End to End Machine Learning Project
Predict if a Flight Would Be On-Time for American Airlines

Salma Alqahtani
ID Numer: 441813779
441813779@kku.edu.sa

Samar Alqahtani
ID Numer: 441814067
441814067@kku.edu.sa

April 16, 2020

1 Project Idea

In this project, we use machine learning workflow to process and transform dataset of American airlines to create a prediction model. This model must predictive whether a flight would arrive $15+$ minutes after the scheduled arrival time with a good accuracy. Towards this end, we conduct numerical experiments on a dataset that contains data for the number of on-time, delayed, canceled, and diverted flights to/from U.S. airports that occurred in January 2015.

Keywords. Binary Classification; Predictive Model; Logistic Regression; Random Forest

2 Selection of Dataset

Delay is one of the most remembered performance indicator of any transportation system, since flight delays have negative consequences on airlines, airports and passengers. On-time operation of the airports and airlines schedules are the target of all airports and airlines stockholders in order to fulfill with passengers and customer requirements as well as getting more new customers. The U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics tracks the on-time performance of domestic flights operated by large air carriers. The data is automatically downloaded as a CSV file by nagivation to the following link: `https://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236&DB_Short_Name=On-Time` from DOT's website.

The data contains $469,968$ flights with features categorized as follows: (i) Information about flight (day of the month, day of the week, tail number, flight number) (ii) Information about origin and destination (origin airport, destination airport) (iii) Information about the departure (departure time, departure delay) (iv) Information about the arrival (arrival time, arrival delay) (v) Information about diversion, cancellation, distance.

The dataset contains a big number of row samples. Therefore, it will take a long time to process. To speed up the processing of our numerical experiments, let's restrict data to only flight between certain important large of airports ALT, LAX, ORD, DFW, JFK, SFO, CLT, LAS, PHX. We then get a $32,716$ observations of 22 features. Figure 1 presents the names of all the features followed by its types. As can be seen we have both categorical and continuous features. For our study, the *target* feature is $y =$ `ARR_DEL15`. In the data Y has a character type belonging to {"1.00,"0.00"}. In the section of preparing data, we will process to transform it in discrete type, since we will use the R machine learning package `caret`, that only considers discrete or real variables rather than character type variables. Since the target variable Y takes two classes $C1 =$"1.00," and $C2 =$"0.00," corresponding to the existing and a non-existing of delay greater

than 15 minutes, respectively. Hence the machine learning problem considered in this project is *binary classification,* that is for all $i = 1, 2, \ldots, 32,716$.

$$y_i = \begin{cases} 1 & \text{if delay} \geq 15\text{minutes} \\ 0 & \text{otherwise.} \end{cases}$$

```
'data.frame':   32716 obs. of  22 variables:
 $ DAY_OF_MONTH        : int  1 1 1 1 1 1 1 1 1 1 ...
 $ DAY_OF_WEEK         : int  4 4 4 4 4 4 4 4 4 4 ...
 $ OP_UNIQUE_CARRIER   : chr  "AA" "AA" "AA" "AA" ...
 $ OP_CARRIER_AIRLINE_ID: int  19805 19805 19805 19805 19805 19805 19805 19805 19805 19805 ...
 $ OP_CARRIER          : chr  "AA" "AA" "AA" "AA" ...
 $ TAIL_NUM            : chr  "N003AA" "N437AA" "N437AA" "N3DTAA" ...
 $ OP_CARRIER_FL_NUM   : int  1298 1389 1389 1311 1335 1336 1337 1337 1410 1393 ...
 $ ORIGIN_AIRPORT_ID   : int  11298 11298 12889 11057 11298 14107 11298 12889 12892 11298 ...
 $ ORIGIN_AIRPORT_SEQ_ID: int  1129803 1129803 1288903 1105703 1129803 1410702 1129803 1288903 1289203 1129803 ...
 $ ORIGIN              : chr  "DFW" "DFW" "LAS" "CLT" ...
 $ DEST_AIRPORT_ID     : int  10397 12889 11298 13930 12889 11298 12889 11298 12889 14771 ...
 $ DEST_AIRPORT_SEQ_ID : int  1039705 1288903 1129803 1393003 1288903 1129803 1288903 1129803 1288903 1477101 ...
 $ DEST                : chr  "ATL" "LAS" "DFW" "ORD" ...
 $ DEP_TIME            : int  2000 1717 1854 1256 2258 603 1741 1911 2215 1751 ...
 $ DEP_DEL15           : chr  "1.00" "0.00" "0.00" "0.00" ...
 $ DEP_TIME_BLK        : chr  "1800-1859" "1700-1759" "1900-1959" "1200-1259" ...
 $ ARR_TIME            : int  2312 1804 2333 1357 12 919 1827 2343 2325 1925 ...
 $ ARR_DEL15           : chr  "1.00" "0.00" "0.00" "0.00" ...
 $ CANCELLED           : chr  "0.00" "0.00" "0.00" "0.00" ...
 $ DIVERTED            : chr  "0.00" "0.00" "0.00" "0.00" ...
 $ DISTANCE            : chr  "731.00" "1055.00" "1055.00" "599.00" ...
 $ X                   : logi  NA NA NA NA NA NA ...
```

Figure 1: Overview and description of the features in the dataset.

3 Descriptive Statistics to perform EDA

Before introducing the prediction models, we will explain the dataset has been used. This aims to give readers a better idea to understand the used models and the principle we have used according to the dataset characteristics. In Figure 2, we report a statistics summary of the dataset, where for each continuous features we report its minimum, maximum, and mean values, and the 1^{st}, median, and 3^{th} quantiles. We further notice that we have categorical features in our dataset, in particular our target features (YES/NO delayed). Therefore, we will investigate the changes of the categorical features to well apply `caret` R-package of machine learning.

```
 DAY_OF_MONTH    DAY_OF_WEEK    OP_UNIQUE_CARRIER  OP_CARRIER_AIRLINE_ID  OP_CARRIER          TAIL_NUM        OP_CARRIER_FL_NUM
Min.   : 1.00   Min.   :1.000   Length:32716       Min.   :19393         Length:32716       Length:32716       Min.   :   1
1st Qu.: 8.00   1st Qu.:2.000   Class :character   1st Qu.:19790         Class :character   Class :character   1st Qu.: 472
Median :16.00   Median :4.000   Mode  :character   Median :19805         Mode  :character   Mode  :character   Median :1115
Mean   :15.92   Mean   :4.053                      Mean   :20018                                               Mean   :1438
3rd Qu.:24.00   3rd Qu.:6.000                      3rd Qu.:20355                                               3rd Qu.:1993
Max.   :31.00   Max.   :7.000                      Max.   :21171                                               Max.   :6531

ORIGIN_AIRPORT_ID ORIGIN_AIRPORT_SEQ_ID    ORIGIN          DEST_AIRPORT_ID DEST_AIRPORT_SEQ_ID     DEST            DEP_TIME
Min.   :10397     Min.   :1039705      Length:32716       Min.   :10397   Min.   :1039705     Length:32716       Min.   :   1
1st Qu.:11298     1st Qu.:1129803      Class :character   1st Qu.:11298   1st Qu.:1129803     Class :character   1st Qu.: 924
Median :12892     Median :1289203      Mode  :character   Median :12892   Median :1289203     Mode  :character   Median :1323
Mean   :12783     Mean   :1278352                         Mean   :12779   Mean   :1277910                        Mean   :1333
3rd Qu.:13930     3rd Qu.:1393003                         3rd Qu.:13930   3rd Qu.:1393003                        3rd Qu.:1733
Max.   :14771     Max.   :1477101                         Max.   :14771   Max.   :1477101                        Max.   :2400
                                                                                                                 NA's   :550
 DEP_DEL15       DEP_TIME_BLK          ARR_TIME     ARR_DEL15          CANCELLED          DIVERTED           DISTANCE
Length:32716       Length:32716       Min.   :   1   Length:32716       Length:32716       Length:32716       Length:32716
Class :character   Class :character   1st Qu.:1129   Class :character   Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Median :1534   Mode  :character   Mode  :character   Mode  :character   Mode  :character
                                      Mean   :1504
                                      3rd Qu.:1937
                                      Max.   :2400
                                      NA's   :565

   X
Mode:logical
NA's:32716
```

Figure 2: Statistics summary of the dataset

4 Visualization to perform EDA

In this section, we report some plots about the features in our dataset in order to get a an idea about the features. In Figure 3 we report the proportion of on time and delayed flights. Almost 80% of the flight are on time and 20% are delayed. Figure 4 shows the fraction of flights grouped by airlines. The airlines are shown using IATA airline codes. For example, label AA is for Alaska

Airlines. Figure 2 shows the arrival-delay distribution during each day of every month in 2015. For example, label 1 denotes the delay distribution for the first day of every month in 2015. Figure 5 shows flight arrivals status depending on departure delay. Figure 6 shows number of flights delayed grouped by departure time block.

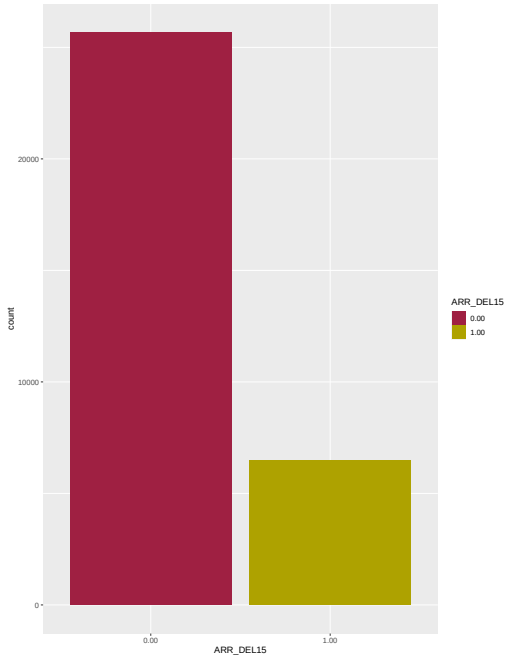

Figure 3: Proportion of On Time Delayed Flights

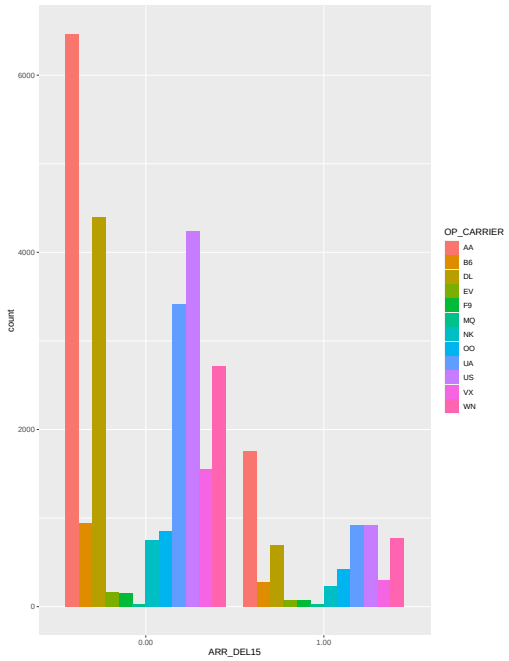

Figure 4: Proportion of On Time Delayed flights grouped by airline

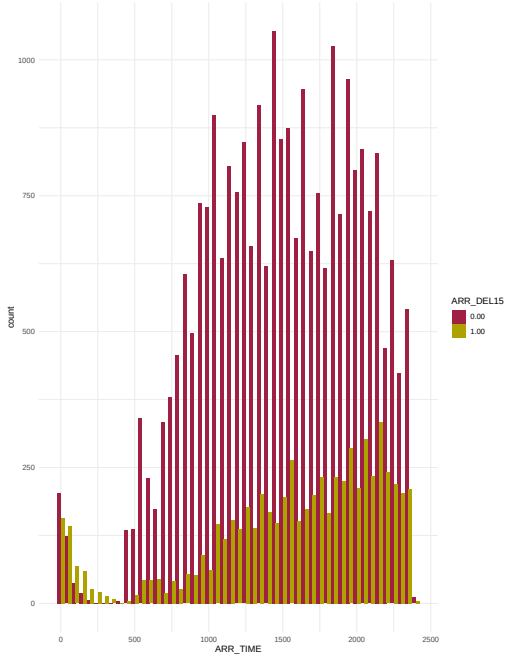

Figure 5: Plot for flight arrivals status depending on departure delay

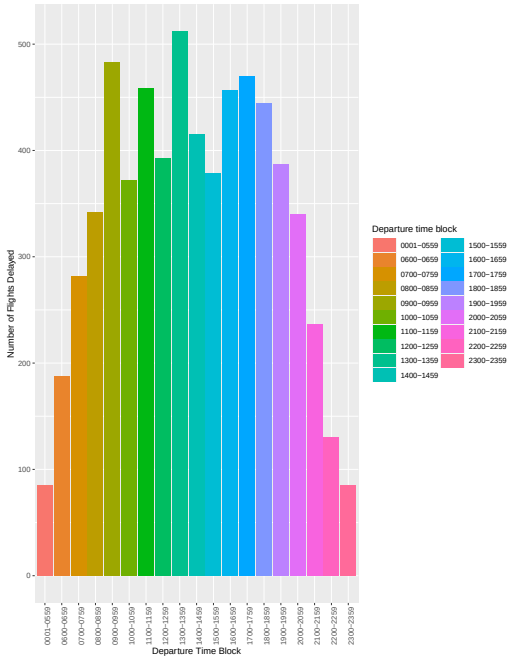

Figure 6: Plot for number of flights delayed grouped by departure time block

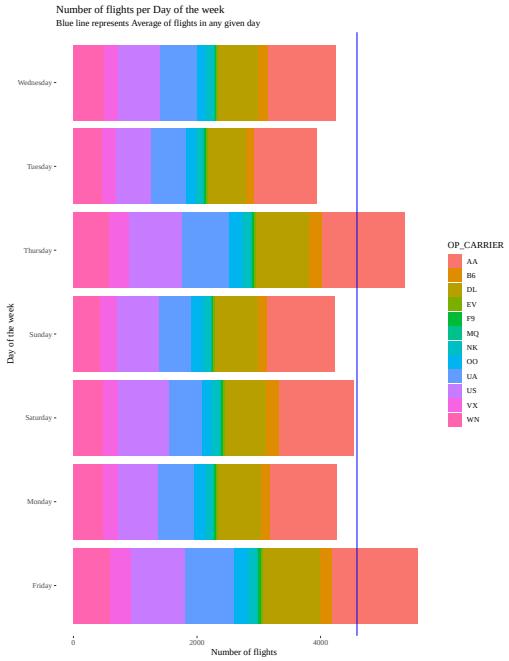

Figure 7: Plot for number of flights per day of the week

5 Data Preparation

Data Cleaning. In Figure 1, we notice the last column X with values of NA. The X column was created as part of the import from CSV. And NA is the mean the data was Not Available. So it looks like that column can be dropped. In general we want to eliminate any columns that we do not need. In particular, we want to eliminate columns that are duplicates or provide the same information. We can do this by: (i) Visual inspect if we have columns that are really the same. But visual inspection is error prone and does not deal with a second issue of correlation. (ii) Often there are correlated columns such as an ID and the text value for the ID. And these highly correlated columns usually do not add information about the how the data causes changes in the results, but do cause the effect of a field to be overly amplified because some algorithm naively treat ever columns as being independent and just as important.

In looking at the data, we see the possible correlations between ORIGIN_AIRPORT_SEQ_ID and DEST_AIRPORT_ID and between DEST_AIRPORT_SEQ_ID and DEST_AIRPORT_ID.

```
> cor(dataframe[c("ORIGIN_AIRPORT_SEQ_ID", "ORIGIN_AIRPORT_ID")])
                    ORIGIN_AIRPORT_SEQ_ID ORIGIN_AIRPORT_ID
ORIGIN_AIRPORT_SEQ_ID                     1                 1
ORIGIN_AIRPORT_ID                         1                 1

> cor(dataframe[c("DEST_AIRPORT_ID", "DEST_AIRPORT_ID")])
                  DEST_AIRPORT_ID DEST_AIRPORT_ID.1
DEST_AIRPORT_ID                 1                 1
DEST_AIRPORT_ID.1               1                 1
```

Figure 8: Correlations between some features in the dataset

Additionally, OP_UNIQUE_CARRIER and OP_CARRIER also look related, actually they look like identical. We can see if the are identical by filtering the rows to those we they are different. R makes this easy, no loops to write. Finally we drop all rows where the target variable ARR_DEL15

is empty or NA. We use the following code in our R script.

Data Preprocessing. Sometimes algorithm perform better when you change the fields into factors which are like enumeration values in other languages. This allows the algorithm to use count of when a value is a discrete value. Let also change some other columns factors. The following figure show this preprocessing to the dataset

```
'data.frame':   32124 obs. of  15 variables:
 $ DAY_OF_MONTH     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ DAY_OF_WEEK      : Factor w/ 7 levels "Friday","Monday",..: 5 5 5 5 5 5 5 5 5 5 ...
 $ OP_CARRIER       : Factor w/ 12 levels "AA","B6","DL",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ TAIL_NUM         : Factor w/ 3122 levels "N001AA","N002AA",..: 3 1168 1168 922 1641 1062 1730 1730 2701 1044
 ...
 $ OP_CARRIER_FL_NUM: int  1298 1389 1389 1311 1335 1336 1337 1337 1410 1393 ...
 $ ORIGIN           : Factor w/ 9 levels "ATL","CLT","DFW",..: 3 3 5 2 3 8 3 5 6 3 ...
 $ DEST             : Factor w/ 9 levels "ATL","CLT","DFW",..: 1 5 3 7 5 3 5 3 5 9 ...
 $ DEP_TIME         : int  2000 1717 1854 1256 2258 603 1741 1911 2215 1751 ...
 $ DEP_DEL15        : Factor w/ 2 levels "2","3": 2 1 1 1 2 1 2 2 1 1 ...
 $ DEP_TIME_BLK     : Factor w/ 19 levels "0001-0559","0600-0659",..: 14 13 15 8 18 2 12 13 18 13 ...
 $ ARR_TIME         : int  2312 1804 2333 1357 12 919 1827 2343 2325 1925 ...
 $ ARR_DEL15        : Factor w/ 2 levels "0.00","1.00": 2 1 1 1 2 1 2 2 1 1 ...
 $ CANCELLED        : int  1 1 1 1 1 1 1 1 1 1 ...
 $ DIVERTED         : int  1 1 1 1 1 1 1 1 1 1 ...
 $ DISTANCE         : int  31 1 1 28 1 35 1 1 20 5 ...
```

Figure 9: Preprocessing of some features in the dataset

Let see how many arrival delayed vs non delayed flights. We use `tapply` R-command to see how many time `ARR_DEL15` is TRUE, and how many times it is FALSE. We should check how many departure delayed vs non delayed flights. The fact that we have a reasonable number of delays $(6460/(25664 + 6460)) = 0.201$ (20%) is important. So almost 20% of the dataset are delayed flights.

```
> round(prop.table(table(dataframe$ARR_DEL15))*100,2)

      0.00  1.00
0.00 79.89 20.11
```

Figure 10: Percentage of Ontime/Delayed flights

Splitting Data. The whole dataset is shuffled in a consistent way and split into Training and Validation Sets with 70% of data constituting the Training Set and 30% of data constituting the Validation Set. Since our laptop does not have a good capacity of RAM memory, and after our data cleaning, we build our machine leaning algorithms on the following features to explain the target variable of Delayed +15 minutes.

```
# Set the columns we are going to use to train algorithm
featureCols <- c("ARR_DEL15", "DAY_OF_WEEK", "OP_CARRIER", "DEST","ORIGIN","DEP_TIME_BLK")
```

Figure 11: Considered features to explain Ontime/Delayed flights

6 Model Building: Machine Learning Algorithms

The goal here is to identify flights that are likely to be delayed. In the machine learning literature this is called a binary classification using supervised learning. We are bucketing flights into delayed or ontime(hence binary classification). (Note: Prediction and classification are two main big goals of data mining and data science.

Logistic regression provides us with a probability of belonging to one or the two cases (delayed or ontime). Since probability ranges from 0 to 1, we will use the 0.5 cutoff to determine which

bucket to put our probability estimates in. If the probability estimate from the logistic regression is equal to or greater than 0.5 then we assign it to be ontime else it's delayed.

6.1 Logistic Regression

Logistic regression is a simple classification algorithm that uses the following hypothesis:

$$f_\beta(x) = g(\beta^\top x) = \frac{1}{1 + e^{-\beta^\top x}} \text{ where } \beta^\top x = \beta_0 + \sum_{j=1}^{p} \beta_j x_j,$$

and where p is the number of features. In our setting in our setting we have 22 features, then $p = 22$. We can find parameter β that best describes our training data using the maximum likelihood estimation and gradient descent algorithm. the like-lihood of the logistic model reads as

$$\ell(\beta) = \sum_{i=1}^{n} \big(y_i \log(f_\beta(x_i)) + (1 - y_i) \log(1 - f_\beta(x_i))\big),$$

where i is the number of samples. In our case $n = 32716$. Recall that the dimension of the data set is 32716×22.

In the following figure we report the outputs of the Logistic Regression model. Note that the accuracy giving by this model is almost 80% which seems good as a level of prediction. We further emphasize that the accuracy is calculated on the testing set with 10-fold cross-validation.

```
Generalized Linear Model

32124 samples
    5 predictor
    2 classes: '0.00', '1.00'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 28912, 28912, 28911, 28911, 28911, 28912, ...
Resampling results:

  Accuracy   Kappa
  0.7992154  0.01807733
```

Figure 12: Output of the Logistic Regression model with an $accuracy = 79,92\%$

6.2 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set

```
Call:
 randomForest(x = traindata[-1], y = traindata$ARR_DEL15, importance = TRUE,      proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 21.99%
Confusion matrix:
      0.00 1.00 class.error
0.00 17062  903   0.0502644
1.00  4043  479   0.8940734
> |
```

Figure 13: Output of the Random Forest model with an $accuracy = 89,40\%$

7 Model Evaluation

Confusion Matrix. A confusion matrix is a technique for summarizing the performance of a classification algorithm. Classification accuracy alone can be misleading if you have an unequal number of observations in each class or if you have more than two classes in your dataset. Calculating a confusion matrix can give you a better idea of what your classification model is getting right and what types of errors it is making.

```
> logRegConfMat <- confusionMatrix(logRegPrediction, testdata[,"ARR_DEL15"]
> logRegConfMat
Confusion Matrix and Statistics

          Reference
Prediction 0.00 1.00
      0.00 7678 1909
      1.00   21   29

               Accuracy : 0.7997
                 95% CI : (0.7916, 0.8077)
    No Information Rate : 0.7989
    P-Value [Acc > NIR] : 0.4254

                  Kappa : 0.0193

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.99727
            Specificity : 0.01496
         Pos Pred Value : 0.80088
         Neg Pred Value : 0.58000
             Prevalence : 0.79890
         Detection Rate : 0.79672
   Detection Prevalence : 0.99481
      Balanced Accuracy : 0.50612

       'Positive' Class : 0.00
```

Figure 14: Testing Model's Accuracy for Logistic Regression model

```
> rfConfMat <- confusionMatrix(rfValidation, testdata[,"ARR_DEL15"])
> rfConfMat
Confusion Matrix and Statistics

          Reference
Prediction 0.00 1.00
      0.00 7317 1746
      1.00  382  192

               Accuracy : 0.7792
                 95% CI : (0.7708, 0.7874)
    No Information Rate : 0.7989
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0671

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.95038
            Specificity : 0.09907
         Pos Pred Value : 0.80735
         Neg Pred Value : 0.33449
             Prevalence : 0.79890
         Detection Rate : 0.75926
   Detection Prevalence : 0.94044
      Balanced Accuracy : 0.52473

       'Positive' Class : 0.00
```

Figure 15: Testing Model's Accuracy for Random Forest model

8 Visualization of Finals Results

Here we visuliaze the Confusion Matrix of Logitic Regression and Random Forest models for the prediction task.

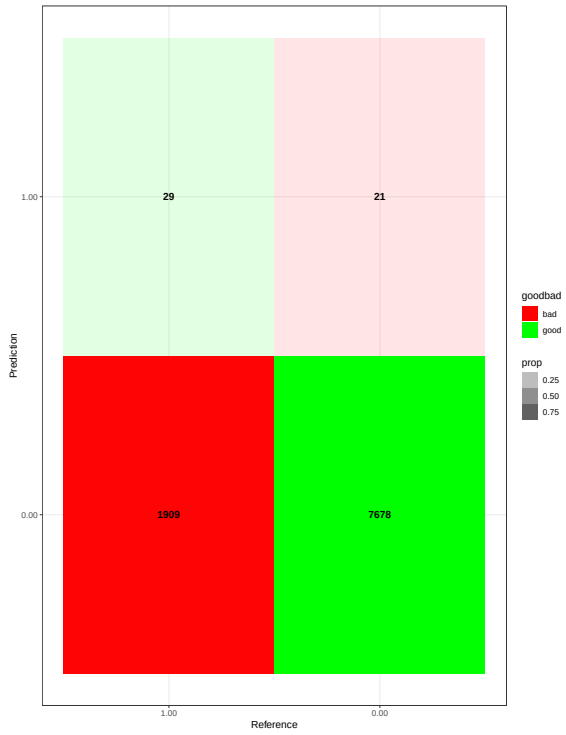

Figure 16: Visualization of the confusion matrix for the Logistic regression model

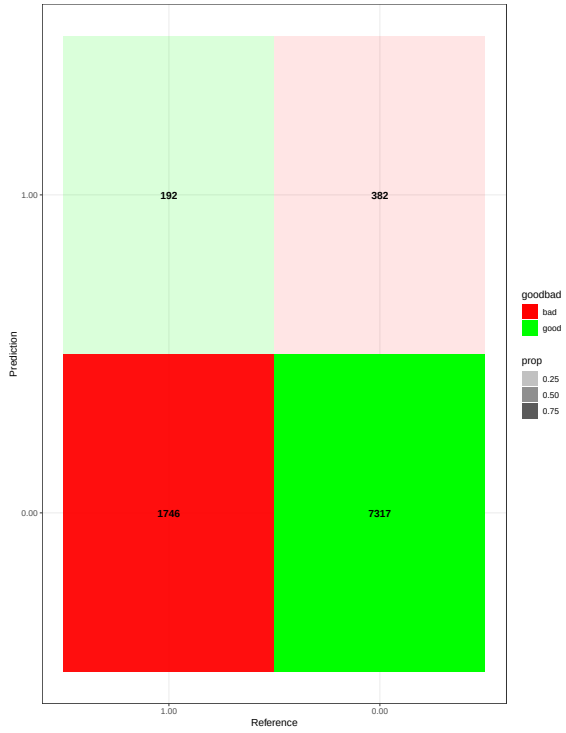

Figure 17: Visualization of the confusion matrix for the Logistic regression model

9 Repository on Github

The project was published on a repository in Github websiten, and it can be downloaded at the following link: http://github/salma-samar/PredictUSflightsJan2015.

10 Discussion

In this project we studied a 2 machine learning workflow models for binary classification: Logistic Regression and RandomForest in order to help the user to analyse and comprehend the best or the worst time to fly within United States of America. As an extension of this work, it is worth to investigate other binary classification algorithms such Support Vector Machine (SVM), Tree Decision, and compare the performance of prediction between these algorithms. In a more advanced level, we would like to investigate deep neural networks (DNN) within this dataset.

References

1 Wikipedia: https://en.wikipedia.org/

2 Machine Learning Mastery: https://machinelearningmastery.com

3 RPubs: https://rpubs.com