is the probability of the predictor given class
- P(x) is the prior probability of the predictor

The probability/likelihood of a class is dependent on all the features and is given by -

$P(C|X) = P(x_1|C) * P(x_2|C) * P(x_3|C) * P(x_4|C) * \ldots\ldots * P(x_n|C) * P(C)$

In our project, for the airline data, the likelihood of flight being delayed is dependent on all the features stated in section 3.1 and is given by -

*$P(Yes_i|X)$/Likelihood of Yes = $P(DayOfWeek=dofW_i|Yes)$ * $P(DayOfMonth=dofM_i|Yes)$ * $P(Origin=o_i|Yes)$ * $P(Destination=d_i|Yes)$ * $P(Distance=dist_i|Yes)$ * $P(HourOfDay=hofD_i|Yes)$ * $P(Yes)$*

Similarly, the likelihood of flight not being delayed is given by -

*$P(No_i|X)$/Likelihood of No = $P(DayOfWeek=dofW_i|No)$ * $P(DayOfMonth=dofM_i|No)$ * $P(Origin=o_i|No)$ * $P(Destination=d_i|No)$ * $P(Distance=dist_i|No)$ * $P(HourOfDay=hofD_i|No)$ * $P(No)$*

We compare the value of *P(Yes|X)* and *P(No|X)* and select the one with higher probability as the predicted value.

Note that, all the features selected for the Naive Bayes model has numerical data which has to be converted to categorical data. This can be achieved by computing normal distribution for each feature, given by -

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\sigma^2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Where **σ** is given by-

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}, \text{ where } \mu = \frac{1}{N}\sum_{i=1}^{N} x_i.$$

4.2. k-Nearest Neighbors

K-Nearest Neighbors algorithm classifies the train data based on some similarity measure for example distance functions. KNN has been widely used in pattern recognition methods for classification and regression. The class of given test data is classified based on majority of the

class of its k neighbors. The data is classified by a majority vote of its neighbors, with the data being assigned to the class most common among its k nearest neighbors [8].

Airline dataset contains different factors with different magnitude, hence it is important to standardize the train data. We are finding out the minimum and maximum value of each feature from the training data and find the standardized variable from the formula.

$$X_{i,\ 0\ to\ 1} = \frac{X_i - X_{Min}}{X_{Max} - X_{Min}}$$

For airline dataset we have employed Euclidean distance function to determine the nearest neighbors.

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

d(train,test) = d(train,test) = sqrt.((train(DayOfWeek) - (test(DayOfWeek)) ^ 2 + (train(DayOfMonth) - (test(DayOfMonth)) ^ 2 + …… + (train(Origin) - (test(Origin)) ^2))

Train Dataset					
DepartureTime	**DayOfWeek**	**……..**	**ArrivalTime**	**IsDelayed**	**Euclidian Distance**
0.445696969	0.23256789	……..	0.12345465	Yes	0.2345
0.345787878	0.3475675	……..	0.09876767	Yes	0.3445
0.456666666	0.32436498	……..	0.1456673	No	0.3256
0.427020201	0.22364856	……..	0.14484252	No	0.2227
0.43250505	0.25769567	……..	0.15594885	Yes	0.2532
0.437989898	0.255069906	……..	0.16705517	No	0.2537
0.443474747	0.247703568	……..	0.1781615	Yes	0.2443
0.448959595	0.24033723	……..	0.18926782	No	0.2448
Test Data					
DepartureTime	**DayOfWeek**	**……..**	**ArrivalTime**	**IsDelayed**	
0.46575768	0.2347595	……..	0.1234533	????	
k=3	IsDelayed = Yes				
K=1	IsDelayed = No				

Optimal value of K is chosen by inspecting the data and testing the model with different values as shown in figure below.

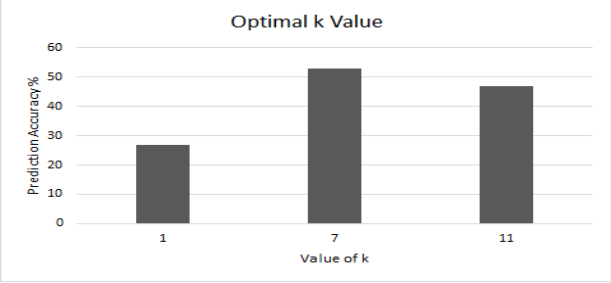

Figure 2

5. Novelty

We wanted to determine if the flight details by itself can be used to make prediction without any weather data. Weather data in most cases are reliable utmost a week in advance. Hence

prediction delays for future flight is not reliable and might change unpredictably. Our main aim is to provide flight delay prediction which is reliable based on flight details alone. Further we wanted to have these predictions generated at the time of flight booking, so that user can make smart selection to avoid flights with probable delay.

6. Analysis and Design

6.1 Requirement Analysis - Functional Requirements

1. The system should accept and cleanse the dataset in mentioned CSV format for training and testing.
2. The system should extract the required features from the cleansed dataset.
3. The system shall provide predictions based user model selection (KNN, Naive Bayes, Weka KNN, Weka Naive Bayes)
4. The system should generate models based on Naive Bayes and k Nearest Neighbors algorithms from the training dataset.
5. The system should use the generated models to predict flight delays.
6. The system shall generate models based on Naive Bayes and k Nearest Neighbors algorithms using the Weka library.
7. The system should use the generated models to predict flight delays using Weka library.
8. The system should validate the prediction results with the validation data and generate prediction accuracy and confusion matrix.
9. The system shall provide prediction accuracy which is better than a coin toss.
10. The system shall provide prediction accuracy which is comparable to the predictions obtained from the Weka models.

6.2 Requirement Analysis - Non-functional Requirements

1. The system should be capable of handling data in the order of millions of records.
2. The system should be able to provide results within a span of 30 minutes.
3. The system shall be usable with minimum effort.

6.3 Design Architecture

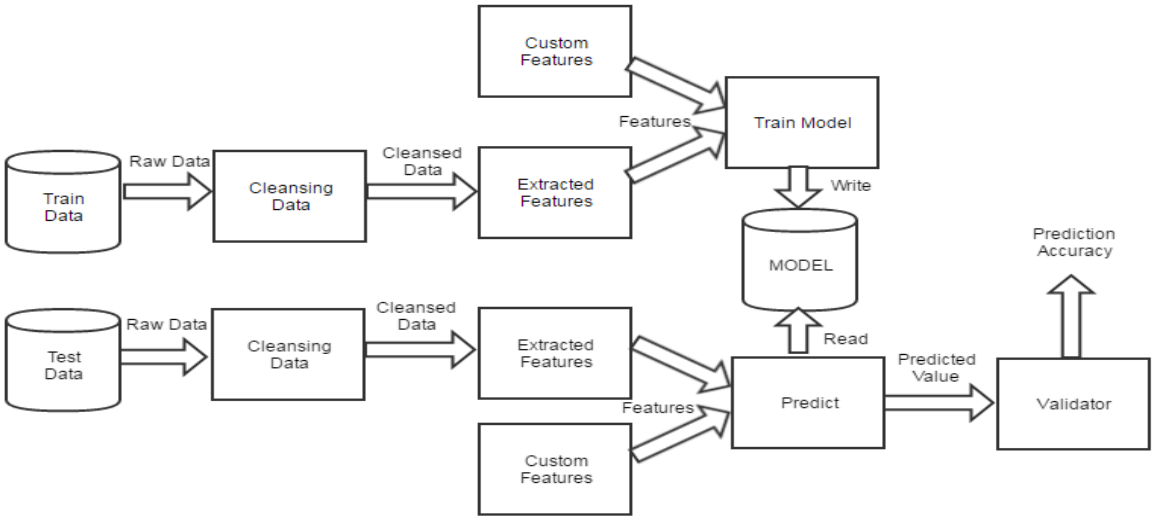

Figure 3

The process model diagram from Figure 3 depicts the flow of control and data through the system. Two branches of process are carried out for training and test data, each by independent MapReduce jobs. First MapReduce job performs the cleansing, feature extraction of training dataset and builds the model based on underlying algorithm. Second MapReduce job performs the cleansing, feature extraction of test dataset and generates prediction for each record. The predicted value is then validated against validation data from Validator process and prediction accuracy is provided.

6.4 Experiments and Results

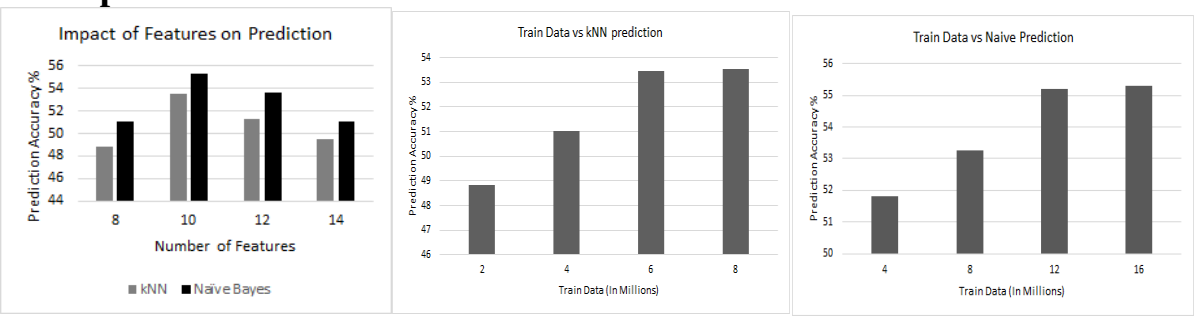

Figure 4

We performed several runs with varied training dataset and features to get the best possible prediction accuracy. Feature sets of size 8 to 14 were used for prediction and we observed that 10 features provided the best accuracy. For kNN classifier, training dataset of 6 million records gave the optimal prediction accuracy, similarly for Naive Bayes classifier, 12 million records gave the optimal prediction accuracy.

6.5 Result Evaluation

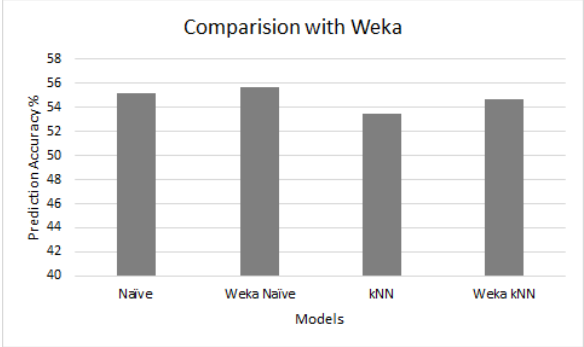

Figure 5

We compared prediction accuracy for our implementation of AI algorithms with Weka implementation. Based on the figure 5 we can observe that our implementation is comparable with Weka implementation.

7. AI Algorithms/Models

We have mainly used 2 learning algorithms which are k nearest neighbors and Naive Bayes'. We have explained how it has been applied to our project in section 4. The below sections provides the detailed work allocation of individual team members.

7.1 Naveen - Work allocation

- Data Cleansing - Raw data consist of high number of rows which does not conforms to right format. Detecting and removing these noise from raw dataset.
- Feature extraction - Finding right attribute from the dataset which makes an impact on delay prediction and implementing derived/custom features
- K Nearest Neighbors algorithm Implementation - Own implementation of KNN algorithm customized for airline dataset which has been explained in section 4.2.
- K Nearest Neighbors MapReduce job - Mappers and Reducers to train the knn model and test the data on generated model.

7.2 Karthik - Work allocation

- Prediction Accuracy Evaluation - Evaluate the predicted values against the actual data. Compute the prediction accuracy based on evaluation.
- Weka Comparison – Build Weka compatible attributes/features. Build MapReduce job to train and test Weka NB and KNN models.
- Naive Bayes algorithm Implementation - Own implementation of Naive Bayes algorithm customized for airline dataset which has been explained in section 4.1.
- Naive Bayes MapReduce job - Mappers and Reducers to train the Naive Bayes model and test the data on generated model.

8. Implementation Tools

- Programming Language: Java 7
- Framework: Apache Hadoop MapReduce 2.6.3
- Dataset: - United States Department of Transportation - http://www.transtats.bts.gov/
- IDE: Eclipse Luna 4.4.2
- Evaluation library: Weka Dev 3.7.2
- Validation tool: Validator.java (own implementation)
- System requirements -
    - Processor - Multi Core CPU with 1.7 GHz- 2 GHz, 64 bit
    - RAM - At Least 4GB
    - Memory - At Least 10GB of free space

9. Discussion

Our main goal of the project was to apply the learning algorithms to predict flight delay without the use of weather data and study the impact of various features. We have identified the appropriate features that influence flight delays. Further, modifying the feature set by adding or removing features or changing the existing features might not provide good results. Overall, the Naive Bayes and k nearest neighbors performed pretty well against their respective Weka implementations. K nearest neighbors takes longer time to predict delays as it involves in-memory comparisons with the train data. Hence the train data size should be restricted to an optimal number of records (around 6 million). Due to this reason, Naive Bayes performs slightly better than K Nearest, as it is trained against a larger data set.

10. Conclusion and Future Work

From our implementation of Naive Bayes and k-Nearest Neighbors learning algorithm on large training dataset, we can infer that simple probabilistic model like Naive Bayes performs better than pattern recognition model like k-Nearest Neighbors. Naive Bayes performs better than k-Nearest Neighbors on accuracy, execution time and space usage. Since k-Nearest Neighbors involves storing entire training dataset and comparing it with test data, it requires lot of computational effort to predict delays. Whereas Naive Bayes has to store only the normal distribution for each feature which it uses to predict delays. Similar probabilistic models can perform better and can be employed for determining the flight delays while booking the air ticket. Related work performed by Raj and Rafael to predict flight delay using weather data gives higher accuracy [3]. Although their work has achieved higher accuracy, the prediction involved the use of weather data which has already occurred. But in reality to predict future flight delays we do not get accurate weather estimates, hence their prediction can be largely skewed. Since our prediction does not involve future weather data we are eliminating this skew and providing a concrete approach to predict flight delays.

Our implementation can result in improved accuracy by using larger training dataset. This can be achieved by employing distributed system to train our model. Rather than using different machine learning algorithms, we believe using data with different features, better domain knowledge can achieve higher accuracy, as all our implementations yielded similar results.

11. Users' Manual

Please find the user's manual, to run this project, in the link below -
https://github.com/karthikchandranna/Artificial-Intelligence/blob/master/FlightDelayPrediction/README

12. References

[1] Bureau of Transportation Statistics (BTS) - http://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236&DB_Short_Name=On-Time

[2] Rebollo, Juan J and Balakrishnan, Hamsa. Characterization and Prediction of Air Traffic Delays

[3] Bandyopadhyay, Raj and Guerrero, Rafael (2012) Predicting Airline Delays.

[4] Naul, Brett (2008) Airline Departure Delay Prediction

[5] Lawson, Dieterich and Castillo, William. Predicting Flight Delays

[6] Zhang, Min-Ling and Zhou, Zhi-Hua. A k-Nearest Neighbor Based Algorithm for Multi-label Classification.

[7] Gu, Zhenmei and Cercone, Nick (2006). Naïve Bayes Modeling with Proper Smoothing for Information Extraction.

[8] Wikipedia references, Naive Bayes Classifier and k-Nearest Neighbors Classifier https://en.wikipedia.org/wiki/Naive_Bayes_classifier
https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm