Assignment 7 - Part 1: Prediction

Team: Sudeep Vaka, Kaushik Veluru, Aditya Ramesha

Date: 03/19/2016

File Structure:

Prediction

- Forest
- FlightWritable
- FlightModelMapper
- FlightClassifyReducer
- Evaluation

Data Cleaning:

The Flight data with 36 historical files is first proccessed in R, to get only the required features. Files from the bucket a6history/ are merged to one file with headers. An extra column with letter 'M' is added to check this file as a Model file. Same is applied for the test file from a6test and a column with letter 'T' is added. The features we thought would be important are,

- Month (Winter and holiday months tend have more delays)
- Depart Hour (Flights departing later in the day tend to be late)
- Day of Month (Will cover for holidays - like last week of Nov, Dec)
- Day of Week (Weekends are more prone to delays)
- Distance (More distance - more probability of delays)

R scripts to extract the above features and the class label are present in process.r

```
csvfiles <- dir("/Users/Sudeep/Downloads/prediction/a6history/")
df <- ldply(csvfiles, read.csv, header=T)
history <- df[,c(1, 3, 5, 6, 7, 11, 12, 21, 30, 56, 48, 43)]
history$DATA <- "M"
write.csv(history, file = "a6history.csv")
```

Now that we have the filtered data, we use the files a6history.csv, a6test.csv to build and evaluate the classifier.

Mapper (Data Split):

Both the histroy and test file is given as input to the Mapper. Each flight record is sent from Mapper thrice with three different keys

- Carrier
- Destination

- Origin

```
mapKey.set(flight.getCarrier());
context.write(mapKey, flight);
mapKey.set(flight.getDestination());
context.write(mapKey, flight);
mapKey.set(flight.getOrigin());
context.write(mapKey, flight);
```

Reducer (Building Classifier):

We decided to use the Naive Bayes Classifier provided by weka libarary to build our prediction model. In our observation we found that the Naive Bayes classifier worked faster than the RandomForest and yielded similar results in accuracy.

The attributes to model the classifier are defined in the setup method

```
attributes.addElement(new Attribute("Month"));
attributes.addElement(new Attribute("DepartureHour"));
attributes.addElement(new Attribute("WeekDay"));
attributes.addElement(new Attribute("Distance"));
attributes.addElement(new Attribute("MonthDay"));
FastVector possibleClasses = new FastVector(2);
possibleClasses.addElement("1");
possibleClasses.addElement("0");
attributes.addElement(new Attribute("Delayed",possibleClasses));
```

From the accumulated values, each Flight Writable object is checked if it is a Model record or test record and written to an Instance(weka.core.Instance). Model Instances are used to update the Naive Bayes Classifier and test Instances are classified against the naive bayes classifier object. Since we are sending each record thrice, the key [FL_NUM]*[FL_DATE]*[CRS_DEP_TIME] will have three predicted values, each one from different model.

```
for (FlightWritable fl : values) {
    Instance inst=new Instance(trainInstns.numAttributes());
    inst.setDataset(trainInstns);
    inst.setValue(0, fl.getMonth());
    inst.setValue(1, getHH(fl.getDepartHour()));
    inst.setValue(2, fl.getWeekday());
    inst.setValue(3, fl.getDistanceGroup());
    inst.setValue(4, fl.getMonthday());
    if(fl.getIsModel())
    {
        inst.setValue(5, fl.getDelayed()+"");
        build(inst);
    }
    else
    {
        predict(inst,fl);
    }
```

```
}
```

In the cleanup method, The class label is decided from the three predicted labels based on majority and written to context in the following format.

```
[FL_NUM]_[FL_DATE]_[CRS_DEP_TIME]   logical
1457_1998-08-19_0640    FALSE
1631_1998-01-23_1115    TRUE
407_1998-12-10_1715 TRUE
403_1998-07-21_0800 FALSE
350_1998-05-17_1155 FALSE
```

'TRUE' indicates that the respective flightg is delayed and vice versa

Evaluation :

The output from the reducer and the a6validate files are read from Evaluation.java and the confusion matrix is calculated from the True positives and True negtatives.

```
//Predicted delayed
if(prediction.get(line[0]).equals("TRUE"))
{
    //validation delayed
    if(line[1].equals("TRUE"))
        tp++;
    else
        fp++;
    //validation not delayed
}
//Predicted not delayed
else
{
    //validation delayed
    if(line[1].equals("TRUE"))
        fn++;
    else
        tn++;
    //validation not delayed
}
```

The Confusion Matrix calculated from the evaluation is as below

Predicted\Validation	**Delayed**	**Not Delayed**	**Total**
Delayed	1748174.0(tp)	1663723.0(fp)	
Not Delayed	718907.0(fn)	986047.0(tn)	
Total			5116851.0

```
Precision: 0.5123
Recall:0.7086
```

```
Accuracy:0.5344
```

The above accuracy is calculated by (tp+tn)/(Total) (The percenatage of correctly classified flights)

The sum of the percentage of on-time flights misclassified as delayed and the percentage of delayed flights misclassified as on-time: 46.56%

Execution Time:

Time taken to process and trim the data in R (For history and test): ~ 20 mins

Time taken to to build model and predict on test input: ~ 10 mins

Time taken to Evaluate the model: ~ 30 secs