Assignment A6

Smitha Bangalore Naresh & Ajay Subramanya

March 5, 2016

This project predicts arrival delays of flights using a model that is trained on three years of airline data. A flight is considered delayed if the arrival delay is greater than zero.

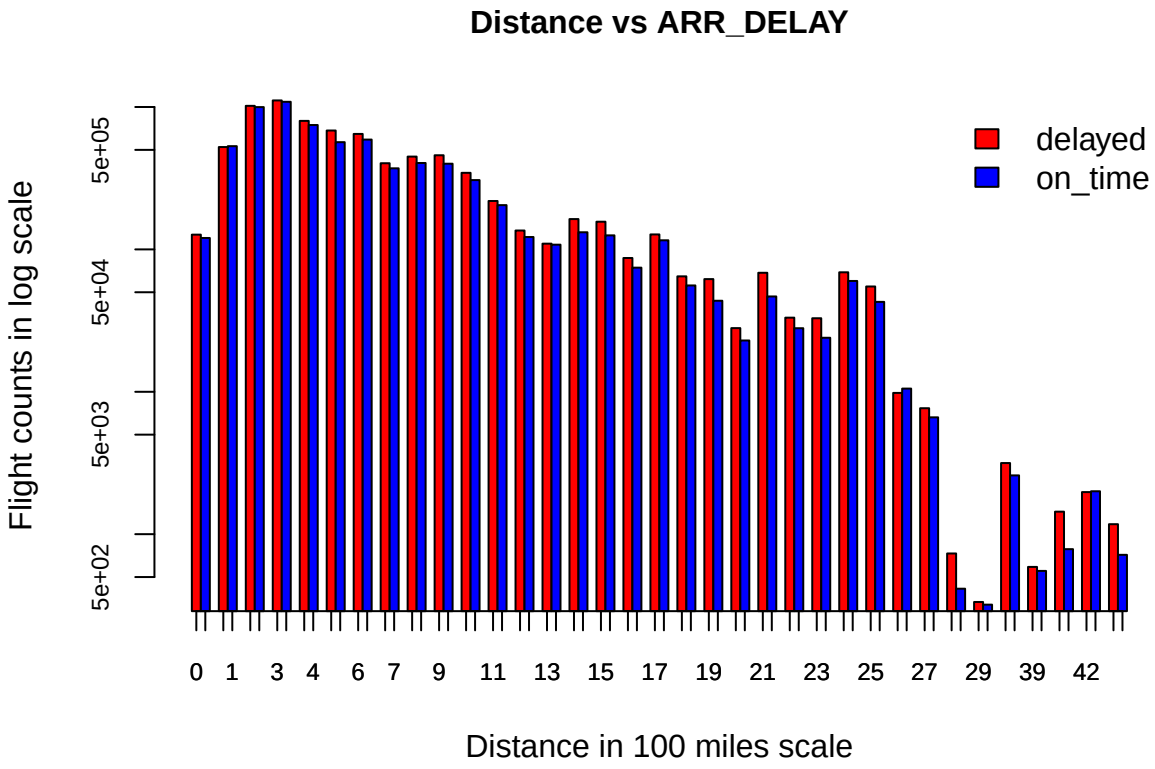
Insight to Data

Given flight dataset for training contains data for 3 years 1995, 1996, 1997. Test data contains data for 1998 year for which prediction of ARR_DELAY has to be done based on previous 3 years.

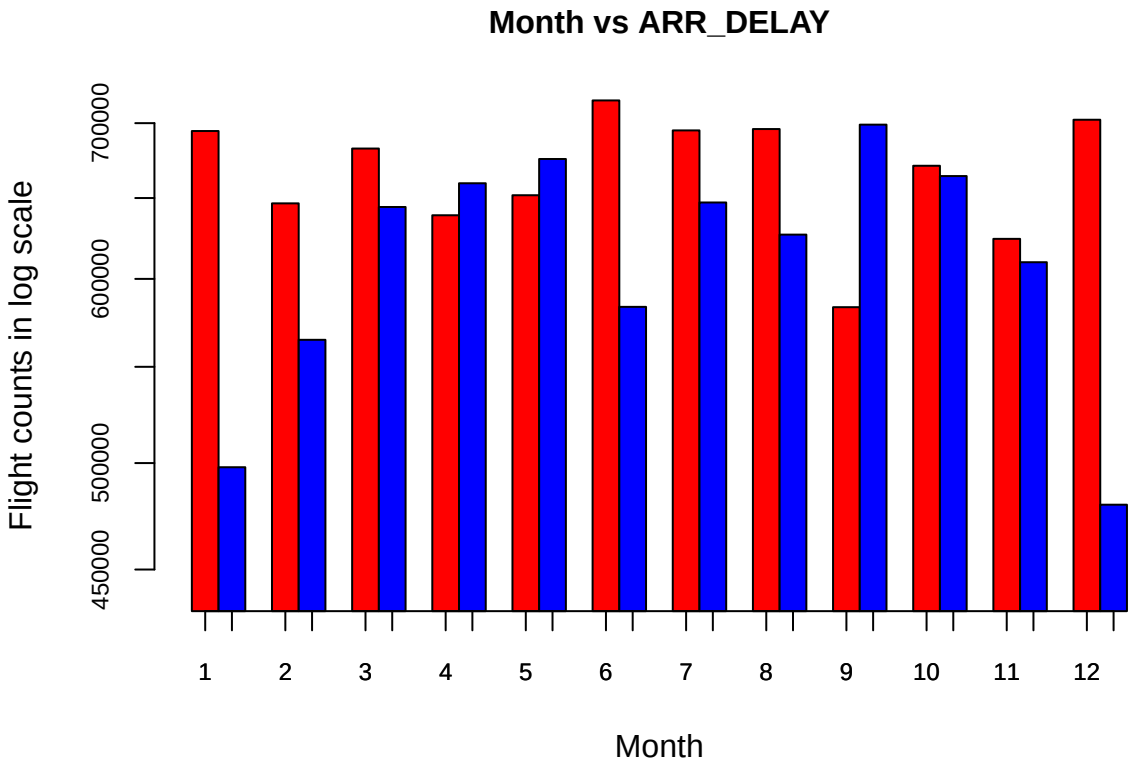
For the training dataset these were delayed and ontime counts.

	delayed	ontime
Full	8002517	7352895
1995	2589708	2497838
1996	2765843	2332631
1997	2646966	2522426

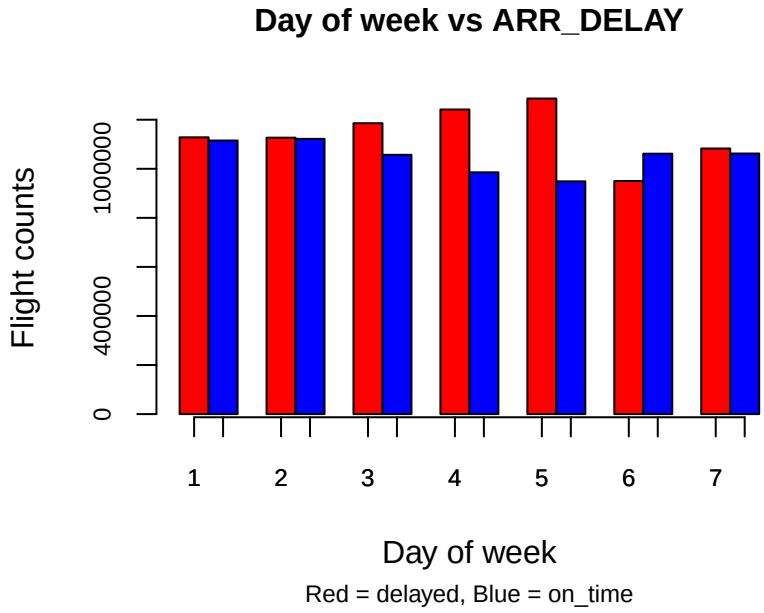
To build a efficient model, we first needed to get insights into the data. We did some analysis on the features that we felt would affect the arrival delay , below are the graphs depicting how ARR_DELAY varies with each feature individually.



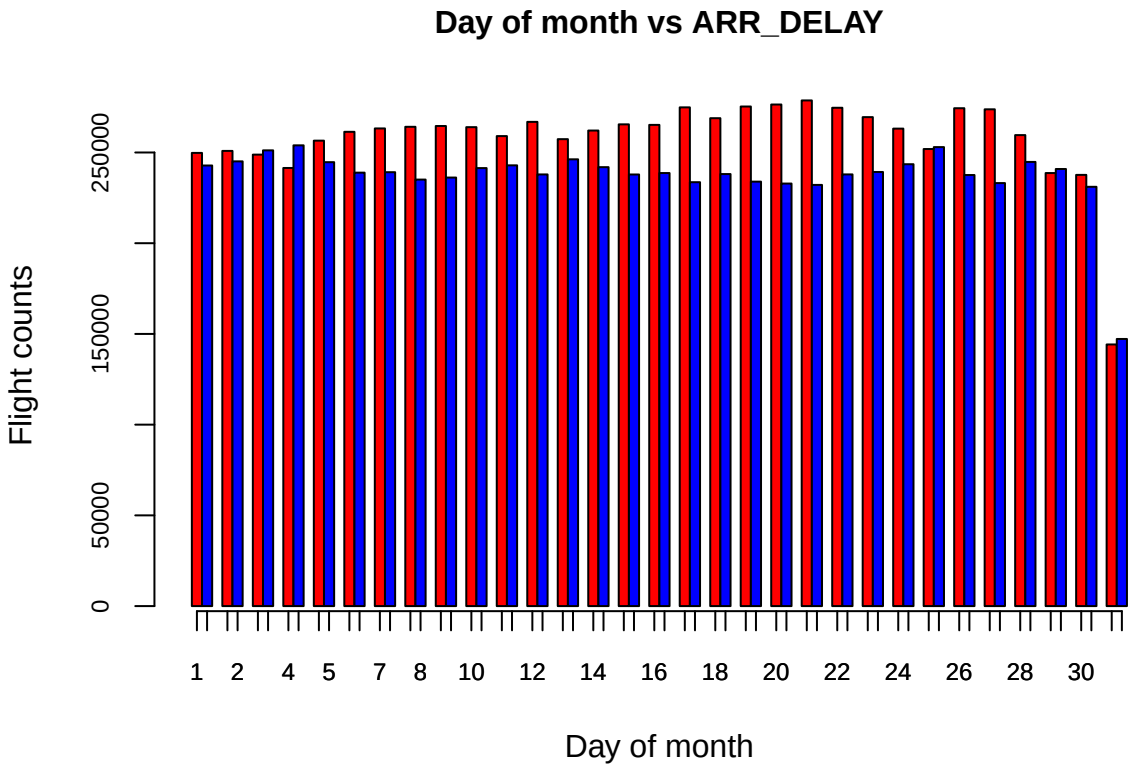
Distance : We were not able to find a consistent pattern with distance which would affect arrival delay, but we were able to understand that majority of the flights delayed were in the range 100 miles to 600 miles.



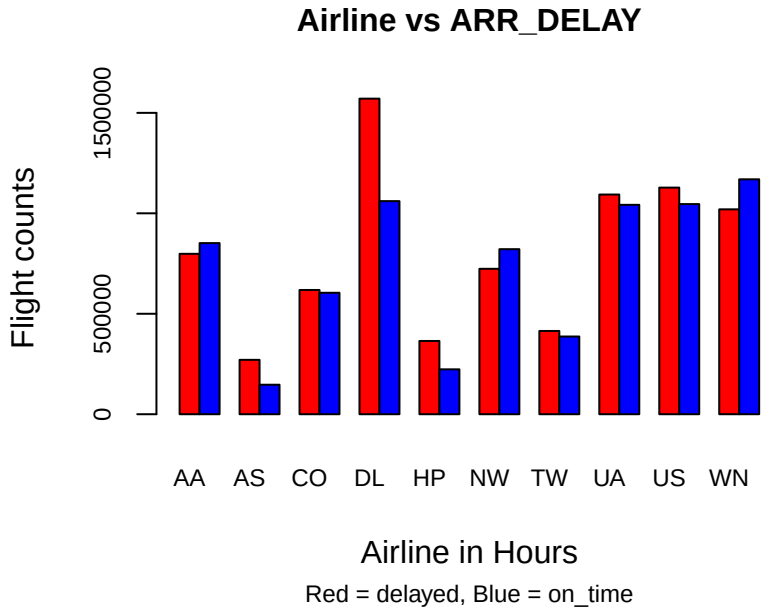
Month : This analysis was consistent with what we expected, Highest delays in December / January and moderate delays in the summer months.



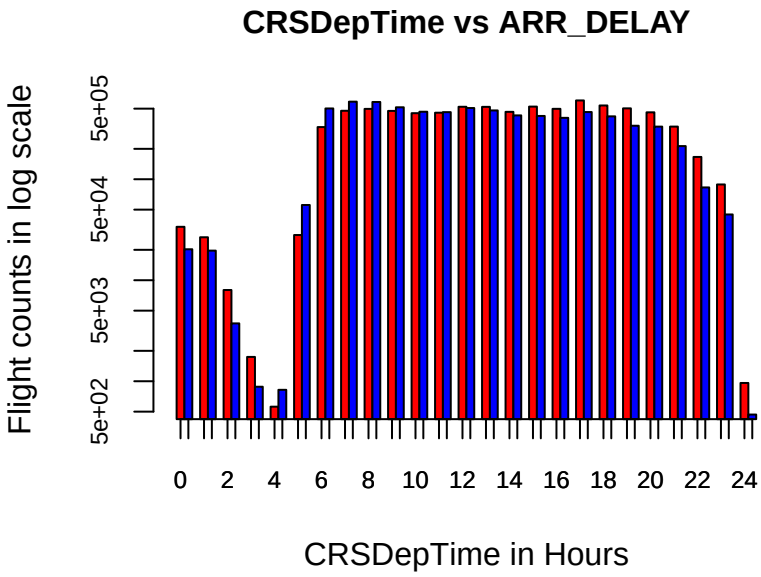
Day Of Week : This was also consistent with our assumption, higher delays in the days closer to the weekend.

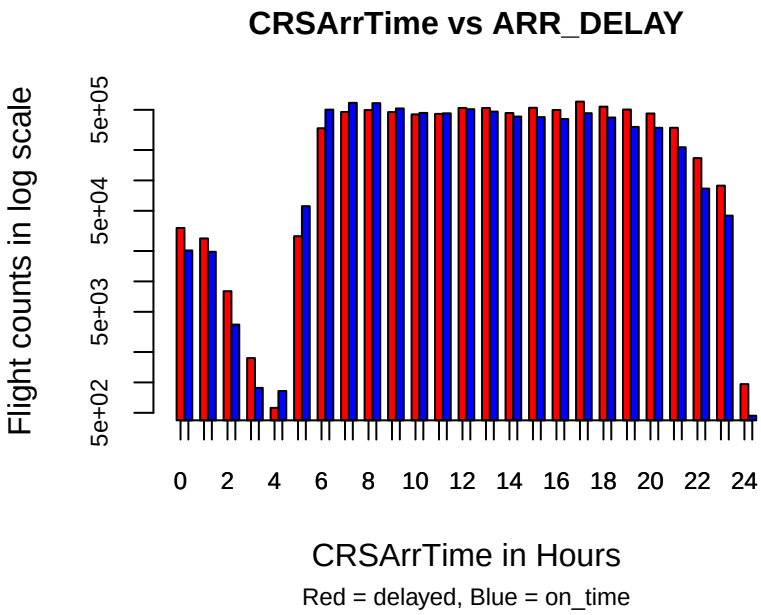


Day Of month : could not infer a great deal from the days of the month. The delays are random.

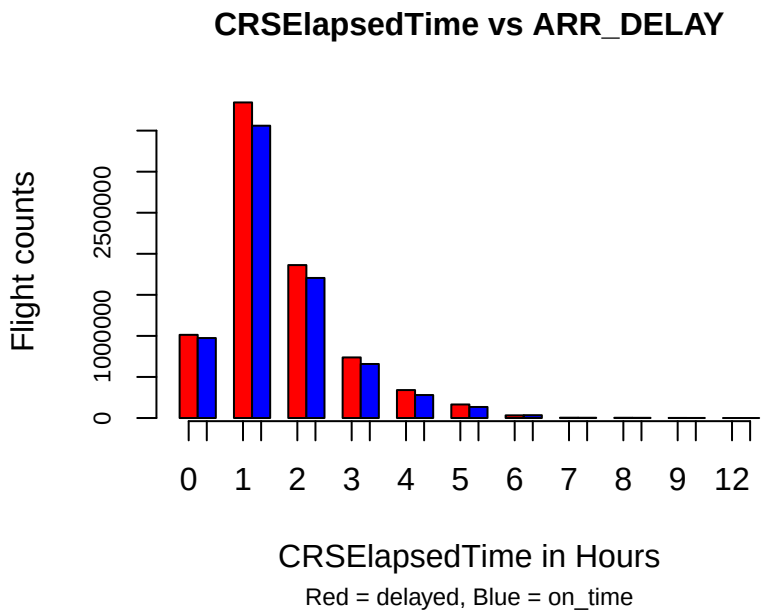


Airlines : Shows that Delta Airlines (DL) contribute most to the delay, may be this is because they have the most number of flights.





CRSDepTime/CRSArrTime : This too was consistent with what we had in mind , most delays are during peak hours (Noon to night). The least delays are observed in the mornings.



CRSElapsedTime : This too was consistent with what we thought, most number of flights are short duration and hence have most number of delays.

Origin and Dest airports were also analyzed. But could not find a way represent huge data in a single graph. Some major airports such as ATL, ORD, DFW, LAX, STL etc saw greater delays.

Features selected

Flight dataset contains missing data, invalid data etc which we sanitize and select rows. Each row contains 110 columns, not everything is relevant or suitable for using to build a model. So we select a subset of features based on the above analysis which may influence the flight ARR_DELAY.

```
Set of features which are selected as follows :
  * DAY_OF_MONTH - ranges from 1 - 31
  * DAY_OF_WEEK - ranges from 1 - 7
  * UNIQUE_CARRIER - taken carriers present in dataset and other airports taken as OTHER
  * ORIGIN - taken 50 frequent air traffic airports and other airports taken as OTHER
  * DEST - taken 50 frequent air traffic airports and other airports taken as OTHER
  * CRS_DEP_TIME - taken in Hours
  * CRS_ARR_TIME - taken in Hours
  * CRS_ELAPSED_TIME - taken in Minutes
  * ARR_DELAY - either TRUE or FALSE => Prediction variable
  * DISTANCE - taken miles

Other fields such as CANCELLED, DEP_DELAY are present in training data cannot be used as these
fields are not present for prediction future data in test.
```

Design

We experimented 2 choices for your key (month or quarter) It was noticed that using quater lead to huge dataflow to reducers and algorithms maxxed out of memory. With month as a key dataflow to each reducers

were significantly reduced and we could experiment with different algorithms.

- Two map reduce jobs are used in pipeline to get the prediction on test.
- First mappper reads training data files and emits month as key and features mentioned above along with year after checking for sanity.
- In first reducer for all years we get a particular month data. We build the model (as described below in Model sections) and save the model file in hdfs.
- Second mapper jobs kicks in after the first job completes successfully. Second mapper reads test data and picks rows which passes sanity test. Month is emitted as key along with features mentioned above along with year, flight date, flight number.
- In second reducer model file stored is read based on key from the file system and then each instance of the test data for a given month is classified and result is written as output in the format , prediction results.
- Now the predicted output (copied to local filesystem in case of cloud) and merged to form a single perdicted.csv file. The validate file is to be present on local filesystem.
- Another java program runs in local to calculate the confusion matrix by reading predicted.csv and 98validate.csv.gz(actual). Confusion matrix along with percentage of correct/error perdictions is written to a file.

Model used

Weka jar is used build and evalute models.

Various models such as NaiveBayes, RandomForest, J48, Decesion Stump, Bagging etc were built and evaluated for accuracy in weka. Some models such as J48, Bagging takes very long time to exceute.

Final model which we felt more suitable is creating RandomForest for each year and then using majority voting to predict a row in test data. Other models results along with accuracy is mentioned below.

```
Ex: There are 3 years in training dataset 1995,1996,1997
    Build a random forest model for each year with month still as a key.
    There are 3 random forest models (i.e. based on number of years)

    Now run each row of test data on each of the 3 models and get the 3 predictions.
    Use majority voting to predict the final prediction.
    If 3 trees give (TRUE, FALSE, TRUE) as predictions => TRUE

    Since we have already odd number of trees we do not need bother of breaking ties.
```

Also some advantages of using Random forest model :

```
  * Runs very fast when compared to other algorithms such as Decesion Tree, Logistic Regression
  * Does not expect linear features or even features that interact linearly
  * Can handle nominal and numeric attributes
  * Can handle high dimensional spaces as well as large number of training examples.
```

Experimentation was done on different parameters of Random Forest like choosing number of trees, number of parameters, max depth of the tree.

Results and Interpretation

Results when considering overall test data and validate data.

Random Forest

Random forest with Depth 4, Num of Trees 15, Num of Features 5

Confusion Matrix

	TRUE	FALSE
TRUE	1536130	905285
FALSE	1362312	1259631

NOTE : TRUE - means ARR_DELAY > 0(flight delayed), FALSE means ARR_DELAY <= 0(on time)

Correct Prediction :55.21%

Error Prediction :44.78%

Recall is the proportion of flight delays that were correctly identified : 62.91%

Precision is the proportion of the predicted flight delays that were correct : 52.99%

Also confusion matrix per month is computed and saved in ConfusionMatrix.txt

Performace

Time taken : 7 mins Master:1 m3.xlarge Core:4 m3.xlarge

J48 with pruning

Confusion Matrix

	TRUE	FALSE
TRUE	1417724	1023691
FALSE	1240344	1381599

Correct Prediction :55.28%

Error Prediction : 44.71%

Recall is the proportion of flight delays that were correctly identified : 58.06%

Precision is the proportion of the predicted flight delays that were correct : 53.33%

Performace

Time taken : 3+ Hrs Pseudo with 4gb Heap Size

Conclusion

- Using the above design we get an accuracy of 55% , also this takes 7 minutes to run on a cluster with 5 machines (1 master + 4 slaves) . We could improve the accuracy by analyzing the data deeper and figuring out patterns which later could be added to the model , to get better predictions.
- We could argue that although the dataset is huge, it may not contain features which would predict ARR_DELAY correctly. Additional features such as weather, average security delays at arrival or departure airport could have helped us get better predictions.
- We were limited by memory to run many algorithms which require huge in-memory computations. Also the other better way to improve accuracy is increase data size (may be 3 years is not enough) and instead of using all data use random samples and reduce data size and run experiments.