# Traffic Flow Analysis

1. Feature Engineering:

   Code:

```
#library imports
library(dplyr)
library(tidyr) #used for separate() function
library(lattice) #used for correlation heatmap
library(reshape2) #used for melt function in heatmap
library(ggplot2)

#import dataset of one month traffic data
data <- read.csv("oneMonthDataset.csv")
#add id column with serial number
data$id <- seq.int(nrow(data))

#summary of orignal dataset
summary(data)

#feature engineering
numericReplacedData <- data
dataColNames <- colnames(numericReplacedData)
#repace low, normal, high, heavy with numeric 1, 2, 3 and 4 for easier
analysis
numericReplacedData %>% distinct(Traffic.Situation)
numericReplacedData %>% count(Traffic.Situation)
numericReplacedData$Traffic.Situation <- c(low = 1, normal = 2, high = 3,
heavy = 4)[numericReplacedData$Traffic.Situation]
#replace days of week sunday to saturady with numeric values 1 to 7
numericReplacedData %>% distinct(Day.of.the.week)
numericReplacedData %>% count(Day.of.the.week)
numericReplacedData$Day.of.the.week <- c(Sunday = 1, Monday = 2, Tuesday
= 3, Wednesday = 4, Thursday = 5, Friday = 6, Saturday =
7)[numericReplacedData$Day.of.the.week]
#separate time into hour minute and seconds (AM/PM is attachd with
seconds in Part3 column)
numericReplacedData <- numericReplacedData %>% separate(Time, into =
c('Hour', 'Minute', 'Part3'), sep = ':')
#separate seconds and AM/PM
numericReplacedData <- numericReplacedData %>% separate(Part3, into =
c('Seconds', 'Part.Of.Day'), sep = ' ')
#replace Am and PM with numeric 0 nd 1 for easier analysis
numericReplacedData %>% distinct(Part.Of.Day)
numericReplacedData %>% count(Part.Of.Day)
```

```
numericReplacedData$Part.Of.Day <- c(AM = 0, PM =
1)[numericReplacedData$Part.Of.Day]
#check for distinct vales in newly created columns
numericReplacedData %>% distinct(Hour)
numericReplacedData %>% distinct(Minute)
numericReplacedData %>% distinct(Seconds)
numericReplacedData %>% distinct(Part.Of.Day)
#remove 'Seconds' column as there is only 1 distinct values
numericReplacedData$Seconds <- NULL
#convert 'hour' and 'minute' into numeric data type
numericReplacedData$Hour <- as.integer(numericReplacedData$Hour)
numericReplacedData$Minute <- as.integer(numericReplacedData$Minute)
#summary of modified dataset (after factoring all columns into numeric
datatype)
summary(numericReplacedData)

#correlation and heat map for upadted dataset (numericReplacedData)
#find correlation between all numeric columns
corDataSet <- subset(numericReplacedData, select = -c( Date,
Day.of.the.week, id))
correlationMatrix = round(cor(corDataSet), 3)
correlationMatrix
#plot heat map for the correlation matrix
meltedCorrelationMatrix <- melt(correlationMatrix)
meltedCorrelationMatrix
ggplot(data = meltedCorrelationMatrix, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  geom_text(aes(Var2, Var1, label = value), color = "black", size = 4)

par(mfrow = c(2, 2))
hist(numericReplacedData$CarCount , main = "Car Count", xlab =
"Weight",ylim = c(0,1000),col = "yellow",border = "blue")
hist(numericReplacedData$BikeCount , main = "Bike Count", xlab =
"Weight",ylim = c(0,1000),col = "yellow",border = "blue")
hist(numericReplacedData$TruckCount , main = "Truck Count", xlab =
"Weight",ylim = c(0,1000),col = "yellow",border = "blue")
hist(numericReplacedData$BusCount , main = "Bus Count", xlab =
"Weight",ylim = c(0,1000),col = "yellow",border = "blue")

write.csv(numericReplacedData, "numericDataset.csv", row.names = FALSE)
```

Output:

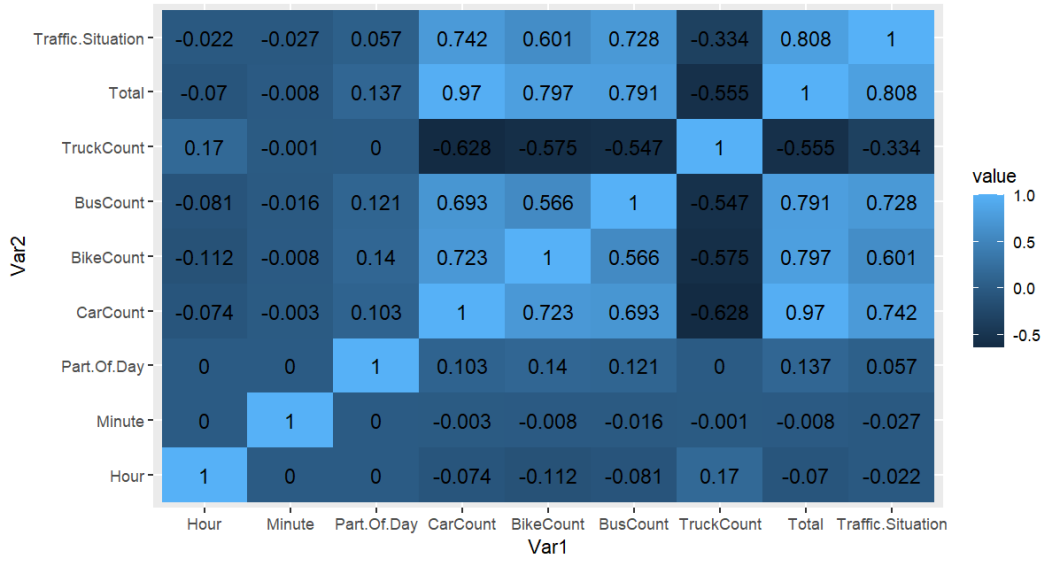
```
> summary(data)
     Time                Date         Day.of.the.week       CarCount        BikeCount      
 Length:2976        Min.   :  1.00   Length:2976        Min.   :  6.0   Min.   : 0.00  
 Class :character   1st Qu.:  8.00   Class :character   1st Qu.: 19.0   1st Qu.: 5.00  
 Mode  :character   Median : 16.00   Mode  :character   Median : 64.0   Median :12.00  
                    Mean   : 16.00                      Mean   : 68.7   Mean   :14.92  
                    3rd Qu.: 24.00                      3rd Qu.:107.0   3rd Qu.:22.00  
                    Max.   : 31.00                      Max.   :180.0   Max.   :70.00  
    BusCount        TruckCount        Total        Traffic.Situation        id       
 Min.   : 0.00   Min.   : 0.00   Min.   : 21.0   Length:2976        Min.   :   1.0  
 1st Qu.: 1.00   1st Qu.: 6.00   1st Qu.: 55.0   Class :character   1st Qu.: 744.8  
 Median :12.00   Median :14.00   Median :109.0   Mode  :character   Median :1488.5  
 Mean   :15.28   Mean   :15.32   Mean   :114.2                      Mean   :1488.5  
 3rd Qu.:25.00   3rd Qu.:23.00   3rd Qu.:164.0                      3rd Qu.:2232.2  
 Max.   :50.00   Max.   :40.00   Max.   :279.0                      Max.   :2976.0  
>
```

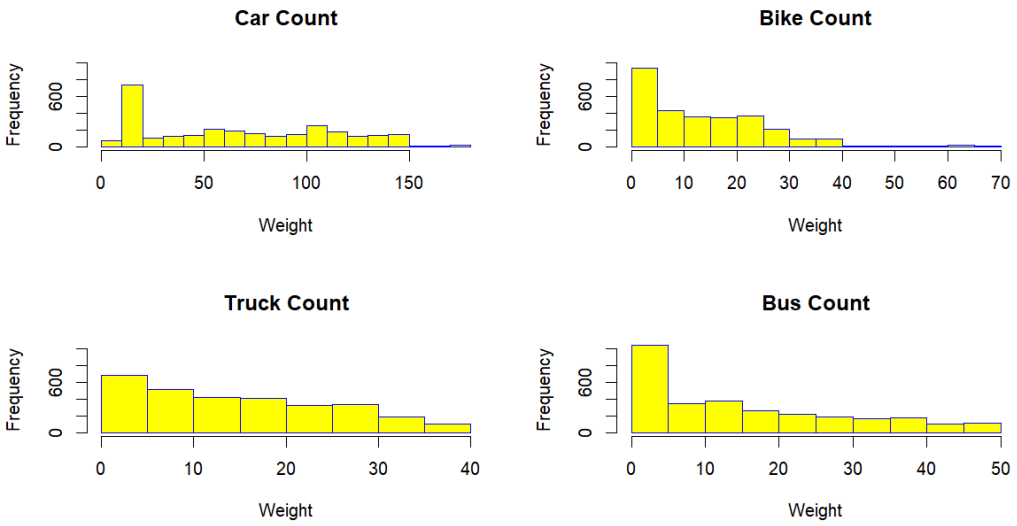
```
> numericReplacedData %>% distinct(Traffic.Situation)
  Traffic.Situation
1             heavy
2               low
3            normal
4              high
> numericReplacedData %>% count(Traffic.Situation)
  Traffic.Situation    n
1             heavy  689
2              high  321
3               low  298
4            normal 1668
```

```
> numericReplacedData %>% distinct(Part.Of.Day)
  Part.Of.Day
1          AM
2          PM
> numericReplacedData %>% count(Part.Of.Day)
  Part.Of.Day    n
1          AM 1488
2          PM 1488
```

```
> numericReplacedData %>% distinct(Day.of.the.week)
  Day.of.the.week
1         Tuesday
2       Wednesday
3        Thursday
4          Friday
5        Saturday
6          Sunday
7          Monday
> numericReplacedData %>% count(Day.of.the.week)
  Day.of.the.week   n
1          Friday 384
2          Monday 384
3        Saturday 384
4          Sunday 384
5        Thursday 480
6         Tuesday 480
7       Wednesday 480
```

```
> summary(numericReplacedData)
      Hour           Minute       Part.Of.Day       Date        Day.of.the.week    CarCount     
 Min.   : 1.00   Min.   : 0.00   Min.   :0.0   Min.   :  1.00   Min.   :1       Min.   :  6.0  
 1st Qu.: 3.75   1st Qu.:11.25   1st Qu.:0.0   1st Qu.:  8.00   1st Qu.:2       1st Qu.: 19.0  
 Median : 6.50   Median :22.50   Median :0.5   Median : 16.00   Median :4       Median : 64.0  
 Mean   : 6.50   Mean   :22.50   Mean   :0.5   Mean   : 16.00   Mean   :4       Mean   : 68.7  
 3rd Qu.: 9.25   3rd Qu.:33.75   3rd Qu.:1.0   3rd Qu.: 24.00   3rd Qu.:6       3rd Qu.:107.0  
 Max.   :12.00   Max.   :45.00   Max.   :1.0   Max.   : 31.00   Max.   :7       Max.   :180.0  
   BikeCount        BusCount        TruckCount        Total        Traffic.Situation
 Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 21.0   Min.   :1.000    
 1st Qu.: 5.00   1st Qu.: 1.00   1st Qu.: 6.00   1st Qu.: 55.0   1st Qu.:2.000    
 Median :12.00   Median :12.00   Median :14.00   Median :109.0   Median :2.000    
 Mean   :14.92   Mean   :15.28   Mean   :15.32   Mean   :114.2   Mean   :2.471    
 3rd Qu.:22.00   3rd Qu.:25.00   3rd Qu.:23.00   3rd Qu.:164.0   3rd Qu.:3.000    
 Max.   :70.00   Max.   :50.00   Max.   :40.00   Max.   :279.0   Max.   :4.000    
       id        
 Min.   :   1.0  
 1st Qu.: 744.8  
 Median :1488.5  
 Mean   :1488.5  
 3rd Qu.:2232.2  
 Max.   :2976.0  
>
```



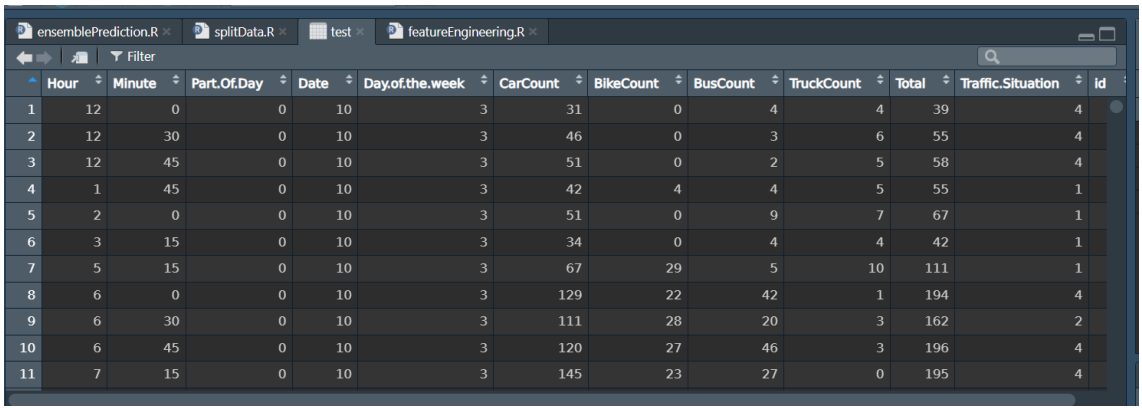

2. Split data:

   Code:

```
#library imports
library(lattice) #used for correlation heatmap
library(reshape2) #used for melt funtion in heatmap
library(ggplot2)

#import dataset of one month traffic data
data <- read.csv("numericDataset.csv")

#summary of orignal dataset
summary(data)

#splitting training and testing data
```

```
train <- data %>% dplyr::sample_frac(0.70)
write.csv(train, "train.csv", row.names = FALSE)
test  <- dplyr::anti_join(data, train, by = 'id')
write.csv(test, "test.csv", row.names = FALSE)

empty <- data.frame(model = c(""), accuracy = c(""))
write.csv(empty, "Accuracy.csv", row.names = FALSE)

#xtest <- subset(test, select = -c(Traffic.Situation))
#ytest <- subset(test, select = c(Traffic.Situation))
#xtrain <- subset(train, select = -c(Traffic.Situation))
#ytrain <- subset(train, select = c(Traffic.Situation))
```

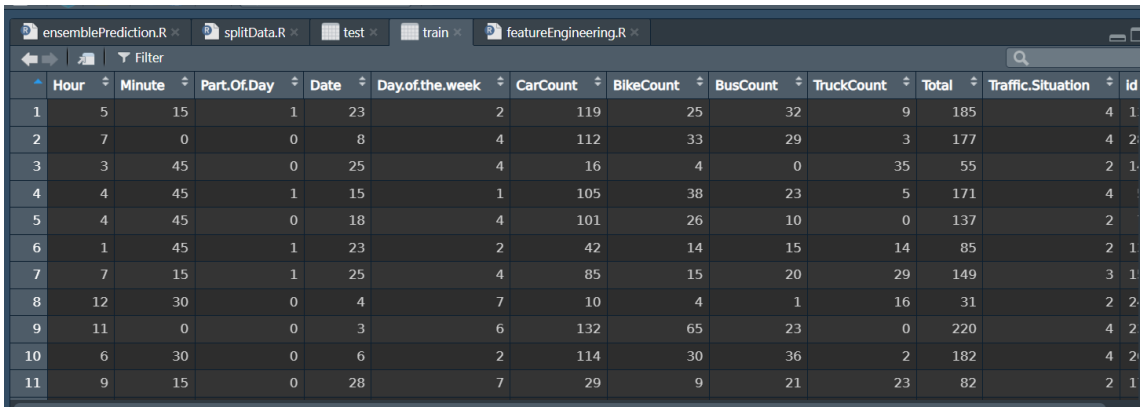
Output:



ensemblePrediction.R | splitData.R | test | featureEngineering.R

Filter

| | Hour | Minute | Part.Of.Day | Date | Day.of.the.week | CarCount | BikeCount | BusCount | TruckCount | Total | Traffic.Situation | id |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 12 | 0 | 0 | 10 | 3 | 31 | 0 | 4 | 4 | 39 | 4 | |
| 2 | 12 | 30 | 0 | 10 | 3 | 46 | 0 | 3 | 6 | 55 | 4 | |
| 3 | 12 | 45 | 0 | 10 | 3 | 51 | 0 | 2 | 5 | 58 | 4 | |
| 4 | 1 | 45 | 0 | 10 | 3 | 42 | 4 | 4 | 5 | 55 | 1 | |
| 5 | 2 | 0 | 0 | 10 | 3 | 51 | 0 | 9 | 7 | 67 | 1 | |
| 6 | 3 | 15 | 0 | 10 | 3 | 34 | 0 | 4 | 4 | 42 | 1 | |
| 7 | 5 | 15 | 0 | 10 | 3 | 67 | 29 | 5 | 10 | 111 | 1 | |
| 8 | 6 | 0 | 0 | 10 | 3 | 129 | 22 | 42 | 1 | 194 | 4 | |
| 9 | 6 | 30 | 0 | 10 | 3 | 111 | 28 | 20 | 3 | 162 | 2 | |
| 10 | 6 | 45 | 0 | 10 | 3 | 120 | 27 | 46 | 3 | 196 | 4 | |
| 11 | 7 | 15 | 0 | 10 | 3 | 145 | 23 | 27 | 0 | 195 | 4 | |



ensemblePrediction.R | splitData.R | test | train | featureEngineering.R

Filter

| | Hour | Minute | Part.Of.Day | Date | Day.of.the.week | CarCount | BikeCount | BusCount | TruckCount | Total | Traffic.Situation | id |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 5 | 15 | 1 | 23 | 2 | 119 | 25 | 32 | 9 | 185 | 4 | 1 |
| 2 | 7 | 0 | 0 | 8 | 4 | 112 | 33 | 29 | 3 | 177 | 4 | 2 |
| 3 | 3 | 45 | 0 | 25 | 4 | 16 | 4 | 0 | 35 | 55 | 2 | 1 |
| 4 | 4 | 45 | 1 | 15 | 1 | 105 | 38 | 23 | 5 | 171 | 4 | |
| 5 | 4 | 45 | 0 | 18 | 4 | 101 | 26 | 10 | 0 | 137 | 2 | |
| 6 | 1 | 45 | 1 | 23 | 2 | 42 | 14 | 15 | 14 | 85 | 2 | 1 |
| 7 | 7 | 15 | 1 | 25 | 4 | 85 | 15 | 20 | 29 | 149 | 3 | 1 |
| 8 | 12 | 30 | 0 | 4 | 7 | 10 | 4 | 1 | 16 | 31 | 2 | 2 |
| 9 | 11 | 0 | 0 | 3 | 6 | 132 | 65 | 23 | 0 | 220 | 4 | 2 |
| 10 | 6 | 30 | 0 | 6 | 2 | 114 | 30 | 36 | 2 | 182 | 4 | 2 |
| 11 | 9 | 15 | 0 | 28 | 7 | 29 | 9 | 21 | 23 | 82 | 2 | 1 |

3. knnModel:

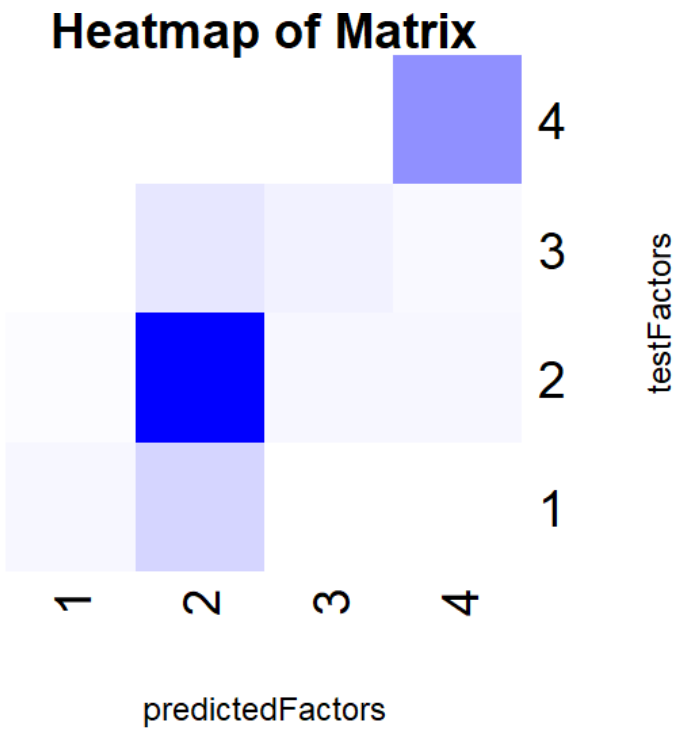
Code:

```
#library imports
library(ggplot2)
library(caret) #confusion matrix
library(gmodels)
library(class)
```

```
#import dataset of one month traffic data
numericData <- read.csv("numericDataset.csv")
testData <- read.csv("test.csv")
trainData <- read.csv("train.csv")

#summary of test and training dataset
summary(testData)
summary(trainData)

predictedTrafficSituation <- knn(train = trainData, test = testData, cl =
trainData$Traffic.Situation, k=11)
actualTrafficSituation <- factor(testData$Traffic.Situation, levels = c("1", "2",
"3", "4"))
conf_matrix <- confusionMatrix(table(actualTrafficSituation,
predictedTrafficSituation))
conf_matrix
conf_matrix$overall['Accuracy']
accuracy <- read.csv("Accuracy.csv", header = TRUE)
accuracy <- rbind(accuracy, c("knn", conf_matrix$overall['Accuracy']))
write.csv(accuracy, "Accuracy.csv", row.names = FALSE)
heatmap(conf_matrix$table,
        Rowv = NA,
        Colv = NA,
        col = colorRampPalette(c("white", "blue"))(100), # Choose a color
gradient
        scale = "none",
        xlab = "predictedFactors",
        ylab = "testFactors",
        main = "Heatmap of Matrix",
        labRow = c(1,2,3,4),
        labCol = c(1,2,3,4))
predictedData = data.frame(testData$Traffic.Situation,
predictedTrafficSituation)
write.csv(predictedData, "knnPredictions.csv")
```

Output:

```
> conf_matrix
Confusion Matrix and Statistics

                     predictedTrafficSituation
actualTrafficSituation   1   2   3   4
                     1  16  77   0   1
                     2   9 466  18  15
                     3   0  40  29  14
                     4   3   1   4 200

Overall Statistics

               Accuracy : 0.7962
                 95% CI : (0.7683, 0.8222)
    No Information Rate : 0.654
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6357

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity           0.57143   0.7979  0.56863   0.8696
Specificity           0.90983   0.8641  0.93587   0.9879
Pos Pred Value        0.17021   0.9173  0.34940   0.9615
Neg Pred Value        0.98498   0.6935  0.97284   0.9562
Prevalence            0.03135   0.6540  0.05711   0.2576
Detection Rate        0.01792   0.5218  0.03247   0.2240
Detection Prevalence  0.10526   0.5689  0.09295   0.2329
Balanced Accuracy     0.74063   0.8310  0.75225   0.9287
> conf_matrix$overall['Accuracy']
 Accuracy
0.7961926
```




4. rfModel:

   Code:

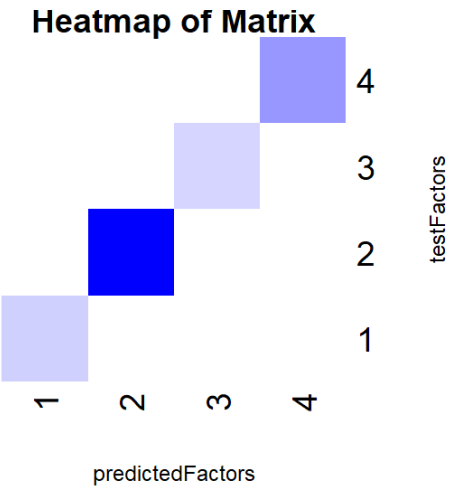
```
#library imports
library(ggplot2)
library(randomForest)
library(caret) #confusion matrix

#import dataset of one month traffic data
numericData <- read.csv("numericDataset.csv")
testData <- read.csv("test.csv")
trainData <- read.csv("train.csv")

#summary of test and training dataset
summary(testData)
summary(trainData)

#random forest
rfModel <- randomForest(Traffic.Situation ~ ., data = trainData)
importance(rfModel)
varImpPlot(rfModel)
predictedTrafficSituation <- predict(rfModel, testData, type= "response")
predictedTrafficSituation <- round(predictedTrafficSituation)
predictedTrafficSituation <- factor(predictedTrafficSituation, levels = c("1", "2",
"3", "4"))
actualTrafficSituation <- factor(testData$Traffic.Situation, levels = c("1", "2",
"3", "4"))
```

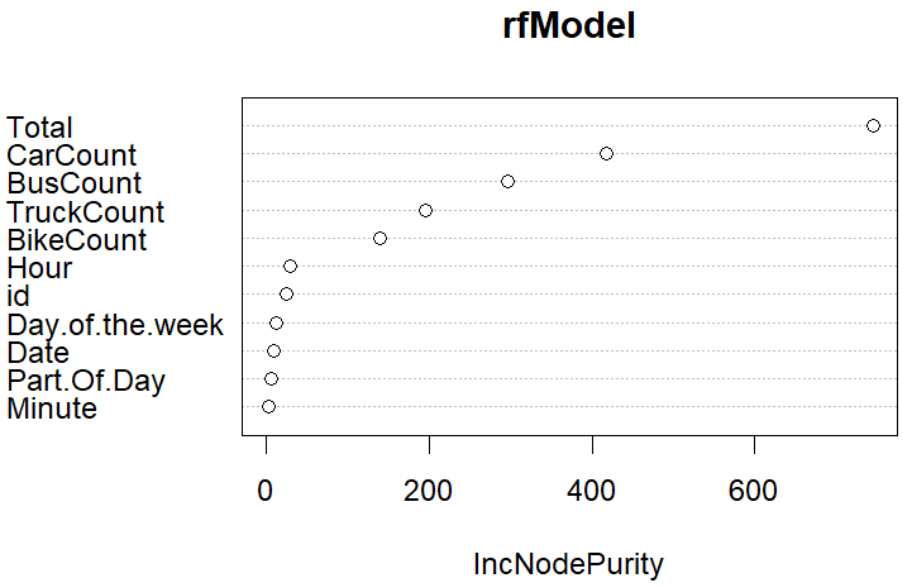
```
conf_matrix <- confusionMatrix(table(actualTrafficSituation,
predictedTrafficSituation))
conf_matrix
conf_matrix$overall['Accuracy']
accuracy <- read.csv("Accuracy.csv", header = TRUE)
accuracy <- rbind(accuracy, c("rf", conf_matrix$overall['Accuracy']))
write.csv(accuracy, "Accuracy.csv", row.names = FALSE)
heatmap(conf_matrix$table,
        Rowv = NA,
        Colv = NA,
        col = colorRampPalette(c("white", "blue"))(100), # Choose a color
gradient
        scale = "none",
        xlab = "predictedFactors",
        ylab = "testFactors",
        main = "Heatmap of Matrix",
        labRow = c(1,2,3,4),
        labCol = c(1,2,3,4))
predictedData = data.frame(testData$Traffic.Situation,
predictedTrafficSituation)
write.csv(predictedData, "rfPredictions.csv")
```

Output:

```
Confusion Matrix and Statistics

                     predictedTrafficSituation
actualTrafficSituation   1   2   3   4
                     1  91   2   1   0
                     2   0 505   2   1
                     3   0   1  82   0
                     4   0   0   3 205

Overall Statistics

               Accuracy : 0.9888
                 95% CI : (0.9795, 0.9946)
    No Information Rate : 0.5689
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9814

 Mcnemar's Test P-Value : NA
```

```
Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity            1.0000   0.9941  0.93182   0.9951
Specificity            0.9963   0.9922  0.99876   0.9956
Pos Pred Value         0.9681   0.9941  0.98795   0.9856
Neg Pred Value         1.0000   0.9922  0.99259   0.9985
Prevalence             0.1019   0.5689  0.09854   0.2307
Detection Rate         0.1019   0.5655  0.09183   0.2296
Detection Prevalence   0.1053   0.5689  0.09295   0.2329
Balanced Accuracy      0.9981   0.9932  0.96529   0.9954
> conf_matrix$overall['Accuracy']
 Accuracy
0.9888018
```






5. svmModle:

Code:

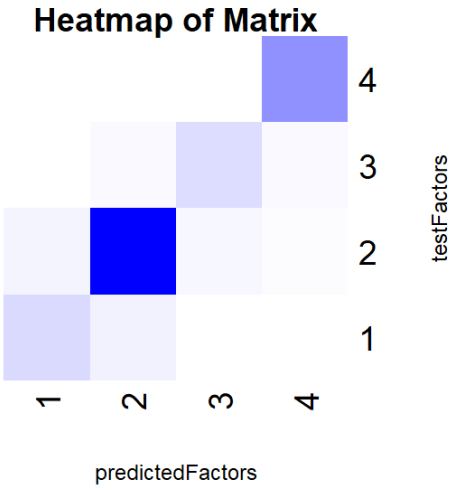
```
#library imports
library(ggplot2)
library(e1071)
library(caret) #confusion matrix

#import dataset of one month traffic data
numericData <- read.csv("numericDataset.csv")
testData <- read.csv("test.csv")
trainData <- read.csv("train.csv")

classifier = svm(formula = Traffic.Situation ~ ., data = trainData, type = 'C-
classification', kernel = 'linear')
predictedFactors = predict(classifier, newdata = testData)
testFactors <- factor(testData$Traffic.Situation, levels = c("1", "2", "3", "4"))
conf_matrix <- confusionMatrix(table(testFactors, predictedFactors))
conf_matrix
conf_matrix$overall['Accuracy']
accuracy <- read.csv("Accuracy.csv", header = TRUE)
accuracy <- rbind(accuracy, c("svm", conf_matrix$overall['Accuracy']))
write.csv(accuracy, "Accuracy.csv", row.names = FALSE)
heatmap(conf_matrix$table,
        Rowv = NA,
        Colv = NA,
        col = colorRampPalette(c("white", "blue"))(100), # Choose a color
gradient
```

```
        scale = "none",
        xlab = "predictedFactors",
        ylab = "testFactors",
        main = "Heatmap of Matrix",
        labRow = c(1,2,3,4),
        labCol = c(1,2,3,4))
predictedData = data.frame(testData$Traffic.Situation,
predictedTrafficSituation)
write.csv(predictedData, "svmPredictions.csv")
```

Output:

```
> conf_matrix
Confusion Matrix and Statistics

          predictedFactors
testFactors   1   2   3   4
          1  67  27   0   0
          2  19 465  18   6
          3   0  11  61  11
          4   3   0   4 201

Overall Statistics

               Accuracy : 0.8891
                 95% CI : (0.8667, 0.909)
    No Information Rate : 0.5633
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8163

 Mcnemar's Test P-Value : NA
```

```
Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity           0.75281   0.9245  0.73494   0.9220
Specificity           0.96642   0.8897  0.97284   0.9896
Pos Pred Value        0.71277   0.9154  0.73494   0.9663
Neg Pred Value        0.97247   0.9013  0.97284   0.9752
Prevalence            0.09966   0.5633  0.09295   0.2441
Detection Rate        0.07503   0.5207  0.06831   0.2251
Detection Prevalence  0.10526   0.5689  0.09295   0.2329
Balanced Accuracy     0.85961   0.9071  0.85389   0.9558
> conf_matrix$overall['Accuracy']
 Accuracy
0.8891377
```




6. xgbModel:

Code:

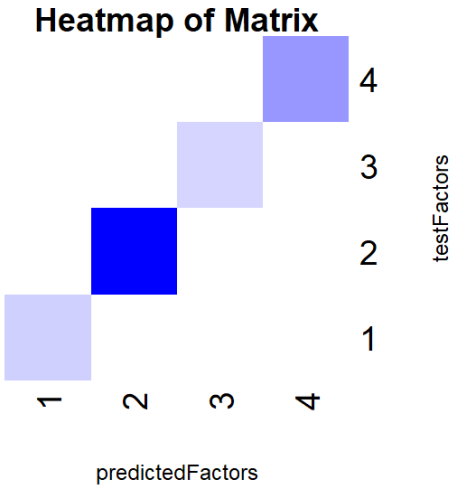
```
#library imports
library(ggplot2)
library(xgboost)
library(caret) #confusion matrix

#import dataset of one month traffic data
numericData <- read.csv("numericDataset.csv")
```

```
testData <- read.csv("test.csv")
trainData <- read.csv("train.csv")

#summary of test and training dataset
summary(testData)
summary(trainData)

#XGBoost
#Full dataset --> numericReplacedData
#Testing dataset --> test
#Training dataset --> train

numericData$Traffic.Situation <- as.factor(numericData$Traffic.Situation)
xgb_model <- xgboost(data = as.matrix(trainData[, !(names(trainData) %in%
"Traffic.Situation")]),
                     label = trainData$Traffic.Situation,
                     nrounds = 100, # Number of boosting rounds
                     verbose = TRUE)
testDataFrame <- as.matrix(testData[, -which(names(testData) ==
"Traffic.Situation")])
predictedTrafficSituation <- predict(xgb_model, testDataFrame)
actualTrafficSituation = as.factor(testData$Traffic.Situation)
predictedTrafficSituation <- factor(round(predictedTrafficSituation), levels =
levels(actualTrafficSituation))
conf_matrix <- confusionMatrix(actualTrafficSituation,
predictedTrafficSituation)
conf_matrix
conf_matrix$overall['Accuracy']
accuracy <- read.csv("Accuracy.csv", header = TRUE)
accuracy <- rbind(accuracy, c("xgb", conf_matrix$overall['Accuracy']))
write.csv(accuracy, "Accuracy.csv", row.names = FALSE)
heatmap(conf_matrix$table,
        Rowv = NA,
        Colv = NA,
        col = colorRampPalette(c("white", "blue"))(100), # Choose a color
gradient
        scale = "none",
        xlab = "predictedFactors",
        ylab = "testFactors",
        main = "Heatmap of Matrix",
        labRow = c(1,2,3,4),
        labCol = c(1,2,3,4))
predictedData = data.frame(testData$Traffic.Situation,
predictedTrafficSituation)
write.csv(predictedData, "xgbPredictions.csv")
```

Output:

```
> conf_matrix
Confusion Matrix and Statistics

          Reference
Prediction   1   2   3   4
         1  93   0   0   1
         2   0 508   0   0
         3   0   0  83   0
         4   0   0   0 208

Overall Statistics

               Accuracy : 0.9989
                 95% CI : (0.9938, 1)
    No Information Rate : 0.5689
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9981

 Mcnemar's Test P-Value : NA
```

```
Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity            1.0000   1.0000  1.00000   0.9952
Specificity            0.9988   1.0000  1.00000   1.0000
Pos Pred Value         0.9894   1.0000  1.00000   1.0000
Neg Pred Value         1.0000   1.0000  1.00000   0.9985
Prevalence             0.1041   0.5689  0.09295   0.2340
Detection Rate         0.1041   0.5689  0.09295   0.2329
Detection Prevalence   0.1053   0.5689  0.09295   0.2329
Balanced Accuracy      0.9994   1.0000  1.00000   0.9976
> conf_matrix$overall['Accuracy']
 Accuracy
0.9988802
```




7. Ensemble Predictions:

Code:

```
library(caret)

knn <- read.csv("knnPredictions.csv")
svm <- read.csv("svmPredictions.csv")
rf <- read.csv("rfPredictions.csv")
xgb <- read.csv("xgbPredictions.csv")
accuracy <- read.csv("Accuracy.csv")

predictedValuesdf <- data.frame(check = rep(NA, nrow(knn)))
predictedValuesdf$actual <- knn$testData.Traffic.Situation
predictedValuesdf$knn <- knn$predictedTrafficSituation
predictedValuesdf$svm <- svm$predictedTrafficSituation
predictedValuesdf$rf <- rf$predictedTrafficSituation
predictedValuesdf$xgb <- xgb$predictedTrafficSituation
predictedValuesdf <- predictedValuesdf[-1]

write.csv(predictedValuesdf, "predictedTraffic.csv")

logKnn <- data.frame(log(knn$predictedTrafficSituation))
```

```
logRf <- data.frame(log(rf$predictedTrafficSituation))
logSvm <- data.frame(log(svm$predictedTrafficSituation))
logXgb <- data.frame(log(xgb$predictedTrafficSituation))

meanPredicted <-
data.frame((logKnn$log.knn.predictedTrafficSituation.+logRf$log.rf.predictedT
rafficSituation.+logSvm$log.svm.predictedTrafficSituation.)/3)
ensemblePrediction <-
as.integer(exp(meanPredicted$X.logKnn.log.knn.predictedTrafficSituation....lo
gRf.log.rf.predictedTrafficSituation....))
actualPrediction <- predictedValuesdf$actual

conf_matrix <- confusionMatrix(table(actualPrediction, ensemblePrediction))
conf_matrix
conf_matrix$overall['Accuracy']
```

Output:

```
> conf_matrix
Confusion Matrix and Statistics

                ensemblePrediction
actualPrediction   1   2   3   4
               1  91   3   0   0
               2   9 496   2   1
               3   0  43  40   0
               4   0   3   2 203

Overall Statistics

               Accuracy : 0.9295
                 95% CI : (0.9106, 0.9454)
    No Information Rate : 0.6103
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.879

 Mcnemar's Test P-Value : NA
```

```
Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity            0.9100   0.9101  0.90909   0.9951
Specificity            0.9962   0.9655  0.94935   0.9927
Pos Pred Value         0.9681   0.9764  0.48193   0.9760
Neg Pred Value         0.9887   0.8727  0.99506   0.9985
Prevalence             0.1120   0.6103  0.04927   0.2284
Detection Rate         0.1019   0.5554  0.04479   0.2273
Detection Prevalence   0.1053   0.5689  0.09295   0.2329
Balanced Accuracy      0.9531   0.9378  0.92922   0.9939
> conf_matrix$overall['Accuracy']
 Accuracy
0.9294513
```