HW1

Shariq Mian

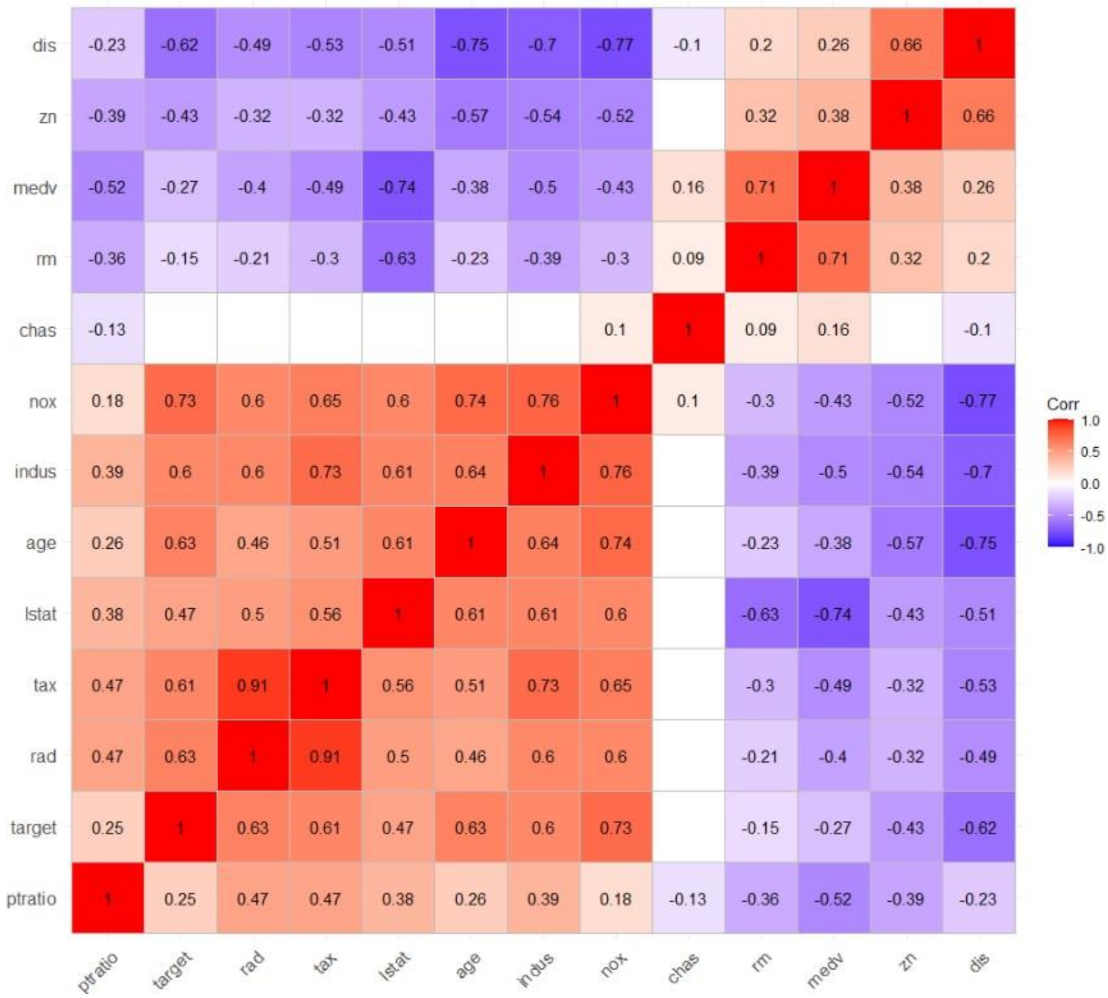
10/9/2022

In this assignment we will explore two data set. One is the Chicago Crime Data and the other one is 2018 Flight Delays Data. The Chicago Crime Data consist of 466 rows 13 columns while 2018 flight consist 7.2 million rows and 28 columns of data. Both the data were from Kaggle and al culoms have labels except one column in the Flight Delays data, which is called Unnamed Column 27.

Chicago Crime Data

For the Crime data, While the data set contains many variables, the target variable is our actual target variable or dependent variable. Here its either a 0 meaning that the crime rate is above the median crime rate or a 1 which is below the median crime rate. Here we can apply two different type of classification algorithm. First a Logistic Regression, because the data is bounded by a probability between 0 and 1 we can apply the binary logistic regression, where the .5 mark is divider for the classification for high crime risk and low crime risk . The other regression I think we can apply here is KNN. Because since there are binary outcomes, we can define the target variable the same way less than .5 and greater than .5 as 0 and 1 outcomes and then apply the KNN algorithm where based on similar attributes its able to predict whether the area is median crime rate or not. Also, since it's a classification problem, the confusion matrix is a very helpful table, which can help us determine some of the classification metrics such as accuracy, precision, error rate, sensitivity and F1 score.

Below we have a heat map of the correlation matrix to see whether we have to remove any variable due to multi collinearity. This is important if we want to choose our model to remove any overfitting. Some of the high correlated variables are in dus vs nox and age vs nox. Since there are highly correlated independent variables, we will remove 2 out of 3 of these variables for our model



In our model we remove all the variables with P value greater than .05. Our model turns out to be target ~ nox + age + rad + tax + ptratio + lstat. Here our AIC score is 228.35 with only lstat having a P value less than .05 and insignificant. With nox has the biggest effect on the model with the lowest P value and coefficient of 34.05.

The reason to choose Logistic Regression for this Data Set instead of the KNN here was that because the data set is small thus KNN will be very sensitive to the size of the data or any irrelevant features. Also because the data set is small, with Logistic Regression can derive the confident level while in KNN we can only output the results, thus making it the right choice for this data set where it would be the right algorithm to use for business decision.

2018 Flight Delays

For the second data set, the data set is much larger 7.2 million rows, and while I was able to run the data set in R, I was unbale to run it as RMD file because the computation time was too much for my R to handle the and would crash. The structure of dataset indicates that there are 2 variables, that consists of a variety of categorical and continuous variables.

```
'data.frame':   7213446 obs. of  28 variables:
 $ FL_DATE            : chr  "2018-01-01" "2018-01-01" "2018-01-01" "2018-01-01" ...
 $ OP_CARRIER         : chr  "UA" "UA" "UA" "UA" ...
 $ ORIGIN             : chr  "EWR" "LAS" "SNA" "RSW" ...
 $ DEST               : chr  "DEN" "SFO" "DEN" "ORD" ...
 $ CRS_DEP_TIME       : int  1517 1115 1335 1546 630 2241 750 1324 2224 1601 ...
 $ CRS_ARR_TIME       : int  1745 1254 1649 1756 922 14 916 1619 638 1813 ...
 $ CRS_ELAPSED_TIME   : num  268 99 134 190 112 93 206 115 314 252 ...
 $ ACTUAL_ELAPSED_TIME: num  250 83 126 182 106 79 193 102 299 237 ...
 $ DEP_TIME           : num  1512 1107 1330 1552 650 ...
 $ ARR_TIME           : num  1722 1230 1636 1754 936 ...
 $ DEP_DELAY          : num  -5 -8 -5 6 20 3 -3 -6 13 -2 ...
 $ ARR_DELAY          : num  -23 -24 -13 -2 14 -11 -16 -19 -2 -17 ...
 $ CARRIER_DELAY      : num  NA NA NA NA NA NA NA NA NA NA ...
 $ WEATHER_DELAY      : num  NA NA NA NA NA NA NA NA NA NA ...
 $ NAS_DELAY          : num  NA NA NA NA NA NA NA NA NA NA ...
 $ SECURITY_DELAY     : num  NA NA NA NA NA NA NA NA NA NA ...
 $ LATE_AIRCRAFT_DELAY: num  NA NA NA NA NA NA NA NA NA NA ...
 $ CANCELLED          : num  0 0 0 0 0 0 0 0 0 0 ...
 $ DIVERTED           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ TAXI_OUT           : num  15 11 15 19 13 15 14 11 10 12 ...
 $ TAXI_IN            : num  10 7 5 6 10 2 6 6 9 8 ...
 $ WHEELS_OFF         : num  1527 1118 1345 1611 703 ...
 $ WHEELS_ON          : num  1712 1223 1631 1748 926 ...
 $ CANCELLATION_CODE  : chr  "" "" "" "" ...
 $ AIR_TIME           : num  225 65 106 157 83 62 173 85 280 217 ...
 $ DISTANCE           : num  1605 414 846 1120 723 ...
 $ OP_CARRIER_FL_NUM  : int  2429 2427 2426 2425 2424 2422 2421 2420 2419 2418 ...
 $ Unnamed..27        : logi  NA NA NA NA NA NA ...
```

Based on the data set the to best choice of algorithm here would have been KNN and Naïve Bias. Once again, KNN seemed like the right choice because it's a classification problem, based on certain attributes we are predicting whether flight will be delayed or not. Here the data set is large enough where we don't have to worry about the effect of specific attribute like in our other dataset which was too small. The other choice was Naïve Bias, which very similar to KNN is a classification algorithm but historically yields better accuracy percentage compare to KNN and Decision trees.

After looking at the pros and cons for both the algorithm, Naïve Bias was the right choice of algorithm for this data set because, we have ~ 7.2 million rows and Naïve Bias tends to perform faster on big volume of data compare to KNN. Because I am working on my local machine this was important, especially because Naïve Bias also gave me challenges with the computation time on my computer knowing that its faster than KNN. Also one important thing to note it that Naïve Bias is not effected by the Curse of Dimensionality and large features. These were enough for us to to choose Naïve Bias over KNN for this data set.

After conducting the missing data analysis, we found that CARRIER_DELAY, WEATHER_DELAY, NAS_DELAY, SECURITY_DELAY, LATE_AIRCRAFT_DELAY had over 80% missing data thus we felt it was appropriate to remove the data. In addition to these four variables, the last variable Unnamed.27 will need to be removed from the dataset since the missing data percentage is 100%. We know that cancelled flights take 2% of the data thus it was not considered especially, Since we are using naive Bayes, an exhaustive algorithm to build a classification model, thus its best we save computation power by reducing any features.

The result of the algorithm that 81% accuracy in classification of delayed flights is obtained with the Naive Bayes model. Cross validation and confusion matrix of training models were not generated as the computation power was not sufficient for the data set however code is provided to obtain such information.

I think its important to take into account Specific algorithms based on the dataset size. In one case we has a data set which was small and KNN model wouldn’t make sense while in the second scenario the data set was too large and once again KNN classification model didn’t make sense and instead a more computation friendly Naive Bias algorithm was chosen. Choices like these are relevant when choosing models for big data.

Appendix: Crime Data

```
library(ggcorrplot)

## Warning: package 'ggcorrplot' was built under R version 4.1.2

## Loading required package: ggplot2

library(caret)

## Loading required package: lattice

library(recipes)

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

##
## Attaching package: 'recipes'

## The following object is masked from 'package:stats':
##
##     step

library(dplyr)
library(heatmaply)

## Warning: package 'heatmaply' was built under R version 4.1.2

## Loading required package: plotly

## Warning: package 'plotly' was built under R version 4.1.2
```

```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout

## Loading required package: viridis

## Warning: package 'viridis' was built under R version 4.1.2

## Loading required package: viridisLite

##
## ======================
## Welcome to heatmaply version 1.3.0
##
## Type citation('heatmaply') for how to cite the package.
## Type ?heatmaply for the main documentation.
##
## The github page is: https://github.com/talgalili/heatmaply/
## Please submit your suggestions and bug-reports at: https://github.com/talg
alili/heatmaply/issues
## You may ask questions at stackoverflow, use the r and heatmaply tags:
##   https://stackoverflow.com/questions/tagged/heatmaply
## ======================

library(GGally)

## Warning: package 'GGally' was built under R version 4.1.2

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

library(MASS)

##
## Attaching package: 'MASS'

## The following object is masked from 'package:plotly':
##
##     select
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
train = read.csv("https://github.com/miansharip/SPS/raw/1b2ce2794a0d14f2d5dea
debc971a643c7cec104/Data%20621/HW1/crime-training-data_modified.csv")

eval = read.csv("https://github.com/miansharip/SPS/raw/1b2ce2794a0d14f2d5dead
ebc971a643c7cec104/Data%20621/HW1/crime-evaluation-data_modified.csv")

glimpse(train)
```

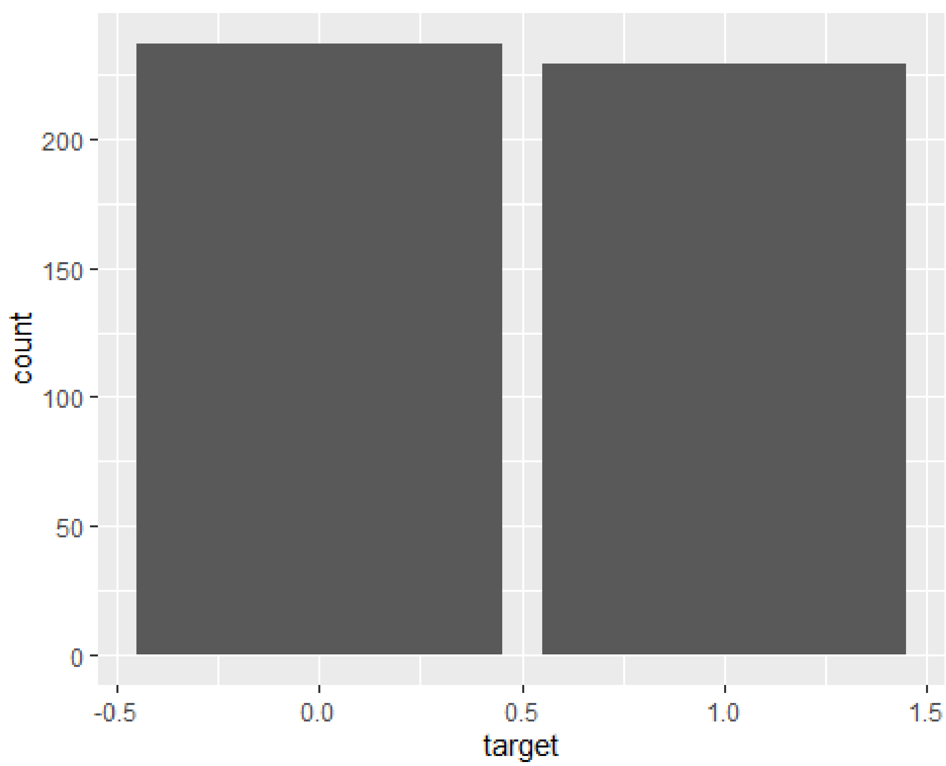
```
## Rows: 466
## Columns: 13
## $ zn      <dbl> 0, 0, 0, 30, 0, 0, 0, 0, 0, 80, 22, 0, 0, 22, 0, 0, 100, 2
0, 0~
## $ indus   <dbl> 19.58, 19.58, 18.10, 4.93, 2.46, 8.56, 18.10, 18.10, 5.19,
3.6~
## $ chas    <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0,~
## $ nox     <dbl> 0.605, 0.871, 0.740, 0.428, 0.488, 0.520, 0.693, 0.693, 0.
515,~
## $ rm      <dbl> 7.929, 5.403, 6.485, 6.393, 7.155, 6.781, 5.453, 4.519, 6.
316,~
## $ age     <dbl> 96.2, 100.0, 100.0, 7.8, 92.2, 71.3, 100.0, 100.0, 38.1, 1
9.1,~
## $ dis     <dbl> 2.0459, 1.3216, 1.9784, 7.0355, 2.7006, 2.8561, 1.4896, 1.
6582~
## $ rad     <int> 5, 5, 24, 6, 3, 5, 24, 24, 5, 1, 7, 5, 24, 7, 3, 3, 5, 5,
24, ~
## $ tax     <int> 403, 403, 666, 300, 193, 384, 666, 666, 224, 315, 330, 398
, 66~
## $ ptratio <dbl> 14.7, 14.7, 20.2, 16.6, 17.8, 20.9, 20.2, 20.2, 20.2, 16.4
, 19~
## $ lstat   <dbl> 3.70, 26.82, 18.85, 5.19, 4.82, 7.67, 30.59, 36.98, 5.68,
9.25~
## $ medv    <dbl> 50.0, 13.4, 15.4, 23.7, 37.9, 26.5, 5.0, 7.0, 22.2, 20.9,
24.8~
## $ target  <int> 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1
, 0,~
```

```
summary(train)
```

```
##        zn             indus             chas              nox
##  Min.   :  0.00   Min.   : 0.460   Min.   :0.00000   Min.   :0.3890
##  1st Qu.:  0.00   1st Qu.: 5.145   1st Qu.:0.00000   1st Qu.:0.4480
##  Median :  0.00   Median : 9.690   Median :0.00000   Median :0.5380
##  Mean   : 11.58   Mean   :11.105   Mean   :0.07082   Mean   :0.5543
##  3rd Qu.: 16.25   3rd Qu.:18.100   3rd Qu.:0.00000   3rd Qu.:0.6240
##  Max.   :100.00   Max.   :27.740   Max.   :1.00000   Max.   :0.8710
##        rm              age               dis               rad
```

```
##  Min.   :3.863   Min.   :  2.90   Min.   : 1.130   Min.   : 1.00
##  1st Qu.:5.887   1st Qu.: 43.88   1st Qu.: 2.101   1st Qu.: 4.00
##  Median :6.210   Median : 77.15   Median : 3.191   Median : 5.00
##  Mean   :6.291   Mean   : 68.37   Mean   : 3.796   Mean   : 9.53
##  3rd Qu.:6.630   3rd Qu.: 94.10   3rd Qu.: 5.215   3rd Qu.:24.00
##  Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.00
##       tax           ptratio          lstat             medv
##  Min.   :187.0   Min.   :12.6   Min.   : 1.730   Min.   : 5.00
##  1st Qu.:281.0   1st Qu.:16.9   1st Qu.: 7.043   1st Qu.:17.02
##  Median :334.5   Median :18.9   Median :11.350   Median :21.20
##  Mean   :409.5   Mean   :18.4   Mean   :12.631   Mean   :22.59
##  3rd Qu.:666.0   3rd Qu.:20.2   3rd Qu.:16.930   3rd Qu.:25.00
##  Max.   :711.0   Max.   :22.0   Max.   :37.970   Max.   :50.00
##      target
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.4914
##  3rd Qu.:1.0000
##  Max.   :1.0000

library(ggplot2)
ggplot(train, aes(x=target))+geom_bar()
```




```
corr=round(cor(train),2)
corr
```

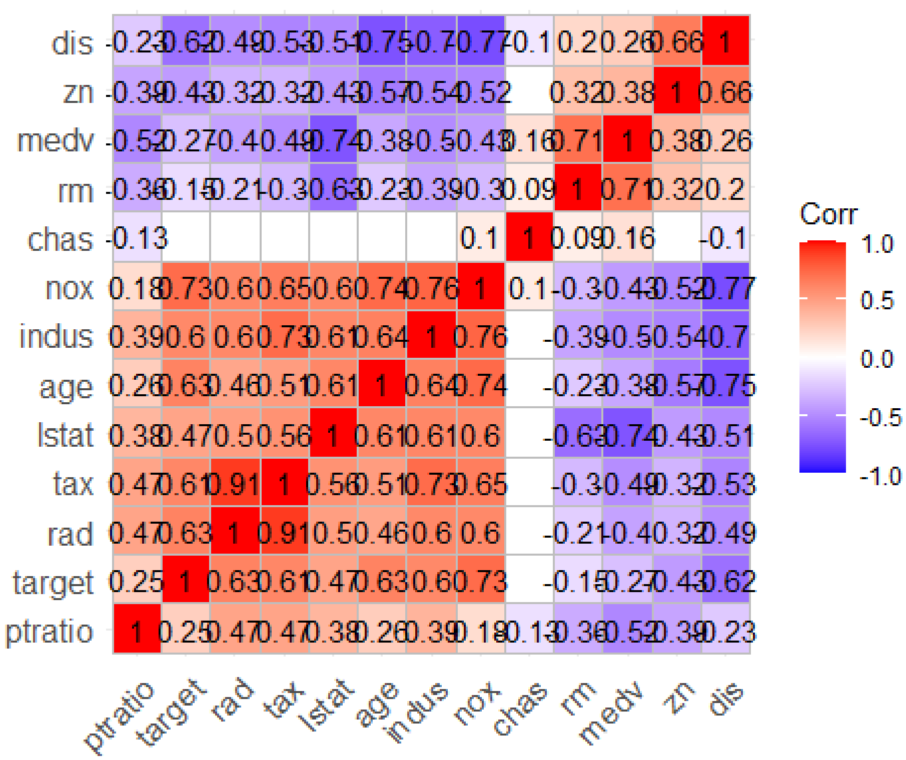
```
##             zn indus  chas   nox    rm   age   dis   rad   tax ptratio lsta
t
## zn       1.00 -0.54 -0.04 -0.52  0.32 -0.57  0.66 -0.32 -0.32   -0.39 -0.4
3
## indus   -0.54  1.00  0.06  0.76 -0.39  0.64 -0.70  0.60  0.73    0.39  0.6
1
## chas    -0.04  0.06  1.00  0.10  0.09  0.08 -0.10 -0.02 -0.05   -0.13 -0.0
5
## nox     -0.52  0.76  0.10  1.00 -0.30  0.74 -0.77  0.60  0.65    0.18  0.6
0
## rm       0.32 -0.39  0.09 -0.30  1.00 -0.23  0.20 -0.21 -0.30   -0.36 -0.6
3
## age     -0.57  0.64  0.08  0.74 -0.23  1.00 -0.75  0.46  0.51    0.26  0.6
1
## dis      0.66 -0.70 -0.10 -0.77  0.20 -0.75  1.00 -0.49 -0.53   -0.23 -0.5
1
## rad     -0.32  0.60 -0.02  0.60 -0.21  0.46 -0.49  1.00  0.91    0.47  0.5
0
## tax     -0.32  0.73 -0.05  0.65 -0.30  0.51 -0.53  0.91  1.00    0.47  0.5
6
## ptratio -0.39  0.39 -0.13  0.18 -0.36  0.26 -0.23  0.47  0.47    1.00  0.3
8
## lstat   -0.43  0.61 -0.05  0.60 -0.63  0.61 -0.51  0.50  0.56    0.38  1.0
0
## medv     0.38 -0.50  0.16 -0.43  0.71 -0.38  0.26 -0.40 -0.49   -0.52 -0.7
4
## target  -0.43  0.60  0.08  0.73 -0.15  0.63 -0.62  0.63  0.61    0.25  0.4
7
##          medv target
## zn       0.38  -0.43
## indus   -0.50   0.60
## chas     0.16   0.08
## nox     -0.43   0.73
## rm       0.71  -0.15
## age     -0.38   0.63
## dis      0.26  -0.62
## rad     -0.40   0.63
## tax     -0.49   0.61
## ptratio -0.52   0.25
## lstat   -0.74   0.47
## medv     1.00  -0.27
## target  -0.27   1.00
```

```
p.train=cor_pmat(train)
head(p.train)
```

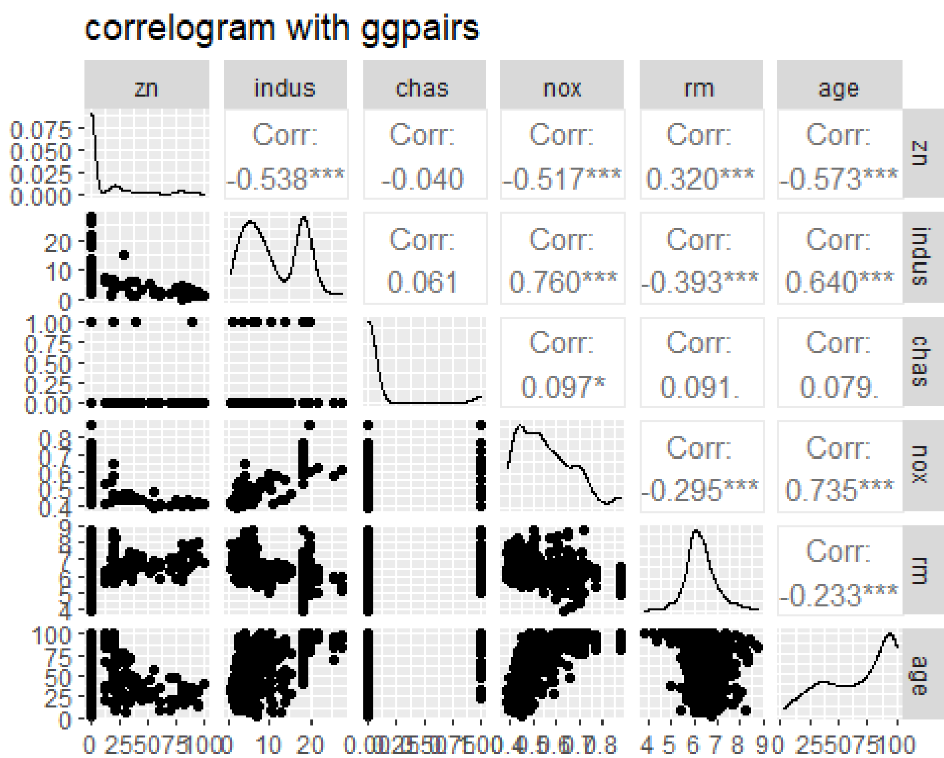
```
##                 zn        indus       chas          nox           rm
## zn    0.000000e+00 2.319176e-36 0.38703901 3.238655e-33 1.528592e-12
## indus 2.319176e-36 0.000000e+00 0.18735258 9.752376e-89 1.238277e-18
## chas  3.870390e-01 1.873526e-01 0.00000000 3.545359e-02 5.086704e-02
```

```
## nox    3.238655e-33 9.752376e-89 0.03545359 0.000000e+00 7.641839e-11
## rm     1.528592e-12 1.238277e-18 0.05086704 7.641839e-11 0.000000e+00
## age    6.040999e-42 5.733837e-55 0.08895386 2.347629e-80 3.730213e-07
##                 age          dis          rad          tax      ptratio
## zn     6.040999e-42 1.220397e-59 3.151098e-12 1.671074e-12 1.782358e-18
## indus  5.733837e-55 7.377362e-71 5.047931e-47 1.997555e-79 8.034926e-19
## chas   8.895386e-02 3.715218e-02 7.320938e-01 3.137655e-01 5.410667e-03
## nox    2.347629e-80 3.603637e-92 4.049854e-46 3.513041e-58 1.308313e-04
## rm     3.730213e-07 1.504750e-05 5.690434e-06 6.118602e-11 9.806780e-16
## age    0.000000e+00 1.225525e-85 8.132680e-26 1.616172e-32 2.236915e-08
##               lstat         medv       target
## zn     1.021425e-22 3.680497e-17 1.415603e-22
## indus  2.864171e-48 2.482578e-30 7.845651e-48
## chas   2.679367e-01 4.628588e-04 8.434811e-02
## nox    3.390324e-46 2.083171e-22 1.680131e-77
## rm     2.448543e-53 2.413698e-71 9.542249e-04
## age    5.572184e-48 2.733223e-17 6.245736e-53
```

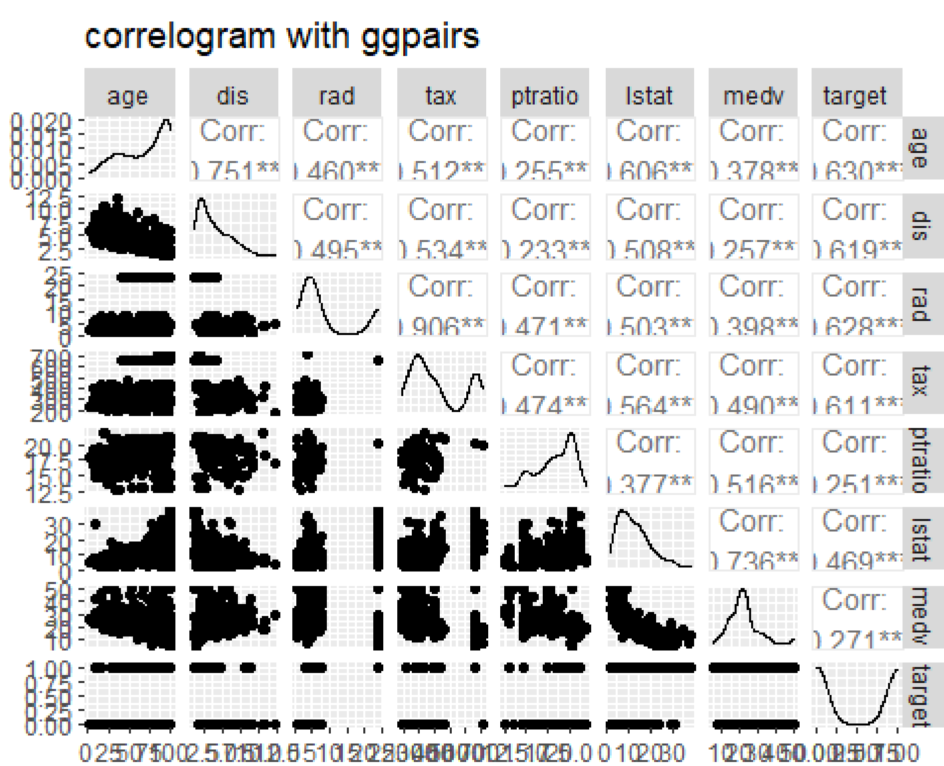
```
ggcorrplot(corr, hc.order = TRUE, lab=TRUE, p.mat =p.train, insig = "blank" )
```




```
train_df=train
ggpairs(train_df[0:6], title='correlogram with ggpairs')
```




```
ggpairs(train_df[6:13], title='correlogram with ggpairs')
```



```r
train$chas<-factor(train$chas)
train$target<-factor(train$target)
eval$chas<-factor(eval$chas)

Model1 <- glm(target ~ ., data = train, family = 'binomial')
summary(Model1)
```

```
##
## Call:
## glm(formula = target ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8464  -0.1445  -0.0017   0.0029   3.4665
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -40.822934   6.632913  -6.155 7.53e-10 ***
## zn           -0.065946   0.034656  -1.903  0.05706 .
## indus        -0.064614   0.047622  -1.357  0.17485
## chas1         0.910765   0.755546   1.205  0.22803
## nox          49.122297   7.931706   6.193 5.90e-10 ***
## rm           -0.587488   0.722847  -0.813  0.41637
## age           0.034189   0.013814   2.475  0.01333 *
## dis           0.738660   0.230275   3.208  0.00134 **
## rad           0.666366   0.163152   4.084 4.42e-05 ***
## tax          -0.006171   0.002955  -2.089  0.03674 *
## ptratio       0.402566   0.126627   3.179  0.00148 **
## lstat         0.045869   0.054049   0.849  0.39608
## medv          0.180824   0.068294   2.648  0.00810 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 192.05  on 453  degrees of freedom
## AIC: 218.05
##
## Number of Fisher Scoring iterations: 9
```

In second model

```r
Model2 <- glm(target ~ nox + age + rad + tax + ptratio + lstat,
               data = train,
               family = 'binomial')
summary(Model2)
```

```
##
## Call:
## glm(formula = target ~ nox + age + rad + tax + ptratio + lstat,
```

```
##     family = "binomial", data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.99446  -0.22115  -0.01487   0.00280   2.77983
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -24.483088   3.632470  -6.740 1.58e-11 ***
## nox          34.052772   5.228865   6.512 7.39e-11 ***
## age           0.020264   0.009997   2.027 0.042652 *
## rad           0.763304   0.140540   5.431 5.60e-08 ***
## tax          -0.009953   0.002603  -3.824 0.000132 ***
## ptratio       0.219358   0.090802   2.416 0.015701 *
## lstat        -0.012575   0.037437  -0.336 0.736940
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 214.35  on 459  degrees of freedom
## AIC: 228.35
##
## Number of Fisher Scoring iterations: 9
```

```
Model3 <- Model1 %>% stepAIC(direction = "backward", trace = FALSE)
summary(Model3)
```

```
##
## Call:
## glm(formula = target ~ zn + nox + age + dis + rad + tax + ptratio +
##     medv, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8295  -0.1752  -0.0021   0.0032   3.4191
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -37.415922   6.035013  -6.200 5.65e-10 ***
## zn           -0.068648   0.032019  -2.144  0.03203 *
## nox          42.807768   6.678692   6.410 1.46e-10 ***
## age           0.032950   0.010951   3.009  0.00262 **
## dis           0.654896   0.214050   3.060  0.00222 **
## rad           0.725109   0.149788   4.841 1.29e-06 ***
## tax          -0.007756   0.002653  -2.924  0.00346 **
## ptratio       0.323628   0.111390   2.905  0.00367 **
## medv          0.110472   0.035445   3.117  0.00183 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 197.32  on 457  degrees of freedom
## AIC: 215.32
##
## Number of Fisher Scoring iterations: 9
```

```
preds1 =predict(Model1, newdata = train)
preds2 =predict(Model1, newdata = train)
preds3 =predict(Model1, newdata = train)
preds1[preds1 >= 0.5] <- 1
preds1[preds1 < 0.5] <- 0
preds1 = as.factor(preds1)
preds2[preds2 >= 0.5] <- 1
preds2[preds2 < 0.5] <- 0
preds2 = as.factor(preds2)
preds3[preds3 >= 0.5] <- 1
preds3[preds3 < 0.5] <- 0
preds3 = as.factor(preds3)
```

```
train_CM1 <- confusionMatrix(preds1, train$target, mode = "everything")
```

```
## Registered S3 methods overwritten by 'proxy':
##   method               from
##   print.registry_field registry
##   print.registry_entry registry
```

```
train_CM2 <- confusionMatrix(preds2, train$target, mode = "everything")
train_CM3 <- confusionMatrix(preds3, train$target, mode = "everything")
```

```
train_CM1
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 225  35
##          1  12 194
##
##                Accuracy : 0.8991
##                  95% CI : (0.8681, 0.9249)
##     No Information Rate : 0.5086
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7979
##
##  Mcnemar's Test P-Value : 0.001332
##
```

```
##             Sensitivity : 0.9494
##             Specificity : 0.8472
##          Pos Pred Value : 0.8654
##          Neg Pred Value : 0.9417
##               Precision : 0.8654
##                  Recall : 0.9494
##                      F1 : 0.9054
##              Prevalence : 0.5086
##          Detection Rate : 0.4828
##    Detection Prevalence : 0.5579
##       Balanced Accuracy : 0.8983
##
##        'Positive' Class : 0
##
```

```
train_CM2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 225  35
##          1  12 194
##
##                Accuracy : 0.8991
##                  95% CI : (0.8681, 0.9249)
##     No Information Rate : 0.5086
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7979
##
##  Mcnemar's Test P-Value : 0.001332
##
##             Sensitivity : 0.9494
##             Specificity : 0.8472
##          Pos Pred Value : 0.8654
##          Neg Pred Value : 0.9417
##               Precision : 0.8654
##                  Recall : 0.9494
##                      F1 : 0.9054
##              Prevalence : 0.5086
##          Detection Rate : 0.4828
##    Detection Prevalence : 0.5579
##       Balanced Accuracy : 0.8983
##
##        'Positive' Class : 0
##
```

```
train_CM3
```

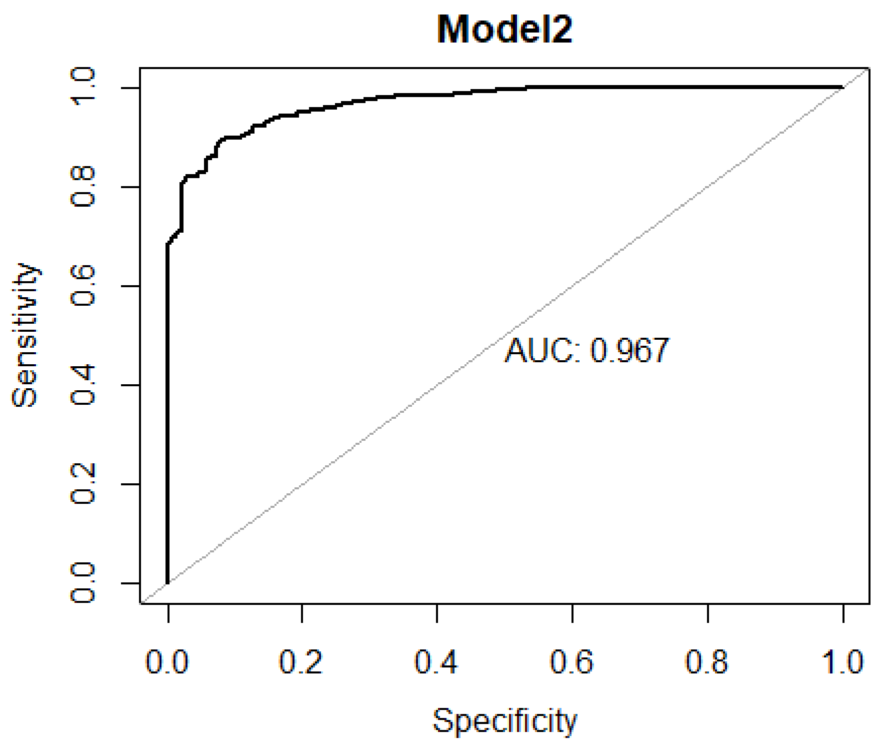
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 225  35
##          1  12 194
##
##                Accuracy : 0.8991
##                  95% CI : (0.8681, 0.9249)
##     No Information Rate : 0.5086
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7979
##
##  Mcnemar's Test P-Value : 0.001332
##
##             Sensitivity : 0.9494
##             Specificity : 0.8472
##          Pos Pred Value : 0.8654
##          Neg Pred Value : 0.9417
##               Precision : 0.8654
##                  Recall : 0.9494
##                      F1 : 0.9054
##              Prevalence : 0.5086
##          Detection Rate : 0.4828
##    Detection Prevalence : 0.5579
##       Balanced Accuracy : 0.8983
##
##        'Positive' Class : 0
##
```

```
getROC <- function(model) {
    name <- deparse(substitute(model))
    pred.prob1 <- predict(model, newdata = train)
    p1 <- data.frame(pred = train$target, prob = pred.prob1)
    p1 <- p1[order(p1$prob),]
    rocobj <- pROC::roc(p1$pred, p1$prob)
    plot(rocobj, asp=NA, legacy.axes = TRUE, print.auc=TRUE,
         xlab="Specificity", main = name)
}
par(mfrow=c(3,3))

getROC(Model2)
```

```
## Setting levels: control = 0, case = 1
```

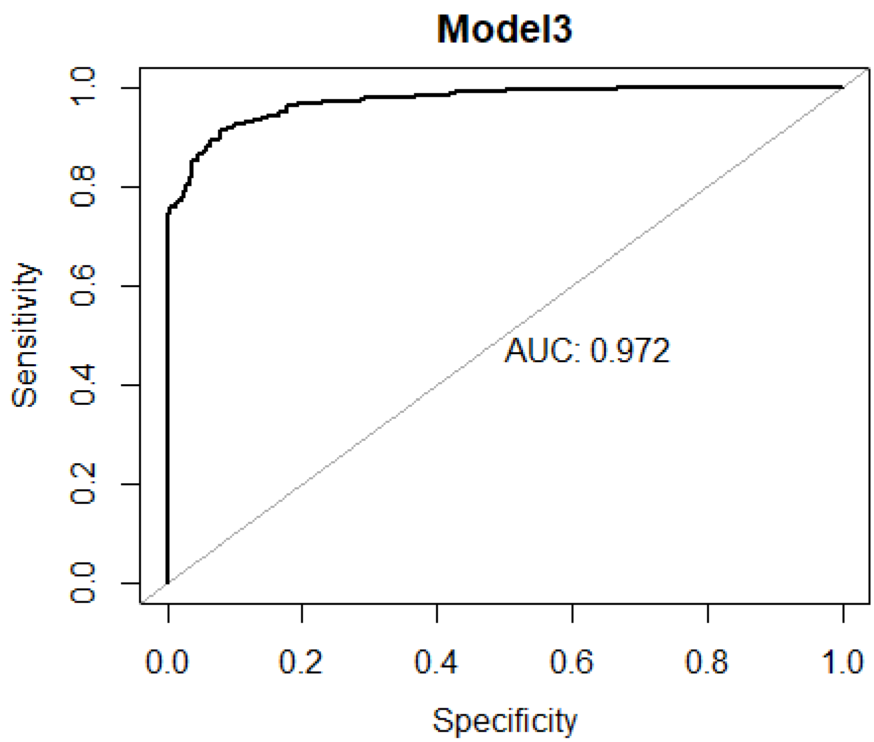
```
## Setting direction: controls < cases
```




```
getROC(Model3)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```




```
eval$chas <- as.factor(eval$chas)
prediction = predict(Model1, newdata = eval)
prediction[prediction >= 0.5] <- 1
prediction[prediction < 0.5] <- 0
prediction = as.factor(prediction)
prediction
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26
##  0  1  1  0  0  0  0  0  0  0  0  0  1  1  0  0  0  1  0  0  0  0  0  0  0
1
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40
##  0  1  1  1  1  1  1  1  1  1  1  1  1  0
## Levels: 0 1
```

---

title: "Data 622 HW1"

author: "Shariq Mian"

date: "10/9/2022"

output:

 html_document: default

```
  word_document: default

  pdf_document: default

---



```{r setup, include=FALSE}

knitr::opts_chunk$set(echo = TRUE)

```


In this data, we will work on 2018 flight delays dataset, where we have about 7.2 million fli
ghts.

We will explore the data, conduct a data preparation, and decide variables that could be use
d for naive Bayes classifier.

##Data Preparation

```{r DP, echo=FALSE,message = FALSE, warning = FALSE}

library(dplyr)

library(ggplot2)

flightdata.df <- read.csv("~/GitHub/SPS/Data 622/2018.csv")



#Reorder the columns in respective of preferred use (e.g. unnamed..27 string variable will
most probably not going to be used)

col.order <- c("FL_DATE","OP_CARRIER","ORIGIN","DEST","CRS_DEP_TIME","CRS_ARR_TIM
E","CRS_ELAPSED_TIME","ACTUAL_ELAPSED_TIME","DEP_TIME","ARR_TIME","DEP_DELA
Y","ARR_DELAY","CARRIER_DELAY","WEATHER_DELAY","NAS_DELAY","SECURITY_DELAY
","LATE_AIRCRAFT_DELAY","CANCELLED","DIVERTED","TAXI_OUT","TAXI_IN","WHEELS_
OFF","WHEELS_ON","CANCELLATION_CODE",

"AIR_TIME","DISTANCE","OP_CARRIER_FL_NUM","Unnamed..27")

flightdata.df <- flightdata.df[,col.order]

str(flightdata.df)

```
```

The structure of dataset indicates that there are 28 variables, that consists of a variety of categorical and continous variables.

```{r}

#Lets remove the variables with high missing data percentages >50% from our focus of analysis.

#flightdata.updated <- flightdata.df[,-c("CARRIER_DELAY","WEATHER_DELAY","NAS_DELAY","SECURITY_DELAY","LATE_AIRCRAFT_DELAY","Unnamed..27 ")]

flightdata.updated <- subset(flightdata.df, select = -c(CARRIER_DELAY,WEATHER_DELAY,NAS_DELAY,SECURITY_DELAY,LATE_AIRCRAFT_DELAY,Unnamed..27) )

str(flightdata.updated)

```

#Data manipulation: Conversion of Minutes to Hours and creation of Output Variable

```{r dm1, echo=FALSE, message = FALSE, warning = FALSE}

#Lets define the carrier, origin and destination as factors

flightdata.updated$ORIGIN <- as.factor(flightdata.updated$ORIGIN)

flightdata.updated$DEST <- as.factor(flightdata.updated$DEST)

flightdata.updated$OP_CARRIER <- as.factor(flightdata.updated$OP_CARRIER)

flightdata.updated$DIVERTED <- as.factor(flightdata.updated$DIVERTED)

#Analyzing and Creating the output variable

delay.breaks <- c(-150,0,15,3000)

delay.brackets <- c("Early", "Ontime", "Late")

FlightStatus <- cut(flightdata.updated$ARR_DELAY, breaks = delay.breaks, labels = delay.brackets)
```

```
#Lets create a variable to indicate whether a flight is delayed or not

flightdata.updated$delayed <- as.factor(ifelse(flightdata.updated$ARR_DELAY>15,1,0))


#Lets convert the unit of delay from minutes to hours and round it to 1 decimal units

flightdata.updated$ARR_DELAY_HRS <- round(flightdata.updated$ARR_DELAY/60,digits =
1)

#Lets do the same for departure delays

flightdata.updated$DEP_DELAY_HRS <- round(flightdata.updated$DEP_DELAY/60,digits =
1)


num.cf<- sum(flightdata.updated$CANCELLED)

per.cf <- 100*num.cf/nrow(flightdata.updated)


cf.df <- subset(flightdata.updated, flightdata.updated$CANCELLED>0)

#head(cf.df)


#Lets create morning, afternoon, evening, and redeye flight groups as dep_slot variable.

flightdata.updated$dep_slot <- NA

#define break points

flight.times <- c(0,600,1200,1700,2400)

flight.types <- c("Redeye","Morning","Afternoon", "Evening")

flightdata.updated$dep_slot <- cut(flightdata.updated$DEP_TIME, breaks = flight.times, lab
els = flight.types)


#Lets group the arrival of flights by the same time slots

flightdata.updated$arr_slot <- NA

flightdata.updated$arr_slot <- cut(flightdata.updated$ARR_TIME, breaks = flight.times, labe
ls = flight.types)
```

```
#convert fl_date character to date class
flightdata.updated$FL_DATE<-as.Date(flightdata.updated$FL_DATE)
class(flightdata.updated$FL_DATE)
#Lets create a variable that will keep the month
flightdata.updated$month <- months(flightdata.updated$FL_DATE)
#Lets create a variable that will keep the day of the week
flightdata.updated$day <- weekdays(flightdata.updated$FL_DATE)
#define the day and month variables as factors (categorical variable)
flightdata.updated$month<-as.factor(flightdata.updated$month)
flightdata.updated$day<- as.factor(flightdata.updated$day)
```

```
#DATA VISUALIZATION
``` {r Data Vis,echo=FALSE, message = FALSE, warning = FALSE}
#Cont. table of flights by flight status (early, ontime, delayed)
prop.table(round(table(FlightStatus), digits=2))

#Contingency table of delayed(1) and nondelayed (0) flights
ct.Delayed <- prop.table(table(flightdata.updated$delayed))
ct.Delayed <- as.data.frame(100*round(ct.Delayed,digits = 2))
rownames(ct.Delayed) <- c("Ontime or Early", "Delayed")
colnames(ct.Delayed) <- c("Code", "Percentage")
ct.Delayed

#CANCELLED FLIGHTS
"Percent of Cancelled Flights:"
round(per.cf,digits=2)
```
```

````
```


```{r data split,echo=FALSE, message = FALSE, warning = FALSE}

flightdata.expdata <- flightdata.updated[,c("OP_CARRIER","ORIGIN","DEST","DIVERTED","month", "day","dep_slot","arr_slot","delayed")]


#Lets tabularize the structure of flightdata.expdata

library(knitr)

library(magrittr)

data.frame(variable = names(flightdata.expdata),

           class = sapply(flightdata.expdata, typeof),

           First_values = sapply(flightdata.expdata, function(x) paste0(head(x),  collapse = ", ")),

           row.names = NULL) %>%

  kable()


# Create training and validation sets.

set.seed(1905)

split <- sample(nrow(flightdata.expdata),nrow(flightdata.expdata)*0.75) #75-25 split is used.

flightdata.data.train <- as.data.frame(flightdata.expdata[split,])

flightdata.data.test <- as.data.frame(flightdata.expdata[-split,])

```


#Building Naive Bayes: naive Bayes classifier model will be developed with the train data.

```{r nb model,echo=FALSE, message = FALSE, warning = FALSE}

library(e1071)

library(FNN)

library(caret)
````

```
flightdata.naive <- naiveBayes(delayed~.,data=flightdata.data.train)

flightdata.naive

#Below is a memory exhaustive and would take a good amount of time to check the conf. m
atrix of train data.Left for the future experimentation.

#pred.class.train <- predict(flightdata.naive, newdata = flightdata.data.train, type = "class")

#train.cf <- confusionMatrix(pred.class.train, flightdata.data.train$delayed)

#train.cf


#Following (caret approach with trainControl includes cross validation)

#However, this training with 10 fold CV takes exponentially higher time

#Therefore, its left here for future experimentation

#train_control <- trainControl(method="cv", number=10)

#flightdata.naive.model = train(x=flightdata.data.train[, c(1:9)],y=flightdata.data.train[, 10],
method ='nb',trControl=train_control)

#str(flightdata.data.train)
```

Lets take a look at the naive Bayes classification model performance with the test data

```{r prob and class mem,echo=FALSE, message = FALSE, warning = FALSE}

library(caret)

#probabilities

pred.prob.fd <- predict(flightdata.naive, newdata = flightdata.data.test, type = "raw")

## predict class membership

pred.class <- predict(flightdata.naive, newdata = flightdata.data.test, type = "class")

test.cf <- confusionMatrix(pred.class, flightdata.data.test$delayed)

test.cf
```

```
```

```{r predicting a new flight,echo=FALSE, message = FALSE, warning = FALSE}
#Define the new passenger data
new.pass.data <- c("AA", "ABQ","JFK",0,"October", "Wednesday","Redeye","Morning","")
#Get the first row from the experiment data to have the column names and structure
new.passenger <- flightdata.expdata[1,]
#Update the passenger data with the new passenger data
new.passenger[,c(1:9)] <- new.pass.data
#Predict the probability of delay
pred.prob.fd <- predict(flightdata.naive, newdata = new.passenger, type = "raw")
pred.prob.fd
## predict class membership
pred.class <- predict(flightdata.naive, newdata = new.passenger, type = "class")
pred.class
#Prediction of new passenger
print(pred.class)
```

##Data Preparation

```
## 'data.frame':    7213446 obs. of  28 variables:
##  $ FL_DATE            : chr  "2018-01-01" "2018-01-01" "2018-01-01" "2018-01-01" ...
##  $ OP_CARRIER         : chr  "UA" "UA" "UA" "UA" ...
##  $ ORIGIN             : chr  "EWR" "LAS" "SNA" "RSW" ...
##  $ DEST               : chr  "DEN" "SFO" "DEN" "ORD" ...
```

```
##  $ CRS_DEP_TIME       : int  1517 1115 1335 1546 630 2241 750 1324 2224 16
01 ...
##  $ CRS_ARR_TIME       : int  1745 1254 1649 1756 922 14 916 1619 638 1813
...
##  $ CRS_ELAPSED_TIME   : num  268 99 134 190 112 93 206 115 314 252 ...
##  $ ACTUAL_ELAPSED_TIME: num  250 83 126 182 106 79 193 102 299 237 ...
##  $ DEP_TIME           : num  1512 1107 1330 1552 650 ...
##  $ ARR_TIME           : num  1722 1230 1636 1754 936 ...
##  $ DEP_DELAY          : num  -5 -8 -5 6 20 3 -3 -6 13 -2 ...
##  $ ARR_DELAY          : num  -23 -24 -13 -2 14 -11 -16 -19 -2 -17 ...
##  $ CARRIER_DELAY      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ WEATHER_DELAY      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ NAS_DELAY          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ SECURITY_DELAY     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ LATE_AIRCRAFT_DELAY: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ CANCELLED          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ DIVERTED           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ TAXI_OUT           : num  15 11 15 19 13 15 14 11 10 12 ...
##  $ TAXI_IN            : num  10 7 5 6 10 2 6 6 9 8 ...
##  $ WHEELS_OFF         : num  1527 1118 1345 1611 703 ...
##  $ WHEELS_ON          : num  1712 1223 1631 1748 926 ...
##  $ CANCELLATION_CODE  : chr  "" "" "" "" ...
##  $ AIR_TIME           : num  225 65 106 157 83 62 173 85 280 217 ...
##  $ DISTANCE           : num  1605 414 846 1120 723 ...
##  $ OP_CARRIER_FL_NUM  : int  2429 2427 2426 2425 2424 2422 2421 2420 2419
2418 ...
##  $ Unnamed..27        : logi  NA NA NA NA NA NA ...
```

The structure of dataset indicates that there are 28 variables, that consists of a variety of categorical and continous variables.

```
#Lets remove the variables with high missing data percentages >50% from our f
ocus of analysis.
#fd.updated <- fd.df[,-c("CARRIER_DELAY","WEATHER_DELAY","NAS_DELAY","SECURIT
Y_DELAY","LATE_AIRCRAFT_DELAY","Unnamed..27 ")]
fd.updated <- subset(fd.df, select = -c(CARRIER_DELAY,WEATHER_DELAY,NAS_DELAY
,SECURITY_DELAY,LATE_AIRCRAFT_DELAY,Unnamed..27) )
str(fd.updated)
```

```
## 'data.frame':    7213446 obs. of  22 variables:
##  $ FL_DATE            : chr  "2018-01-01" "2018-01-01" "2018-01-01" "2018-
01-01" ...
##  $ OP_CARRIER         : chr  "UA" "UA" "UA" "UA" ...
##  $ ORIGIN             : chr  "EWR" "LAS" "SNA" "RSW" ...
##  $ DEST               : chr  "DEN" "SFO" "DEN" "ORD" ...
##  $ CRS_DEP_TIME       : int  1517 1115 1335 1546 630 2241 750 1324 2224 16
01 ...
##  $ CRS_ARR_TIME       : int  1745 1254 1649 1756 922 14 916 1619 638 1813
...
##  $ CRS_ELAPSED_TIME   : num  268 99 134 190 112 93 206 115 314 252 ...
##  $ ACTUAL_ELAPSED_TIME: num  250 83 126 182 106 79 193 102 299 237 ...
##  $ DEP_TIME           : num  1512 1107 1330 1552 650 ...
##  $ ARR_TIME           : num  1722 1230 1636 1754 936 ...
##  $ DEP_DELAY          : num  -5 -8 -5 6 20 3 -3 -6 13 -2 ...
##  $ ARR_DELAY          : num  -23 -24 -13 -2 14 -11 -16 -19 -2 -17 ...
##  $ CANCELLED          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ DIVERTED           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ TAXI_OUT           : num  15 11 15 19 13 15 14 11 10 12 ...
##  $ TAXI_IN            : num  10 7 5 6 10 2 6 6 9 8 ...
##  $ WHEELS_OFF         : num  1527 1118 1345 1611 703 ...
##  $ WHEELS_ON          : num  1712 1223 1631 1748 926 ...
##  $ CANCELLATION_CODE  : chr  "" "" "" "" ...
##  $ AIR_TIME           : num  225 65 106 157 83 62 173 85 280 217 ...
##  $ DISTANCE           : num  1605 414 846 1120 723 ...
##  $ OP_CARRIER_FL_NUM  : int  2429 2427 2426 2425 2424 2422 2421 2420 2419
2418 ...
```

#Data manipulation: Conversion of Minutes to Hours and creation of Output Variable

```
## [1] "Date"
```

#DATA VISUALIZATION

```
## FlightStatus
##     Early    Ontime      Late
## 0.6444452 0.1711120 0.1844428
##                  Code Percentage
```

```
## Ontime or Early    0          82
## Delayed            1          18
## [1] "Percent of Cancelled Flights:"
## [1] 1.62
```

.

#Data Partition In this section, we will look at the final structure of the data and partition it into train and test subsets.

variable	class	First_values
OP_CARRIER	integer	UA, UA, UA, UA, UA, UA
ORIGIN	integer	EWR, LAS, SNA, RSW, ORD, ORD
DEST	integer	DEN, SFO, DEN, ORD, ALB, OMA
DIVERTED	integer	0, 0, 0, 0, 0, 0
month	integer	January, January, January, January, January, January
day	integer	Monday, Monday, Monday, Monday, Monday, Monday
dep_slot	integer	Afternoon, Morning, Afternoon, Afternoon, Morning, Evening
arr_slot	integer	Evening, Afternoon, Afternoon, Evening, Morning, Redeye
delayed	integer	0, 0, 0, 0, 0, 0

#Building Naive Bayes: naive Bayes classifier model will be developed with the train data.

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
```

```
## Y
##         0         1
## 0.8156407 0.1843593
##
## Conditional probabilities:
##    OP_CARRIER
## Y             9E          AA          AS          B6          DL           E
V
##   0 0.033509736 0.125575403 0.035528022 0.037867173 0.142062884 0.02705639
6
##   1 0.032856533 0.133540665 0.029342869 0.060518199 0.096071810 0.03090551
9
##    OP_CARRIER
## Y             F9          G4          HA          MQ          NK           O
H
##   0 0.014611031 0.012942948 0.013055216 0.039723756 0.024862547 0.03669759
7
##   1 0.025520560 0.015858497 0.006205632 0.042206883 0.023264988 0.04151907
2
##    OP_CARRIER
## Y             OO          UA          VX          WN          YV           Y
X
##   0 0.107715756 0.086564758 0.002417231 0.187388959 0.029024785 0.04339580
2
##   1 0.106216261 0.088638747 0.002555020 0.191103217 0.031628079 0.04204745
0
##
##    ORIGIN
## Y             ABE          ABI          ABQ          ABR          ABY
##   0 5.770491e-04 2.781294e-04 3.437115e-03 1.108821e-04 1.344446e-04
##   1 5.815225e-04 2.800302e-04 3.005725e-03 7.154055e-05 1.543232e-04
##    ORIGIN
## Y             ACK          ACT          ACV          ACY          ADK
##   0 1.196603e-04 2.275394e-04 1.951988e-04 4.853403e-04 1.155022e-05
##   1 1.911155e-04 1.890715e-04 2.422159e-04 3.352186e-04 2.146217e-05
##    ORIGIN
## Y             ADQ          AEX          AGS          AKN          ALB
##   0 9.586685e-05 4.781792e-04 6.186299e-04 7.623147e-06 1.736460e-03
```

```
##   1 5.110040e-05 4.445734e-04 6.264909e-04 1.022008e-05 1.564694e-03
##    ORIGIN
## Y           ALO          AMA          ANC          APN          ART
##   0 8.593365e-05 7.777920e-04 2.864686e-03 8.986073e-05 2.310044e-06
##   1 9.913477e-05 6.162708e-04 1.217211e-03 6.745252e-05 4.088032e-06
##    ORIGIN
## Y           ASE          ATL          ATW          AUS          AVL
##   0 8.725038e-04 5.619507e-02 6.050006e-04 8.947495e-03 1.055921e-03
##   1 1.039382e-03 4.733021e-02 5.294001e-04 8.241472e-03 1.122165e-03
##    ORIGIN
## Y           AVP          AZA          AZO          BDL          BET
##   0 4.680150e-04 7.486854e-04 4.007927e-04 4.034262e-03 1.286695e-04
##   1 3.944951e-04 6.653272e-04 3.290865e-04 3.329702e-03 5.518843e-05
##    ORIGIN
## Y           BFF          BFL          BGM          BGR          BHM
##   0 8.339261e-05 3.386525e-04 1.302865e-04 5.306172e-04 2.563918e-03
##   1 5.723244e-05 2.371058e-04 9.709075e-05 6.316009e-04 2.377190e-03
##    ORIGIN
## Y           BIL          BIS          BJI          BKG          BLI
##   0 6.287941e-04 5.269211e-04 1.094961e-04 8.547165e-06 3.361115e-04
##   1 4.006271e-04 4.118692e-04 7.665059e-05 1.941815e-05 2.033796e-04
##    ORIGIN
## Y           BLV          BMI          BNA          BOI          BOS
##   0 1.277455e-04 4.264342e-04 1.024990e-02 2.854522e-03 1.942978e-02
##   1 2.074676e-04 4.476395e-04 1.049193e-02 1.945903e-03 2.432788e-02
##    ORIGIN
## Y           BPT          BQK          BQN          BRD          BRO
##   0 1.198913e-04 1.533870e-04 2.601110e-04 9.540484e-05 3.880875e-04
##   1 8.891469e-05 1.154869e-04 3.444167e-04 6.336449e-05 2.575460e-04
##    ORIGIN
## Y           BRW          BTM          BTR          BTV          BUF
##   0 1.101891e-04 1.032590e-04 1.056383e-03 1.300786e-03 3.652180e-03
##   1 4.701236e-05 3.679229e-05 1.047558e-03 1.420591e-03 3.421683e-03
##    ORIGIN
```

```
## Y            BUR          BWI          BZN          CAE          CAK
##   0 3.684290e-03 1.468241e-02 8.309230e-04 9.898541e-04 9.813069e-04
##   1 3.536147e-03 1.483138e-02 7.092735e-04 7.338017e-04 1.212101e-03
##    ORIGIN
## Y            CDC          CDV          CGI          CHA          CHO
##   0 9.448082e-05 1.108821e-04 8.293060e-05 1.171193e-03 7.387522e-04
##   1 6.132048e-05 6.438650e-05 7.358457e-05 1.245828e-03 8.748388e-04
##    ORIGIN
## Y            CHS          CID          CIU          CKB          CLE
##   0 3.328543e-03 1.297321e-03 8.708868e-05 1.064930e-04 6.768430e-03
##   1 2.889216e-03 1.239696e-03 1.011788e-04 1.226410e-04 6.240380e-03
##    ORIGIN
## Y            CLL          CLT          CMH          CMI          CMX
##   0 3.240992e-04 3.109204e-02 6.357473e-03 3.252543e-04 8.731968e-05
##   1 2.391499e-04 3.609732e-02 6.311921e-03 2.882062e-04 1.236630e-04
##    ORIGIN
## Y            CNY          COD          COS          COU          CPR
##   0 6.029216e-05 1.212773e-04 1.414209e-03 3.125490e-04 1.573140e-04
##   1 4.803437e-05 7.256256e-05 1.550386e-03 2.984263e-04 5.723244e-05
##    ORIGIN
## Y            CRP          CRW          CSG          CVG          CWA
##   0 7.814880e-04 6.246360e-04 1.734843e-04 6.771202e-03 4.019477e-04
##   1 6.888333e-04 6.939434e-04 1.768074e-04 7.072295e-03 3.331746e-04
##    ORIGIN
## Y            CYS          DAB          DAL          DAY          DBQ
##   0 7.854151e-06 5.128299e-04 9.278294e-03 1.880607e-03 1.328276e-04
##   1 1.022008e-05 4.067592e-04 1.253186e-02 2.041972e-03 1.297950e-04
##    ORIGIN
## Y            DCA          DEN          DFW          DHN          DLG
##   0 1.801742e-02 3.278392e-02 3.695678e-02 1.878066e-04 1.108821e-05
##   1 1.869763e-02 3.431392e-02 4.581764e-02 2.279078e-04 1.022008e-05
##    ORIGIN
## Y            DLH          DRO          DRT          DSM          DTW
##   0 3.936316e-04 4.317473e-04 1.663232e-05 2.203089e-03 2.263266e-02
```

```
##   1 3.474827e-04 4.363974e-04 1.430811e-05 1.986783e-03 1.885298e-02
##    ORIGIN
## Y            DVL          EAR          EAU          ECP          EGE
##   0 8.547165e-05 3.303364e-05 1.007179e-04 7.951173e-04 2.862145e-04
##   1 5.927646e-05 1.022008e-05 9.709075e-05 6.101387e-04 3.485047e-04
##    ORIGIN
## Y            EKO          ELM          ELP          ERI          ESC
##   0 1.081101e-04 4.897294e-05 2.427395e-03 1.515389e-04 8.570265e-05
##   1 3.168225e-05 4.803437e-05 2.061390e-03 1.185529e-04 7.358457e-05
##    ORIGIN
## Y            EUG          EVV          EWN          EWR          EYW
##   0 5.715050e-04 6.442714e-04 3.046949e-04 1.817658e-02 6.881622e-04
##   1 5.416642e-04 5.508623e-04 2.381278e-04 2.598353e-02 5.253121e-04
##    ORIGIN
## Y            FAI          FAR          FAT          FAY          FCA
##   0 7.154208e-04 8.050505e-04 1.570830e-03 5.417054e-04 3.481237e-04
##   1 2.647001e-04 7.235816e-04 1.460449e-03 4.936298e-04 2.422159e-04
##    ORIGIN
## Y            FLG          FLL          FLO          FNT          FSD
##   0 1.732533e-04 1.317003e-02 2.425547e-05 5.782041e-04 8.628016e-04
##   1 1.134429e-04 1.557438e-02 2.657221e-05 5.273561e-04 9.770396e-04
##    ORIGIN
## Y            FSM          FWA          GCC          GCK          GEG
##   0 2.829804e-04 9.753008e-04 1.252044e-04 1.030280e-04 1.863282e-03
##   1 2.953603e-04 1.055734e-03 7.256256e-05 7.256256e-05 1.236630e-03
##    ORIGIN
## Y            GFK          GGG          GJT          GNV          GPT
##   0 3.171691e-04 1.101891e-04 5.705810e-04 6.343382e-04 6.255600e-04
##   1 2.054236e-04 1.001568e-04 2.790082e-04 6.019627e-04 4.415074e-04
##    ORIGIN
## Y            GRB          GRI          GRK          GRR          GSO
##   0 6.948614e-04 1.284385e-04 4.324403e-04 2.400598e-03 1.783585e-03
##   1 5.682364e-04 1.665873e-04 3.587248e-04 2.356750e-03 1.702665e-03
##    ORIGIN
```

```
## Y              GSP          GST          GTF          GTR          GUC
##    0 1.935124e-03 1.362926e-05 2.802084e-04 1.453018e-04 3.626770e-05
##    1 1.573892e-03 5.110040e-06 1.573892e-04 1.195749e-04 3.168225e-05
##    ORIGIN
## Y              GUM          HDN          HGR          HHH          HIB
##    0 9.147776e-05 1.258974e-04 1.501529e-05 7.345941e-05 9.910091e-05
##    1 6.234248e-05 1.226410e-04 2.963823e-05 6.847453e-05 3.270425e-05
##    ORIGIN
## Y              HLN          HNL          HOB          HOU          HPN
##    0 2.850595e-04 7.755050e-03 1.062620e-04 7.732874e-03 1.244421e-03
##    1 1.216189e-04 3.358318e-03 6.336449e-05 9.235886e-03 1.738435e-03
##    ORIGIN
## Y              HRL          HSV          HTS          HVN          HYA
##    0 4.767932e-04 1.240494e-03 1.092651e-04 1.284385e-04 1.155022e-05
##    1 4.721677e-04 9.749956e-04 1.737413e-04 1.389931e-04 1.533012e-05
##    ORIGIN
## Y              HYS          IAD          IAG          IAH          ICT
##    0 1.506149e-04 9.556654e-03 1.064930e-04 2.475051e-02 1.526939e-03
##    1 1.400151e-04 9.314580e-03 1.318390e-04 2.193331e-02 1.257070e-03
##    ORIGIN
## Y              IDA          IFP          ILM          IMT          IND
##    0 3.307984e-04 5.775111e-06 9.189357e-04 9.540484e-05 6.828029e-03
##    1 2.299518e-04 1.022008e-05 7.419777e-04 6.540851e-05 6.364043e-03
##    ORIGIN
## Y              INL          ISN          ISP          ITH          ITO
##    0 9.886990e-05 2.018979e-04 8.424732e-04 1.397577e-04 1.015496e-03
##    1 6.438650e-05 1.676093e-04 8.809708e-04 1.205969e-04 3.055804e-04
##    ORIGIN
## Y              JAC          JAN          JAX          JFK          JLN
##    0 5.754321e-04 1.093575e-03 4.401328e-03 1.737939e-02 1.002559e-04
##    1 6.234248e-04 9.668195e-04 4.146286e-03 1.932924e-02 1.430811e-04
##    ORIGIN
## Y              JMS          JNU          KOA          KTN          LAN
##    0 1.432228e-04 6.777670e-04 2.352318e-03 3.543608e-04 5.058997e-04
```

```
##   1 1.062888e-04 3.566808e-04 9.985017e-04 2.493699e-04 4.231113e-04
##    ORIGIN
## Y             LAR          LAS          LAW          LAX          LBB
##   0 8.893671e-05 2.270150e-02 1.783354e-04 3.175133e-02 9.133916e-04
##   1 8.380465e-05 2.215815e-02 1.829394e-04 2.739799e-02 7.859241e-04
##    ORIGIN
## Y             LBE          LBF          LBL          LCH          LCK
##   0 1.607791e-04 9.193977e-05 8.269959e-05 2.873695e-04 1.386027e-04
##   1 9.095870e-05 2.350618e-05 5.212240e-05 1.941815e-04 2.289298e-04
##    ORIGIN
## Y             LEX          LFT          LGA          LGB          LIH
##   0 1.406355e-03 6.401133e-04 2.231341e-02 2.411455e-03 2.216026e-03
##   1 1.413437e-03 5.590383e-04 2.650986e-02 2.031752e-03 1.035294e-03
##    ORIGIN
## Y             LIT          LNK          LRD          LSE          LWB
##   0 1.938127e-03 2.520259e-04 3.446586e-04 2.986887e-04 7.253540e-05
##   1 1.520748e-03 2.371058e-04 2.851402e-04 2.463039e-04 1.083328e-04
##    ORIGIN
## Y             LWS          LYH          MAF          MBS          MCI
##   0 1.173503e-04 1.395267e-04 1.240494e-03 4.151150e-04 8.201813e-03
##   1 3.577028e-05 1.665873e-04 1.138517e-03 3.239765e-04 6.308855e-03
##    ORIGIN
## Y             MCO          MDT          MDW          MEI          MEM
##   0 1.881832e-02 8.815130e-04 1.119655e-02 1.411437e-04 3.216275e-03
##   1 2.138041e-02 8.135183e-04 1.532296e-02 2.033796e-04 3.073178e-03
##    ORIGIN
## Y             MFE          MFR          MGM          MHK          MHT
##   0 6.551286e-04 6.384963e-04 5.680399e-04 2.598800e-04 1.400580e-03
##   1 5.784565e-04 7.205156e-04 5.835665e-04 2.187097e-04 9.790836e-04
##    ORIGIN
## Y             MIA          MKE          MKG          MLB          MLI
##   0 1.201801e-02 4.590982e-03 9.748388e-05 3.668351e-04 6.870072e-04
##   1 1.313996e-02 3.981743e-03 9.913477e-05 2.330178e-04 7.123395e-04
##    ORIGIN
```

```
## Y              MLU          MMH          MOB          MOT          MQT
##    0 4.439905e-04 1.709433e-05 9.062304e-04 3.730722e-04 1.785664e-04
##    1 3.873410e-04 3.679229e-05 8.543986e-04 2.687881e-04 2.115556e-04
##     ORIGIN
## Y              MRY          MSN          MSO          MSP          MSY
##    0 5.521006e-04 1.868133e-03 5.077478e-04 2.297547e-02 7.845835e-03
##    1 5.539283e-04 1.510528e-03 3.679229e-04 1.896642e-02 7.677323e-03
##     ORIGIN
## Y              MTJ          MVY          MYR          OAJ          OAK
##    0 2.058250e-04 6.075417e-05 1.601785e-03 4.225071e-04 7.439036e-03
##    1 2.207537e-04 8.687067e-05 1.445119e-03 3.863190e-04 6.774890e-03
##     ORIGIN
## Y              OGD          OGG          OGS          OKC          OMA
##    0 1.478428e-05 3.869555e-03 1.686332e-05 3.371279e-03 3.552617e-03
##    1 2.452819e-05 2.283166e-03 2.759421e-05 2.723651e-03 3.143696e-03
##     ORIGIN
## Y              OME          ONT          ORD          ORF          ORH
##    0 1.120372e-04 3.095460e-03 4.277301e-02 3.098463e-03 1.090341e-04
##    1 4.701236e-05 2.440555e-03 5.984470e-02 3.133476e-03 2.227977e-04
##     ORIGIN
## Y              OTH          OTZ          OWB          PAH          PBG
##    0 3.950176e-05 1.069551e-04 1.339826e-05 9.540484e-05 1.321345e-04
##    1 1.022008e-04 6.132048e-05 2.759421e-05 1.042448e-04 1.471691e-04
##     ORIGIN
## Y              PBI          PDX          PGD          PGV          PHF
##    0 3.357650e-03 9.385480e-03 7.422173e-04 1.064930e-04 3.686831e-04
##    1 4.002183e-03 6.492816e-03 6.632831e-04 1.502352e-04 3.658788e-04
##     ORIGIN
## Y              PHL          PHX          PIA          PIB          PIE
##    0 1.550594e-02 2.457287e-02 7.357492e-04 1.032590e-04 1.092189e-03
##    1 1.838899e-02 2.301766e-02 7.041635e-04 8.176063e-05 8.727948e-04
##     ORIGIN
## Y              PIH          PIT          PLN          PNS          PPG
##    0 1.931197e-04 6.846279e-03 1.496909e-04 1.656302e-03 1.455328e-05
```

```
##   1 9.504674e-05 6.409012e-03 9.709075e-05 1.233564e-03 2.963823e-05
##    ORIGIN
## Y              PRC          PSC          PSE          PSG          PSM
##   0 3.349564e-05 4.298993e-04 1.067241e-04 1.106511e-04 3.742272e-05
##   1 3.679229e-05 2.933163e-04 1.287730e-04 6.336449e-05 6.234248e-05
##    ORIGIN
## Y              PSP          PUB          PVD          PVU          PWM
##   0 1.514003e-03 1.349066e-04 2.622824e-03 7.276640e-05 1.610332e-03
##   1 1.268312e-03 1.022008e-04 2.296452e-03 8.380465e-05 1.657697e-03
##    ORIGIN
## Y              RAP          RDD          RDM          RDU          RFD
##   0 5.999185e-04 1.556970e-04 5.227631e-04 8.082846e-03 9.586685e-05
##   1 5.948086e-04 1.921375e-04 5.416642e-04 9.438243e-03 9.504674e-05
##    ORIGIN
## Y              RHI          RIC          RKS          RNO          ROA
##   0 1.071861e-04 3.080444e-03 1.014110e-04 2.580782e-03 3.224822e-04
##   1 5.927646e-05 3.541257e-03 6.029847e-05 2.371058e-03 4.404854e-04
##    ORIGIN
## Y              ROC          ROW          RST          RSW          SAF
##   0 1.983635e-03 1.977398e-04 4.333643e-04 4.580356e-03 2.037459e-04
##   1 1.875385e-03 1.665873e-04 4.782997e-04 4.140154e-03 1.614773e-04
##    ORIGIN
## Y              SAN          SAT          SAV          SBA          SBN
##   0 1.288011e-02 5.705348e-03 2.206554e-03 9.930881e-04 9.549724e-04
##   1 1.152621e-02 4.757447e-03 2.024598e-03 9.228732e-04 9.780616e-04
##    ORIGIN
## Y              SBP          SCC          SCE          SCK          SDF
##   0 6.479675e-04 1.464568e-04 1.977398e-04 8.177557e-05 3.163144e-03
##   1 7.399337e-04 5.927646e-05 1.788514e-04 2.248417e-04 3.198885e-03
##    ORIGIN
## Y              SEA          SFB          SFO          SGF          SGU
##   0 2.018424e-02 1.305175e-03 2.345758e-02 1.162414e-03 5.024347e-04
##   1 1.727296e-02 1.506440e-03 2.902298e-02 1.294884e-03 2.575460e-04
##    ORIGIN
```

```
## Y             SHD          SHV          SIT          SJC          SJT
##   0 9.609785e-05 9.392641e-04 2.065180e-04 7.824814e-03 2.104451e-04
##   1 8.891469e-05 8.666627e-04 1.093548e-04 6.843365e-03 1.522792e-04
##    ORIGIN
## Y             SJU          SLC          SLN          SMF          SMX
##   0 3.376823e-03 1.686979e-02 8.500964e-05 6.946304e-03 2.333145e-05
##   1 3.630172e-03 1.158242e-02 1.277510e-04 5.120260e-03 2.963823e-05
##    ORIGIN
## Y             SNA          SPI          SPN          SPS          SRQ
##   0 6.089277e-03 2.432477e-04 3.603669e-05 1.545420e-04 9.018414e-04
##   1 4.664444e-03 2.677661e-04 2.861622e-05 9.913477e-05 7.777480e-04
##    ORIGIN
## Y             STC          STL          STS          STT          STX
##   0 1.917337e-05 9.085174e-03 2.141411e-04 3.541298e-04 1.605481e-04
##   1 2.452819e-05 9.389187e-03 1.911155e-04 3.117124e-04 1.410371e-04
##    ORIGIN
## Y             SUN          SUX          SWF          SWO          SYR
##   0 1.677092e-04 1.392957e-04 2.162202e-04 1.201223e-04 1.889154e-03
##   1 1.287730e-04 2.248417e-04 3.740549e-04 8.482666e-05 1.868230e-03
##    ORIGIN
## Y             TLH          TOL          TPA          TRI          TTN
##   0 8.808200e-04 2.876005e-04 1.038758e-02 4.354434e-04 3.019228e-04
##   1 6.980314e-04 2.790082e-04 1.039893e-02 3.791649e-04 6.193368e-04
##    ORIGIN
## Y             TUL          TUS          TVC          TWF          TXK
##   0 2.428550e-03 2.693512e-03 5.696570e-04 1.998188e-04 1.690953e-04
##   1 2.102270e-03 1.910133e-03 4.507055e-04 1.195749e-04 1.338830e-04
##    ORIGIN
## Y             TYR          TYS          UIN          USA          VEL
##   0 1.965848e-04 2.229193e-03 1.138852e-04 1.235874e-04 4.781792e-05
##   1 1.635213e-04 2.066500e-03 1.604552e-04 2.013356e-04 2.963823e-05
##    ORIGIN
## Y             VLD          VPS          WRG          WYS          XNA
##   0 1.552350e-04 1.122682e-03 1.113441e-04 3.996377e-05 1.741774e-03
```

```
##   1 1.032228e-04 9.341152e-04 6.540851e-05 1.533012e-05 1.757854e-03
##    ORIGIN
## Y           YAK          YNG          YUM
##   0 1.129612e-04 0.000000e+00 2.189922e-04
##   1 5.212240e-05 1.022008e-06 8.891469e-05
##
##    DEST
## Y           ABE          ABI          ABQ          ABR          ABY
##   0 5.654989e-04 2.866765e-04 3.302902e-03 1.161952e-04 1.420677e-04
##   1 5.968526e-04 2.647001e-04 3.598490e-03 6.847453e-05 1.297950e-04
##    DEST
## Y           ACK          ACT          ACV          ACY          ADK
##   0 1.316725e-04 2.273084e-04 1.945057e-04 4.620089e-04 1.386027e-05
##   1 1.287730e-04 1.860054e-04 1.931595e-04 4.455955e-04 1.022008e-05
##    DEST
## Y           ADQ          AEX          AGS          AKN          ALB
##   0 9.540484e-05 4.938875e-04 6.188609e-04 1.016420e-05 1.656302e-03
##   1 5.314441e-05 3.914290e-04 6.990534e-04 5.110040e-06 1.834504e-03
##    DEST
## Y           ALO          AMA          ANC          APN          ART
##   0 9.355680e-05 7.422173e-04 2.742716e-03 9.332580e-05 3.003058e-06
##   1 8.891469e-05 7.562859e-04 1.847790e-03 4.803437e-05 5.110040e-06
##    DEST
## Y           ASE          ATL          ATW          AUS          AVL
##   0 8.494034e-04 5.694213e-02 5.809762e-04 8.862717e-03 1.030511e-03
##   1 1.034272e-03 4.465561e-02 5.314441e-04 8.848545e-03 1.209035e-03
##    DEST
## Y           AVP          AZA          AZO          BDL          BET
##   0 4.511517e-04 6.967094e-04 3.873945e-04 3.831209e-03 1.321345e-04
##   1 4.946518e-04 9.504674e-04 3.638348e-04 4.175924e-03 4.190232e-05
##    DEST
## Y           BFF          BFL          BGM          BGR          BHM
##   0 7.738649e-05 3.247923e-04 1.291315e-04 5.467875e-04 2.468976e-03
##   1 7.051855e-05 3.158004e-04 1.175309e-04 5.743685e-04 2.858556e-03
```

```
##    DEST
## Y             BIL          BIS          BJI          BKG          BLI
##   0 6.119308e-04 5.287692e-04 1.085721e-04 7.623147e-06 3.245612e-04
##   1 4.374194e-04 4.323093e-04 7.256256e-05 1.430811e-05 2.289298e-04
##    DEST
## Y             BLV          BMI          BNA          BOI          BOS
##   0 1.395267e-04 4.234312e-04 1.035778e-02 2.810400e-03 1.919116e-02
##   1 1.645433e-04 4.547935e-04 1.019760e-02 2.235131e-03 2.573109e-02
##    DEST
## Y             BPT          BQK          BQN          BRD          BRO
##   0 1.198913e-04 1.455328e-04 2.340075e-04 9.702187e-05 3.633700e-04
##   1 1.042448e-04 1.042448e-04 4.466175e-04 5.518843e-05 3.188665e-04
##    DEST
## Y             BRW          BTM          BTR          BTV          BUF
##   0 1.129612e-04 1.021040e-04 1.051070e-03 1.283692e-03 3.535061e-03
##   1 4.905638e-05 3.168225e-05 1.029162e-03 1.493154e-03 3.887718e-03
##    DEST
## Y             BUR          BWI          BZN          CAE          CAK
##   0 3.711548e-03 1.504624e-02 8.450143e-04 9.441152e-04 9.290999e-04
##   1 3.613820e-03 1.323602e-02 7.368677e-04 8.819928e-04 1.448185e-03
##    DEST
## Y             CDC          CDV          CGI          CHA          CHO
##   0 9.078475e-05 1.055690e-04 7.992754e-05 1.135387e-03 7.200409e-04
##   1 7.256256e-05 6.540851e-05 8.278264e-05 1.395041e-03 8.860809e-04
##    DEST
## Y             CHS          CID          CIU          CKB          CLE
##   0 3.286962e-03 1.295242e-03 9.193977e-05 1.000249e-04 6.637220e-03
##   1 3.099750e-03 1.258092e-03 7.358457e-05 1.461471e-04 6.875047e-03
##    DEST
## Y             CLL          CLT          CMH          CMI          CMX
##   0 3.328774e-04 3.313736e-02 6.327443e-03 3.095460e-04 7.969653e-05
##   1 2.953603e-04 2.733667e-02 6.480552e-03 3.485047e-04 1.594332e-04
##    DEST
## Y             CNY          COD          COS          COU          CPR
```

```
##    0 6.283321e-05 1.226634e-04 1.362695e-03 2.876005e-04 1.584691e-04
##    1 5.518843e-05 6.643051e-05 1.746612e-03 3.617908e-04 6.234248e-05
##     DEST
## Y            CRP          CRW          CSG          CVG          CWA
##    0 7.436033e-04 6.149338e-04 1.801835e-04 6.797999e-03 4.019477e-04
##    1 8.850589e-04 7.225596e-04 1.624993e-04 7.046745e-03 3.382846e-04
##     DEST
## Y            CYS          DAB          DAL          DAY          DBQ
##    0 6.468125e-06 4.957355e-04 9.759707e-03 1.882455e-03 1.293625e-04
##    1 7.154055e-06 4.885198e-04 1.003407e-02 2.085918e-03 1.522792e-04
##     DEST
## Y            DCA          DEN          DFW          DHN          DLG
##    0 1.796868e-02 3.381882e-02 3.835875e-02 1.887306e-04 1.247424e-05
##    1 1.831234e-02 2.932958e-02 3.890171e-02 2.044016e-04 7.154055e-06
##     DEST
## Y            DLH          DRO          DRT          DSM          DTW
##    0 3.825434e-04 4.317473e-04 1.801835e-05 2.123855e-03 2.308566e-02
##    1 3.924510e-04 4.108472e-04 6.132048e-06 2.236153e-03 1.713192e-02
##     DEST
## Y            DVL          EAR          EAU          ECP          EGE
##    0 8.801269e-05 3.095460e-05 9.401881e-05 7.503024e-04 2.850595e-04
##    1 7.562859e-05 1.737413e-05 1.175309e-04 7.797920e-04 3.280645e-04
##     DEST
## Y            EKO          ELM          ELP          ERI          ESC
##    0 1.048760e-04 5.174500e-05 2.311199e-03 1.499219e-04 9.055374e-05
##    1 2.963823e-05 2.963823e-05 2.674595e-03 1.073108e-04 6.029847e-05
##     DEST
## Y            EUG          EVV          EWN          EWR          EYW
##    0 5.819002e-04 6.378033e-04 2.776673e-04 1.677901e-02 6.994815e-04
##    1 5.498403e-04 6.172928e-04 3.638348e-04 3.215850e-02 4.793217e-04
##     DEST
## Y            FAI          FAR          FAT          FAY          FCA
##    0 6.680649e-04 8.119806e-04 1.602709e-03 5.204530e-04 3.608289e-04
##    1 4.006271e-04 7.123395e-04 1.435921e-03 5.580163e-04 2.565240e-04
```

```
##    DEST
## Y             FLG          FLL          FLO          FNT          FSD
##   0 1.737153e-04 1.325088e-02 2.356245e-05 5.941434e-04 8.706558e-04
##   1 1.267290e-04 1.510426e-02 3.270425e-05 4.895418e-04 9.453573e-04
##    DEST
## Y             FSM          FWA          GCC          GCK          GEG
##   0 2.732783e-04 9.411121e-04 1.185053e-04 1.071861e-04 1.787281e-03
##   1 3.321526e-04 1.155891e-03 9.811276e-05 8.176063e-05 1.507462e-03
##    DEST
## Y             GFK          GGG          GJT          GNV          GPT
##   0 3.197102e-04 1.180433e-04 5.731220e-04 6.068487e-04 5.851343e-04
##   1 1.992915e-04 9.198071e-05 3.086464e-04 6.643051e-04 6.080947e-04
##    DEST
## Y             GRB          GRI          GRK          GRR          GSO
##   0 6.687579e-04 1.434538e-04 4.215831e-04 2.380963e-03 1.731378e-03
##   1 6.499970e-04 1.441031e-04 3.597468e-04 2.503919e-03 2.050148e-03
##    DEST
## Y             GSP          GST          GTF          GTR          GUC
##   0 1.841105e-03 1.224324e-05 2.859835e-04 1.478428e-04 2.979957e-05
##   1 2.116578e-03 8.176063e-06 1.584112e-04 1.205969e-04 2.759421e-05
##    DEST
## Y             GUM          HDN          HGR          HHH          HIB
##   0 8.269959e-05 1.245114e-04 1.778734e-05 7.322841e-05 9.586685e-05
##   1 8.891469e-05 1.308170e-04 2.452819e-05 5.723244e-05 3.577028e-05
##    DEST
## Y             HLN          HNL          HOB          HOU          HPN
##   0 2.802084e-04 7.596581e-03 1.021040e-04 7.858540e-03 1.221783e-03
##   1 1.829394e-04 3.933708e-03 6.132048e-05 8.698309e-03 1.877429e-03
##    DEST
## Y             HRL          HSV          HTS          HVN          HYA
##   0 4.742521e-04 1.204457e-03 1.138852e-04 1.263594e-04 1.201223e-05
##   1 5.161140e-04 1.139539e-03 1.492132e-04 1.502352e-04 1.839614e-05
##    DEST
## Y             HYS          IAD          IAG          IAH          ICT
```

```
##   0 1.425297e-04 9.628034e-03 1.201223e-04 2.466481e-02 1.456021e-03
##   1 1.645433e-04 8.984472e-03 9.095870e-05 2.211932e-02 1.447163e-03
##    DEST
## Y             IDA          IFP          ILM          IMT          IND
##   0 3.314914e-04 7.161138e-06 8.819750e-04 9.609785e-05 6.791762e-03
##   1 2.064456e-04 7.154055e-06 8.881249e-04 4.905638e-05 6.811683e-03
##    DEST
## Y             INL          ISN          ISP          ITH          ITO
##   0 1.011799e-04 2.014359e-04 7.904972e-04 1.457638e-04 1.041137e-03
##   1 5.212240e-05 1.481911e-04 1.034272e-03 1.083328e-04 3.055804e-04
##    DEST
## Y             JAC          JAN          JAX          JFK          JLN
##   0 5.851343e-04 1.086414e-03 4.252330e-03 1.710958e-02 1.044140e-04
##   1 5.621044e-04 9.974797e-04 4.708390e-03 2.043914e-02 1.185529e-04
##    DEST
## Y             JMS          JNU          KOA          KTN          LAN
##   0 1.321345e-04 6.752260e-04 2.352318e-03 3.622150e-04 4.968906e-04
##   1 9.811276e-05 3.914290e-04 9.586434e-04 2.391499e-04 4.047151e-04
##    DEST
## Y             LAR          LAS          LAW          LAX          LBB
##   0 8.593365e-05 2.286367e-02 1.824935e-04 3.136116e-02 8.630326e-04
##   1 8.176063e-05 2.121586e-02 1.747634e-04 2.918344e-02 9.729515e-04
##    DEST
## Y             LBE          LBF          LBL          LCH          LCK
##   0 1.487669e-04 8.708868e-05 7.715549e-05 2.827494e-04 1.526939e-04
##   1 1.430811e-04 6.540851e-05 7.051855e-05 2.432379e-04 1.716973e-04
##    DEST
## Y             LEX          LFT          LGA          LGB          LIH
##   0 1.391802e-03 6.465814e-04 2.152037e-02 2.324829e-03 2.230810e-03
##   1 1.429789e-03 5.590383e-04 2.997243e-02 2.467127e-03 9.177631e-04
##    DEST
## Y             LIT          LNK          LRD          LSE          LWB
##   0 1.845726e-03 2.561839e-04 3.393455e-04 2.966097e-04 7.322841e-05
##   1 1.803844e-03 2.084896e-04 3.178445e-04 2.820742e-04 1.042448e-04
```

```
##    DEST
## Y             LWS          LYH          MAF          MBS          MCI
##   0 1.145782e-04 1.289005e-04 1.201685e-03 3.931696e-04 7.944012e-03
##   1 2.963823e-05 2.146217e-04 1.294884e-03 3.852970e-04 7.035503e-03
##    DEST
## Y             MCO          MDT          MDW          MEI          MEM
##   0 1.869912e-02 8.720418e-04 1.208407e-02 1.420677e-04 3.157831e-03
##   1 2.147750e-02 8.319144e-04 1.137495e-02 1.819174e-04 3.415550e-03
##    DEST
## Y             MFE          MFR          MGM          MHK          MHT
##   0 6.017666e-04 6.588247e-04 5.518696e-04 2.386276e-04 1.320652e-03
##   1 7.470878e-04 6.459090e-04 6.346669e-04 2.963823e-04 1.359271e-03
##    DEST
## Y             MIA          MKE          MKG          MLB          MLI
##   0 1.233541e-02 4.535310e-03 9.886990e-05 3.555158e-04 6.994815e-04
##   1 1.190128e-02 4.486615e-03 1.113989e-04 2.994483e-04 7.000754e-04
##    DEST
## Y             MLU          MMH          MOB          MOT          MQT
##   0 4.257412e-04 1.570830e-05 8.824370e-04 3.786163e-04 1.725603e-04
##   1 4.180012e-04 2.350618e-05 9.688635e-04 2.555020e-04 2.146217e-04
##    DEST
## Y             MRY          MSN          MSO          MSP          MSY
##   0 5.345443e-04 1.844108e-03 4.927325e-04 2.340699e-02 7.772838e-03
##   1 5.539283e-04 1.744568e-03 3.944951e-04 1.704505e-02 7.889901e-03
##    DEST
## Y             MTJ          MVY          MYR          OAJ          OAK
##   0 2.074420e-04 6.814631e-05 1.632277e-03 3.940936e-04 7.519426e-03
##   1 1.911155e-04 5.518843e-05 1.350072e-03 5.150920e-04 6.653272e-03
##    DEST
## Y             OGD          OGG          OGS          OKC          OMA
##   0 1.755634e-05 3.937471e-03 1.848036e-05 3.184165e-03 3.488629e-03
##   1 2.248417e-05 2.010290e-03 8.176063e-06 3.484025e-03 3.435991e-03
##    DEST
## Y             OME          ONT          ORD          ORF          ORH
```

```
##    0 1.083411e-04 2.994511e-03 4.498881e-02 3.006985e-03 1.148092e-04
##    1 4.496835e-05 3.026165e-03 4.975237e-02 3.503443e-03 2.074676e-04
##     DEST
## Y            OTH          OTZ          OWB          PAH          PBG
##    0 4.111879e-05 1.113441e-04 1.270524e-05 8.847470e-05 1.432228e-04
##    1 9.504674e-05 3.577028e-05 2.146217e-05 1.287730e-04 1.062888e-04
##     DEST
## Y            PBI          PDX          PGD          PGV          PHF
##    0 3.275181e-03 9.182427e-03 6.907033e-04 1.018730e-04 3.425796e-04
##    1 4.344556e-03 7.333929e-03 9.576214e-04 2.033796e-04 4.108472e-04
##     DEST
## Y            PHL          PHX          PIA          PIB          PIE
##    0 1.562791e-02 2.434764e-02 7.329771e-04 9.170877e-05 9.940121e-04
##    1 1.779725e-02 2.392010e-02 7.797920e-04 1.400151e-04 1.267290e-03
##     DEST
## Y            PIH          PIT          PLN          PNS          PPG
##    0 1.975088e-04 6.785987e-03 1.501529e-04 1.617262e-03 1.455328e-05
##    1 8.789268e-05 6.672690e-03 8.278264e-05 1.407305e-03 2.350618e-05
##     DEST
## Y            PRC          PSC          PSE          PSG          PSM
##    0 3.557468e-05 4.051818e-04 9.471182e-05 1.127302e-04 4.204281e-05
##    1 2.555020e-05 3.842750e-04 1.890715e-04 7.869461e-05 4.292433e-05
##     DEST
## Y            PSP          PUB          PVD          PVU          PWM
##    0 1.505918e-03 1.365236e-04 2.491845e-03 7.507645e-05 1.577991e-03
##    1 1.331676e-03 1.022008e-04 2.927031e-03 6.029847e-05 1.871296e-03
##     DEST
## Y            RAP          RDD          RDM          RDU          RFD
##    0 5.946054e-04 1.580070e-04 5.257661e-04 8.163235e-03 1.014110e-04
##    1 5.815225e-04 1.604552e-04 4.752337e-04 9.122443e-03 6.438650e-05
##     DEST
## Y            RHI          RIC          RKS          RNO          ROA
##    0 1.111131e-04 3.054803e-03 9.886990e-05 2.539663e-03 3.400385e-04
##    1 4.394634e-05 3.820266e-03 7.971662e-05 2.657221e-03 3.566808e-04
```

```
##    DEST
## Y             ROC          ROW          RST          RSW          SAF
##   0 1.934662e-03 2.037459e-04 4.275892e-04 4.482179e-03 2.012049e-04
##   1 2.062412e-03 1.471691e-04 5.273561e-04 4.612322e-03 1.819174e-04
##    DEST
## Y             SAN          SAT          SAV          SBA          SBN
##   0 1.298222e-02 5.498137e-03 2.177448e-03 9.884680e-04 9.254038e-04
##   1 1.155278e-02 5.812159e-03 2.183009e-03 9.596654e-04 1.083328e-03
##    DEST
## Y             SBP          SCC          SCE          SCK          SDF
##   0 6.398823e-04 1.492289e-04 1.942747e-04 8.939872e-05 3.126645e-03
##   1 7.399337e-04 6.234248e-05 1.778294e-04 1.604552e-04 3.375692e-03
##    DEST
## Y             SEA          SFB          SFO          SGF          SGU
##   0 1.958687e-02 1.178123e-03 2.260494e-02 1.183436e-03 4.950425e-04
##   1 1.985455e-02 2.147239e-03 3.240481e-02 1.201881e-03 2.902502e-04
##    DEST
## Y             SHD          SHV          SIT          SJC          SJT
##   0 8.801269e-05 9.295619e-04 1.963538e-04 7.837057e-03 2.088280e-04
##   1 1.073108e-04 8.850589e-04 1.481911e-04 6.746274e-03 1.563672e-04
##    DEST
## Y             SJU          SLC          SLN          SMF          SMX
##   0 3.142584e-03 1.696820e-02 9.286379e-05 6.747871e-03 2.356245e-05
##   1 4.712479e-03 1.147102e-02 1.165089e-04 5.950130e-03 2.044016e-05
##    DEST
## Y             SNA          SPI          SPN          SPS          SRQ
##   0 6.079113e-03 2.497158e-04 3.557468e-05 1.501529e-04 9.193977e-04
##   1 4.746205e-03 2.810522e-04 2.555020e-05 9.811276e-05 7.757040e-04
##    DEST
## Y             STC          STL          STS          STT          STX
##   0 2.286944e-05 9.380860e-03 2.092900e-04 3.488167e-04 1.543110e-04
##   1 1.533012e-05 8.511282e-03 1.992915e-04 3.638348e-04 1.778294e-04
##    DEST
## Y             SUN          SUX          SWF          SWO          SYR
```

```
##    0 1.734843e-04 1.483049e-04 2.289254e-04 1.157332e-04 1.826321e-03
##    1 1.236630e-04 2.105336e-04 3.219325e-04 8.278264e-05 2.172789e-03
##     DEST
## Y             TLH          TOL          TPA          TRI          TTN
##    0 8.561025e-04 2.774363e-04 1.022472e-02 4.146530e-04 2.986887e-04
##    1 7.757040e-04 3.198885e-04 1.111740e-02 4.588816e-04 5.753905e-04
##     DEST
## Y             TUL          TUS          TVC          TWF          TXK
##    0 2.304731e-03 2.520259e-03 5.477115e-04 1.831865e-04 1.785664e-04
##    1 2.520272e-03 2.678683e-03 4.926078e-04 7.358457e-05 1.308170e-04
##     DEST
## Y             TYR          TYS          UIN          USA          VEL
##    0 1.928887e-04 2.146031e-03 1.145782e-04 1.335206e-04 4.874194e-05
##    1 1.819174e-04 2.540712e-03 1.502352e-04 1.553452e-04 3.474827e-05
##     DEST
## Y             VLD          VPS          WRG          WYS          XNA
##    0 1.540800e-04 1.093113e-03 1.069551e-04 3.580569e-05 1.714284e-03
##    1 9.709075e-05 1.048580e-03 8.891469e-05 2.044016e-05 1.883561e-03
##     DEST
## Y             YAK          YNG          YUM
##    0 1.104201e-04 2.310044e-07 2.129861e-04
##    1 5.621044e-05 1.022008e-06 1.124209e-04
##
##     DIVERTED
## Y    0 1
##    0 1 0
##    1 1 0
##
##     month
## Y         April     August   December   February    January       July
##    0 0.08456888 0.08513345 0.08338106 0.07212028 0.07922690 0.08548366
##    1 0.07667410 0.10649016 0.08013973 0.07195549 0.07227129 0.10590046
##     month
## Y          June      March        May   November    October  September
```

```
##   0 0.08382019 0.08639751 0.08537485 0.08166053 0.08888081 0.08395187
##   1 0.09906527 0.07269031 0.08766171 0.08242800 0.07460045 0.07012303
##
##    day
## Y       Friday    Monday  Saturday    Sunday  Thursday   Tuesday Wednesday
##   0 0.1454081 0.1489022 0.1267447 0.1436591 0.1452403 0.1437268 0.1463187
##   1 0.1665607 0.1581997 0.1025145 0.1359087 0.1603765 0.1394530 0.1369869
##
##    dep_slot
## Y        Redeye    Morning  Afternoon    Evening
##   0 0.05639627 0.39635581 0.28737508 0.25987284
##   1 0.02758502 0.22890116 0.30727894 0.43623488
##
##    arr_slot
## Y        Redeye    Morning  Afternoon    Evening
##   0 0.02589444 0.31483226 0.30297896 0.35629433
##   1 0.09452551 0.13901454 0.24971230 0.51674764
```

#Pivot tables of classifications on train data

```
## [1] "#Contingency table for delayed % by destination"
## [1] "The Worst 10 destination with highest delay probability"
##    Destination Delay_Probability
## 1          YNG              0.50
## 2          OTH              0.34
## 3          CMX              0.31
## 4          PGV              0.31
## 5          PSE              0.31
## 6          BKG              0.30
## 7          BQN              0.30
## 8          EWR              0.30
## 9          TTN              0.30
## 10         ORH              0.29
## [1] "Top 10 origin with lowest delay probability"
##    Destination Ontime_Probability
```

```
## 1          EKO               0.94
## 2          ITO               0.94
## 3          LWS               0.94
## 4          BET               0.93
## 5          BTM               0.93
## 6          DRT               0.93
## 7          OTZ               0.93
## 8          CPR               0.92
## 9          HIB               0.92
## 10         KOA               0.92
## [1] "The Worst 10 origin with highest delay probability"
##    Origin Delay_Probability
## 1     YNG              1.00
## 2     SCK              0.38
## 3     OTH              0.37
## 4     BKG              0.34
## 5     MMH              0.33
## 6     ORH              0.32
## 7     OWB              0.32
## 8     PPG              0.32
## 9     TTN              0.32
## 10    HGR              0.31
## [1] "Top 10 origin with lowest delay probability"
##    Origin Ontime_Probability
## 1     LBF               0.95
## 2     EKO               0.94
## 3     ITO               0.94
## 4     LWS               0.94
## 5     BTM               0.93
## 6     EAR               0.93
## 7     HIB               0.93
## 8     CPR               0.92
## 9     FAI               0.92
## 10    GST               0.92
```

```
## [1] "#Contingency Probability Table by Month"
##
##              Non-delayed Delayed
##   April             0.83    0.17
##   August            0.78    0.22
##   December          0.82    0.18
##   February          0.82    0.18
##   January           0.83    0.17
##   July              0.78    0.22
##   June              0.79    0.21
##   March             0.84    0.16
##   May               0.81    0.19
##   November          0.81    0.19
##   October           0.84    0.16
##   September         0.84    0.16
## [1] "#Contingency Probability Table by Day"
##
##              Non-delayed Delayed
##   Friday            0.79    0.21
##   Monday            0.81    0.19
##   Saturday          0.85    0.15
##   Sunday            0.82    0.18
##   Thursday          0.80    0.20
##   Tuesday           0.82    0.18
##   Wednesday         0.83    0.17
```

Lets take a look at the naive Bayes classification model performance with the test data

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction       0       1
##          0 1396704  288457
##          1   45589   38269
##
```

```
##                Accuracy : 0.8112
##                  95% CI : (0.8106, 0.8117)
##     No Information Rate : 0.8153
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.12
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9684
##             Specificity : 0.1171
##          Pos Pred Value : 0.8288
##          Neg Pred Value : 0.4564
##              Prevalence : 0.8153
##          Detection Rate : 0.7895
##    Detection Prevalence : 0.9526
##       Balanced Accuracy : 0.5428
##
##        'Positive' Class : 0
##
```

Lets assume that we are flyign with American Airlines (AA),from ABQ to JFK,no diversion, on a Wednesday, October 28, 2020, scheduled to be at 1:00AM (Redeye) and arriving at 6:30AM (Morning)

```
##              0          1
## [1,] 0.9591261 0.04087387
## [1] 0
## Levels: 0 1
## [1] 0
## Levels: 0 1
```