Predictions Report

By: Akanksha Mishra,Saahil Singla

This is a Latex report for predicting data assignment. We have trained our model with training data and then predicted the if flight will be delayed or not on the test data. After that we have computed accuracy of our model using confusion matrix.

Implementation and Algorithm: We have written a Map Reduce job which runs on the 36 training data files and trained our model. We send the training data to our Mapper which cleans the data and filter out the unnecessary columns. We take only those columns which are required either for attributes of our model or used to output data. We have used key as "Month" to develop 12 models so that we can train our model on month wise data. In our reducer, we use Naive Bayes algorithm and create our model on basis of attributes present in values being sent by Mapper. It trains a NaiveBayes classifier. For parameter tuning of the classifier, we enable the use of Kernel Estimator and disable Supervised Discretization properties of Wekas NaiveBayes classifier. This model predicts if the flight is delayed or not based on the attributes such as Holidays. We have assumed that flights get delayed over holidays as there is rush during that time. Similarly, other attributes contributes such as some particular airline carrier gets delayed often or there is particular time when the flights gets delayed. So our model is trained on such factors.

We run another Map-reduce job and send this model as an input to our job. Our mapper this time cleans the Test data and sends out the data to reducer in the same key-value pair format. Reducer this time decode the model and read the Model and use it to predict if flight is delayed or not on the test data.

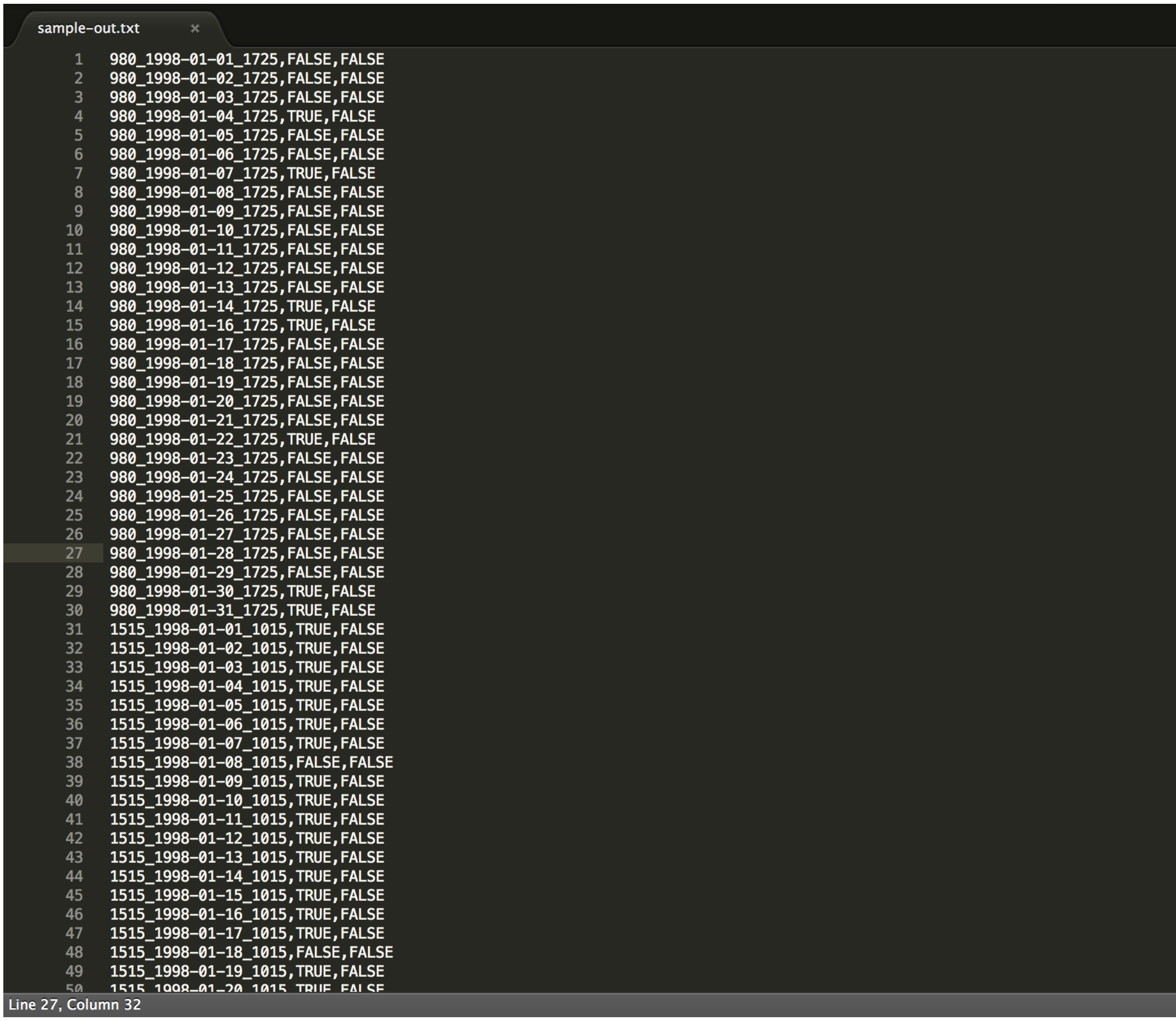
Now, we find accuracy of our program using confusion matrix. We have validation file, so we cross-check our answer with the validation file, create a confusion matrix and compute Accuracy. We check if we have predicted True that flight will be delayed and it was True in the validation file, same for False i.e it was not delayed, (TT and FF) are divided by total (TT+ FF+ TF+FT) values.

EMR takes all the time in which it takes time for starting cluster as well as all the scripts been applied.

Time of Computation			
Machine	Model-Training	Model-Test	input size
Locally	3 mins approx	4 mins approx	all
EMR	Total Time:	9 mins approx	all

Output

Our output is too large to attach, so here is a snapshot from the output:



```
980_1998-01-01_1725,FALSE,FALSE
980_1998-01-02_1725,FALSE,FALSE
980_1998-01-03_1725,FALSE,FALSE
980_1998-01-04_1725,TRUE,FALSE
980_1998-01-05_1725,FALSE,FALSE
980_1998-01-06_1725,FALSE,FALSE
980_1998-01-07_1725,TRUE,FALSE
980_1998-01-08_1725,FALSE,FALSE
980_1998-01-09_1725,FALSE,FALSE
980_1998-01-10_1725,FALSE,FALSE
980_1998-01-11_1725,FALSE,FALSE
980_1998-01-12_1725,FALSE,FALSE
980_1998-01-13_1725,FALSE,FALSE
980_1998-01-14_1725,TRUE,FALSE
980_1998-01-16_1725,TRUE,FALSE
980_1998-01-17_1725,FALSE,FALSE
980_1998-01-18_1725,FALSE,FALSE
980_1998-01-19_1725,FALSE,FALSE
980_1998-01-20_1725,FALSE,FALSE
980_1998-01-21_1725,FALSE,FALSE
980_1998-01-22_1725,TRUE,FALSE
980_1998-01-23_1725,FALSE,FALSE
980_1998-01-24_1725,FALSE,FALSE
980_1998-01-25_1725,FALSE,FALSE
980_1998-01-26_1725,FALSE,FALSE
980_1998-01-27_1725,FALSE,FALSE
980_1998-01-28_1725,FALSE,FALSE
980_1998-01-29_1725,FALSE,FALSE
980_1998-01-30_1725,TRUE,FALSE
980_1998-01-31_1725,TRUE,FALSE
1515_1998-01-01_1015,TRUE,FALSE
1515_1998-01-02_1015,TRUE,FALSE
1515_1998-01-03_1015,TRUE,FALSE
1515_1998-01-04_1015,TRUE,FALSE
1515_1998-01-05_1015,TRUE,FALSE
1515_1998-01-06_1015,TRUE,FALSE
1515_1998-01-07_1015,TRUE,FALSE
1515_1998-01-08_1015,FALSE,FALSE
1515_1998-01-09_1015,TRUE,FALSE
1515_1998-01-10_1015,TRUE,FALSE
1515_1998-01-11_1015,TRUE,FALSE
1515_1998-01-12_1015,TRUE,FALSE
1515_1998-01-13_1015,TRUE,FALSE
1515_1998-01-14_1015,TRUE,FALSE
1515_1998-01-15_1015,TRUE,FALSE
1515_1998-01-16_1015,TRUE,FALSE
1515_1998-01-17_1015,TRUE,FALSE
1515_1998-01-18_1015,FALSE,FALSE
1515_1998-01-19_1015,TRUE,FALSE
1515_1998-01-20_1015,TRUE,FALSE
```

Conclusion

Confusion Matrix: We have the following values and we use this formula to calculate Accuracy.

TT	:	721131
FF	:	1256161
TF	:	1259956
FT	:	465546

Accuracy = ((TT + FT)/(TT+FF+FT+TF))*100
= (1977292)/(3702794)*100

Accuracy: 53.40 percentage

Output from our phython script which finds out the accuracy.

```
”Percentage Accurate: 53.40”
”True True : 721131 ”
”True False: 1259956”
”False True: 465546 ”
’False False: 1256161’
```