Group Assignment

Machine Learning

By:

Sanjeev Sidharth, Amol Pol, Anu Rathi, Jai Kushwaha

Problem Statement and Case:

**This project requires you to understand what mode of transport employees prefers to commute to their office. The attached dataset**[**Cars\_edited.csv**](https://olympus.greatlearning.in/courses/6123/files/714983/download?verifier=gUULzBemUZGdd5VYHQBUztRHKLKGurITGuv8NFbH&wrap=1)**includes employee information about their mode of transport as well as their personal and professional details like age, salary, work exp. We need to predict whether or not an employee will use Car as a mode of transport. Also, which variables are a significant predictor behind this decision?**

Setting the working directory and loading necessary packages

setwd("D:/Study/Great Lakes/machine learning/group assignment")  
library(class)  
library(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

library(mice)

##   
## Attaching package: 'mice'

## The following objects are masked from 'package:base':  
##   
## cbind, rbind

library(scales)  
library(superheat)  
library(corrplot)

## corrplot 0.84 loaded

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(caTools)  
library(usdm)

## Loading required package: sp

## Loading required package: raster

##   
## Attaching package: 'raster'

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

library(nnet)  
library(ROSE)

## Loaded ROSE 0.0-3

library(SmartEDA)

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

library(tidyr)

##   
## Attaching package: 'tidyr'

## The following object is masked from 'package:raster':  
##   
## extract

library(plyr)

## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

data = read.csv('Cars.csv', header = T, na.strings = NA)  
head(data)

## Age Gender Engineer MBA Work.Exp Salary Distance license Transport  
## 1 28 Male 0 0 4 14.3 3.2 0 Public Transport  
## 2 23 Female 1 0 4 8.3 3.3 0 Public Transport  
## 3 29 Male 1 0 7 13.4 4.1 0 Public Transport  
## 4 28 Female 1 1 5 13.4 4.5 0 Public Transport  
## 5 27 Male 1 0 4 13.4 4.6 0 Public Transport  
## 6 26 Male 1 0 4 12.3 4.8 1 Public Transport

## EDA

Five point summary

dim(data) # 444 rows and 9 columns

## [1] 444 9

# target variable -> Transport  
str(data)

## 'data.frame': 444 obs. of 9 variables:  
## $ Age : int 28 23 29 28 27 26 28 26 22 27 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 2 1 2 1 2 2 2 1 2 2 ...  
## $ Engineer : int 0 1 1 1 1 1 1 1 1 1 ...  
## $ MBA : int 0 0 0 1 0 0 0 0 0 0 ...  
## $ Work.Exp : int 4 4 7 5 4 4 5 3 1 4 ...  
## $ Salary : num 14.3 8.3 13.4 13.4 13.4 12.3 14.4 10.5 7.5 13.5 ...  
## $ Distance : num 3.2 3.3 4.1 4.5 4.6 4.8 5.1 5.1 5.1 5.2 ...  
## $ license : int 0 0 0 0 0 1 0 0 0 0 ...  
## $ Transport: Factor w/ 3 levels "2Wheeler","Car",..: 3 3 3 3 3 3 1 3 3 3 ...

# GEnder and transport factor data rest 6 columns are continuous data  
summary(data)

## Age Gender Engineer MBA Work.Exp   
## Min. :18.00 Female:128 Min. :0.0000 Min. :0.0000 Min. : 0.0   
## 1st Qu.:25.00 Male :316 1st Qu.:1.0000 1st Qu.:0.0000 1st Qu.: 3.0   
## Median :27.00 Median :1.0000 Median :0.0000 Median : 5.0   
## Mean :27.75 Mean :0.7545 Mean :0.2528 Mean : 6.3   
## 3rd Qu.:30.00 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.: 8.0   
## Max. :43.00 Max. :1.0000 Max. :1.0000 Max. :24.0   
## NA's :1   
## Salary Distance license Transport   
## Min. : 6.50 Min. : 3.20 Min. :0.0000 2Wheeler : 83   
## 1st Qu.: 9.80 1st Qu.: 8.80 1st Qu.:0.0000 Car : 61   
## Median :13.60 Median :11.00 Median :0.0000 Public Transport:300   
## Mean :16.24 Mean :11.32 Mean :0.2342   
## 3rd Qu.:15.72 3rd Qu.:13.43 3rd Qu.:0.0000   
## Max. :57.00 Max. :23.40 Max. :1.0000   
##

# General eda basis mean mode max min.

Observation: 1. There are 8 independent varaible and 1 target varaible i.e transport 2. 2 categorical variable are there. 3. We van observe there is 1 NA value present in the data.

# NA search  
anyNA(data)

## [1] TRUE

# There is na in data  
sum(is.na(data))

## [1] 1

# 1 na  
# MBA column has na data.  
sapply(data, function(x) sum(is.na(x)))

## Age Gender Engineer MBA Work.Exp Salary Distance license   
## 0 0 0 1 0 0 0 0   
## Transport   
## 0

Observation: 1. There is 1 NA value and that is present in the MBA column.

dat <- data %>%  
 mutate(  
 Engineer = factor(Engineer, levels = c(0,1), labels = c('NonEngineer','Engineer')),  
 MBA = factor(MBA, levels = c(0,1), labels = c('NonMBA', 'MBA')),  
 license = factor(license, levels = c(0,1), labels = c('NonDL', 'DL'))  
 )

Missing Value Imputation Using Mice

# NA imputation using Mice  
  
init = mice(dat, maxit=0)   
meth = init$method  
predM = init$predictorMatrix  
  
predM

## Age Gender Engineer MBA Work.Exp Salary Distance license Transport  
## Age 0 1 1 1 1 1 1 1 1  
## Gender 1 0 1 1 1 1 1 1 1  
## Engineer 1 1 0 1 1 1 1 1 1  
## MBA 1 1 1 0 1 1 1 1 1  
## Work.Exp 1 1 1 1 0 1 1 1 1  
## Salary 1 1 1 1 1 0 1 1 1  
## Distance 1 1 1 1 1 1 0 1 1  
## license 1 1 1 1 1 1 1 0 1  
## Transport 1 1 1 1 1 1 1 1 0

meth # logreg

## Age Gender Engineer MBA Work.Exp Salary Distance license   
## "" "" "" "logreg" "" "" "" ""   
## Transport   
## ""

imputed = mice(dat, method=meth, predictorMatrix=predM)

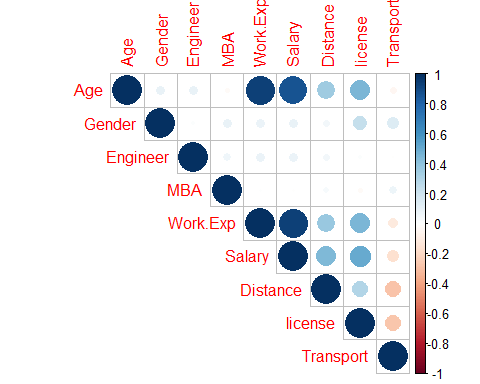
##   
## iter imp variable  
## 1 1 MBA  
## 1 2 MBA  
## 1 3 MBA  
## 1 4 MBA  
## 1 5 MBA  
## 2 1 MBA  
## 2 2 MBA  
## 2 3 MBA  
## 2 4 MBA  
## 2 5 MBA  
## 3 1 MBA  
## 3 2 MBA  
## 3 3 MBA  
## 3 4 MBA  
## 3 5 MBA  
## 4 1 MBA  
## 4 2 MBA  
## 4 3 MBA  
## 4 4 MBA  
## 4 5 MBA  
## 5 1 MBA  
## 5 2 MBA  
## 5 3 MBA  
## 5 4 MBA  
## 5 5 MBA

imputed <- complete(imputed)  
  
sapply(imputed, function(x) sum(is.na(x)))

## Age Gender Engineer MBA Work.Exp Salary Distance license   
## 0 0 0 0 0 0 0 0   
## Transport   
## 0

Correlation Plot

data\_numeric = imputed %>%  
 mutate ( Gender = as.numeric(Gender),  
 Engineer = as.numeric(Engineer),  
 MBA = as.numeric(MBA),  
 license = as.numeric(license),  
 Transport = as.numeric(Transport)  
 )  
M <- cor(data\_numeric)  
corrplot(M, type="upper")

 Observation: 1. Age is highly correlated with salary and Work EXP 2. Work exp and salary are also highly corelated.

# Target value ratio  
prop.table(table(imputed$Transport))

##   
## 2Wheeler Car Public Transport   
## 0.1869369 0.1373874 0.6756757

Observation: 1. 67% data is of Public transport and baised towards it.

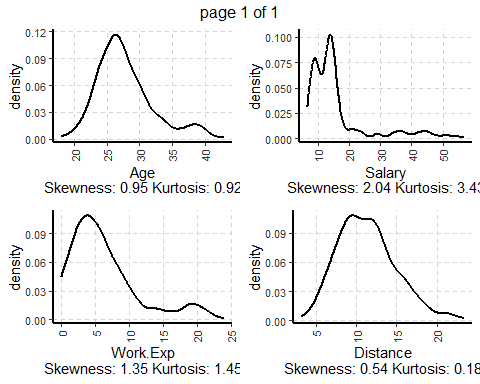
General Summary after imputation

ExpData(data=imputed,type=1)

## Descriptions Obs  
## 1 Sample size (Nrow) 444  
## 2 No. of Variables (Ncol) 9  
## 3 No. of Numeric Variables 4  
## 4 No. of Factor Variables 5  
## 5 No. of Text Variables 0  
## 6 No. of Logical Variables 0  
## 7 No. of Unique Variables 0  
## 8 No. of Date Variables 0  
## 9 No. of Zero variance Variables (Uniform) 0  
## 10 %. of Variables having complete cases 100% (9)  
## 11 %. of Variables having <50% missing cases 0% (0)  
## 12 %. of Variables having >50% missing cases 0% (0)  
## 13 %. of Variables having >90% missing cases 0% (0)

Univariate analysis

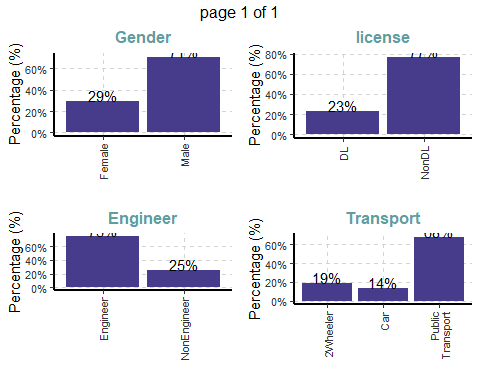
# Desnity plot for numerical data (univariate)  
plot1 <- ExpNumViz(imputed,target=NULL,nlim=10,Page=c(2,2),sample=4)  
plot1[[1]]



# Frequency chart for categorical variable (univariate)  
ExpCTable(imputed,Target=NULL,margin=1,clim=10,nlim=3,round=2,bin=NULL,per=T)

## Variable Valid Frequency Percent CumPercent  
## 1 Gender Female 128 28.83 28.83  
## 2 Gender Male 316 71.17 100.00  
## 3 Gender TOTAL 444 NA NA  
## 4 Engineer Engineer 335 75.45 75.45  
## 5 Engineer NonEngineer 109 24.55 100.00  
## 6 Engineer TOTAL 444 NA NA  
## 7 MBA MBA 112 25.23 25.23  
## 8 MBA NonMBA 332 74.77 100.00  
## 9 MBA TOTAL 444 NA NA  
## 10 license DL 104 23.42 23.42  
## 11 license NonDL 340 76.58 100.00  
## 12 license TOTAL 444 NA NA  
## 13 Transport 2Wheeler 83 18.69 18.69  
## 14 Transport Car 61 13.74 32.43  
## 15 Transport Public Transport 300 67.57 100.00  
## 16 Transport TOTAL 444 NA NA

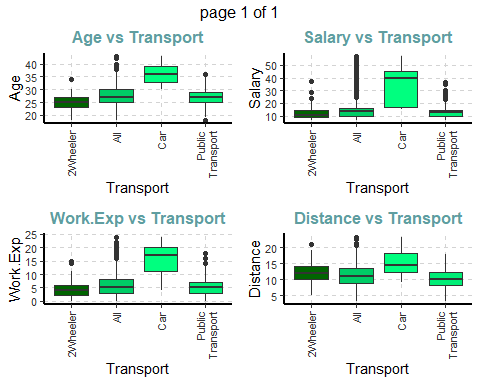
# BArplot for categorical data  
plot2 <- ExpCatViz(imputed,target=NULL,col ="slateblue4",clim=10,margin=2,Page = c(2,2),sample=4)  
plot2[[1]]



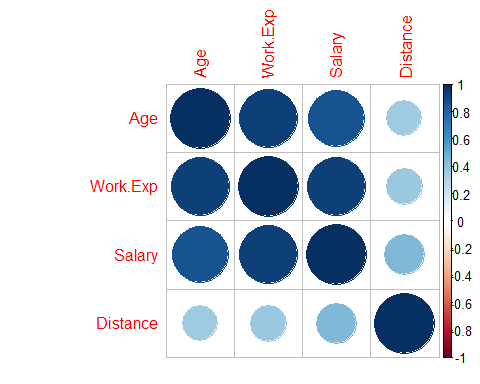
# for numeric data  
ExpNumStat(imputed,by="GA",gp="Transport",Qnt=seq(0,1,0.1),MesofShape=2,Outlier=TRUE,round=2)

## Vname Group TN nNeg nZero nPos NegInf PosInf  
## 1 Age Transport:All 444 0 0 444 0 0  
## 5 Age Transport:Public Transport 300 0 0 300 0 0  
## 9 Age Transport:2Wheeler 83 0 0 83 0 0  
## 13 Age Transport:Car 61 0 0 61 0 0  
## 4 Distance Transport:All 444 0 0 444 0 0  
## 8 Distance Transport:Public Transport 300 0 0 300 0 0  
## 12 Distance Transport:2Wheeler 83 0 0 83 0 0  
## 16 Distance Transport:Car 61 0 0 61 0 0  
## 3 Salary Transport:All 444 0 0 444 0 0  
## 7 Salary Transport:Public Transport 300 0 0 300 0 0  
## 11 Salary Transport:2Wheeler 83 0 0 83 0 0  
## 15 Salary Transport:Car 61 0 0 61 0 0  
## 2 Work.Exp Transport:All 444 0 29 415 0 0  
## 6 Work.Exp Transport:Public Transport 300 0 16 284 0 0  
## 10 Work.Exp Transport:2Wheeler 83 0 13 70 0 0  
## 14 Work.Exp Transport:Car 61 0 0 61 0 0  
## NA\_Value Per\_of\_Missing sum min max mean median SD CV IQR  
## 1 0 0 12320.0 18.0 43.0 27.75 27.0 4.42 0.16 5.00  
## 5 0 0 8044.0 18.0 36.0 26.81 27.0 2.96 0.11 4.00  
## 9 0 0 2097.0 18.0 34.0 25.27 25.0 2.86 0.11 4.00  
## 13 0 0 2179.0 30.0 43.0 35.72 36.0 3.43 0.10 6.00  
## 4 0 0 5027.5 3.2 23.4 11.32 11.0 3.61 0.32 4.62  
## 8 0 0 3094.5 3.2 17.9 10.31 10.0 3.01 0.29 4.05  
## 12 0 0 999.7 5.1 21.0 12.04 11.9 3.32 0.28 4.10  
## 16 0 0 933.3 9.0 23.4 15.30 14.4 3.74 0.24 5.80  
## 3 0 0 7210.0 6.5 57.0 16.24 13.6 10.45 0.64 5.92  
## 7 0 0 3953.0 6.5 36.6 13.18 12.9 4.81 0.36 4.82  
## 11 0 0 1046.6 6.5 37.0 12.61 10.6 6.05 0.48 5.90  
## 15 0 0 2210.4 15.6 57.0 36.24 39.9 13.04 0.36 28.00  
## 2 0 0 2797.0 0.0 24.0 6.30 5.0 5.11 0.81 5.00  
## 6 0 0 1505.0 0.0 18.0 5.02 5.0 3.16 0.63 4.00  
## 10 0 0 337.0 0.0 15.0 4.06 4.0 3.32 0.82 4.00  
## 14 0 0 955.0 4.0 24.0 15.66 17.0 4.88 0.31 9.00  
## Skewness Kurtosis 0% 10% 20% 30% 40% 50% 60% 70% 80% 90%  
## 1 0.95 0.92 18.0 23.00 24.0 25.00 26.00 27.0 28.00 29.00 30.00 34.00  
## 5 0.17 0.48 18.0 23.00 24.0 25.00 26.00 27.0 27.00 28.00 29.00 30.00  
## 9 0.53 0.81 18.0 22.00 23.0 24.00 24.00 25.0 26.00 26.00 28.00 29.00  
## 13 0.03 -1.23 30.0 31.00 32.0 33.00 34.00 36.0 38.00 38.00 39.00 40.00  
## 4 0.54 0.18 3.2 7.10 8.2 9.10 10.00 11.0 11.98 12.90 14.14 16.27  
## 8 0.29 -0.39 3.2 6.49 7.7 8.57 9.30 10.0 10.84 11.90 12.82 14.51  
## 12 0.16 -0.26 5.1 7.50 9.2 10.70 11.18 11.9 12.74 13.44 15.20 16.28  
## 16 0.31 -0.88 9.0 10.90 11.9 12.60 13.80 14.4 16.30 17.80 18.60 20.80  
## 3 2.04 3.43 6.5 8.50 8.9 10.60 12.62 13.6 14.60 14.90 17.00 34.87  
## 7 2.02 6.16 6.5 8.50 8.9 10.57 12.14 12.9 13.80 14.60 14.82 16.51  
## 11 2.03 4.90 6.5 7.12 8.5 8.80 9.40 10.6 12.80 13.70 14.90 20.50  
## 15 -0.49 -1.10 15.6 16.90 17.0 33.00 36.90 39.9 42.90 45.00 47.00 51.00  
## 2 1.35 1.45 0.0 1.00 2.0 3.00 4.00 5.0 6.00 7.00 9.00 14.00  
## 6 0.93 1.46 0.0 1.00 2.0 3.00 4.00 5.0 5.40 6.00 7.00 9.00  
## 10 0.89 0.71 0.0 0.00 1.0 2.00 2.00 4.0 5.00 5.00 7.00 8.00  
## 14 -0.40 -1.00 4.0 9.00 11.0 12.00 14.00 17.0 19.00 19.00 20.00 21.00  
## 100% LB.25% UB.75% nOutliers  
## 1 43.0 17.50 37.50 25  
## 5 36.0 19.00 35.00 3  
## 9 34.0 17.00 33.00 2  
## 13 43.0 24.00 48.00 0  
## 4 23.4 1.86 20.36 9  
## 8 17.9 2.10 18.30 0  
## 12 21.0 3.75 20.15 1  
## 16 23.4 3.60 26.80 0  
## 3 57.0 0.91 24.61 59  
## 7 36.6 2.56 21.86 15  
## 11 37.0 -0.10 23.50 5  
## 15 57.0 -25.00 87.00 0  
## 2 24.0 -4.50 15.50 38  
## 6 18.0 -3.00 13.00 6  
## 10 15.0 -4.00 12.00 2  
## 14 24.0 -2.50 33.50 0

# Box plot  
plot4 <- ExpNumViz(imputed,target="Transport",type=1,nlim=3,fname=NULL,col=c("darkgreen","springgreen3","springgreen1","springgreen2"),Page=c(2,2),sample=4)  
plot4[[1]]

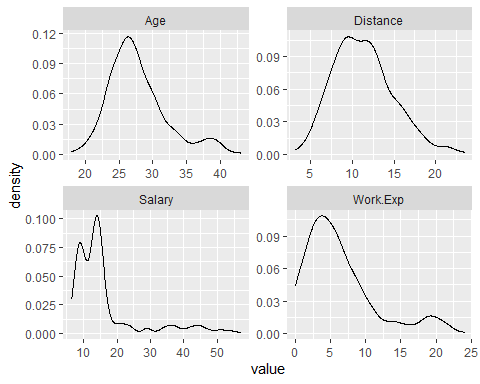
 # Corrplot for NUmeric adata

# corrplot  
imputed %>%  
 #filter(Transport == "Car") %>%  
 select\_if(is.numeric) %>%  
 cor() %>%  
 corrplot::corrplot()



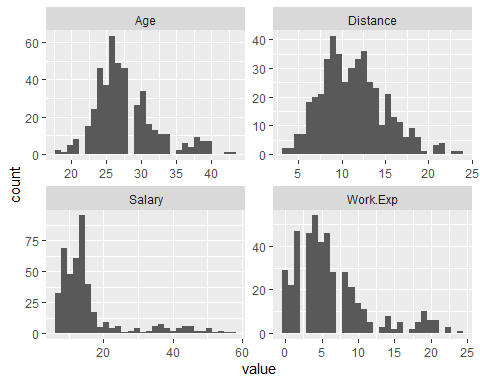
Observation: 1. Age, workexp and salary are highly corelated.

# Density plot for numeric data  
imputed %>%  
 select\_if(is.numeric) %>%   
 gather() %>%   
 ggplot(aes(value)) +   
 facet\_wrap(~ key, scales = "free") +   
 geom\_density()

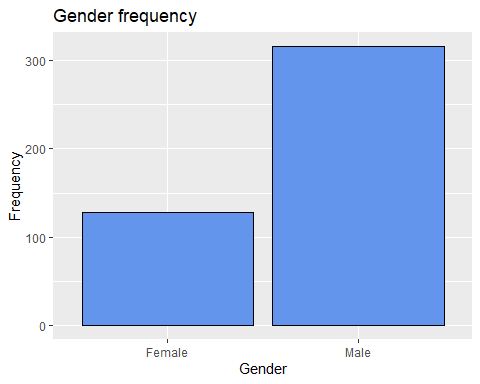


# Histogram for numeric data  
imputed %>%  
 select\_if(is.numeric) %>%   
 gather() %>%   
 ggplot(aes(value)) +  
 facet\_wrap(~ key, scales = "free") +  
 geom\_histogram()

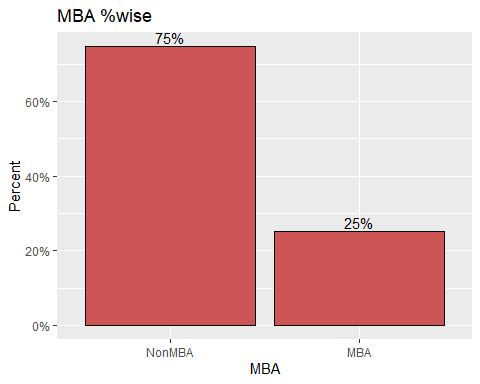
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



# Categorical Univariate  
# Gender  
ggplot(imputed, aes(x = Gender)) +   
 geom\_bar(fill = "cornflowerblue", color="black") +  
 labs(x = "Gender", y = "Frequency", title = "Gender frequency")



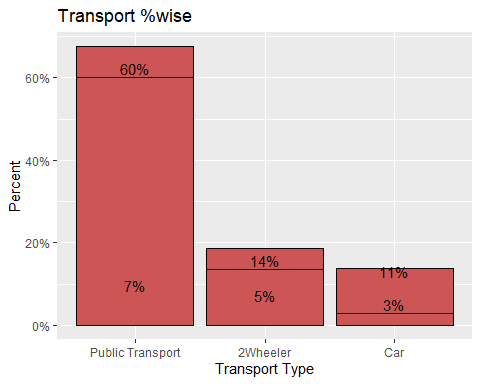
plotdata <- imputed %>%  
 count("MBA") %>%  
 mutate(pct =freq / sum(freq),  
 pctlabel = paste0(round(pct\*100), "%"))  
# plot the bars as percentages,   
# in decending order with bar labels  
ggplot(plotdata,   
 aes(x = reorder(MBA, -pct),  
 y = pct)) +   
 geom\_bar(stat = "identity",   
 fill = "indianred3",   
 color = "black") +  
 geom\_text(aes(label = pctlabel),   
 vjust = -0.25) +  
 scale\_y\_continuous(labels = percent) +  
 labs(x = "MBA",   
 y = "Percent",   
 title = "MBA %wise")



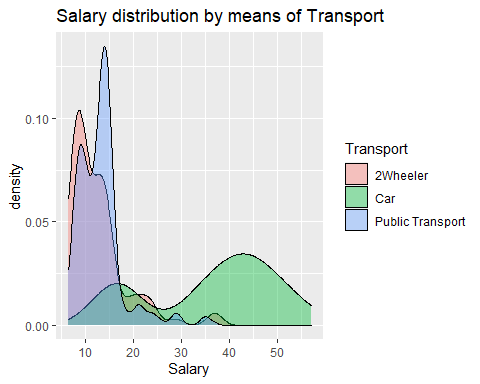
# 75 % are Non Mba employess and 25 % are mba employees

License vs Transport

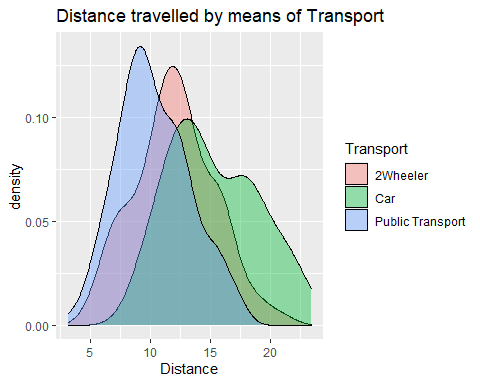
plotdata <- imputed %>% count(c("license", 'Transport')) %>% mutate(pct =freq / sum(freq),pctlabel = paste0(round(pct\*100), "%"))  
#plotdata  
# plot the bars as percentages,   
# in decending order with bar labels  
ggplot(plotdata, aes(x = reorder(Transport, -pct),y = pct)) +   
 geom\_bar(stat = "identity", fill = "indianred3", color = "black", position = 'stack') +  
 geom\_text(aes(label = pctlabel), vjust = -0.25) +  
 scale\_y\_continuous(labels = percent) +labs(x = "Transport Type", y = "Percent", title = "Transport %wise")

 Salary Vs Transport

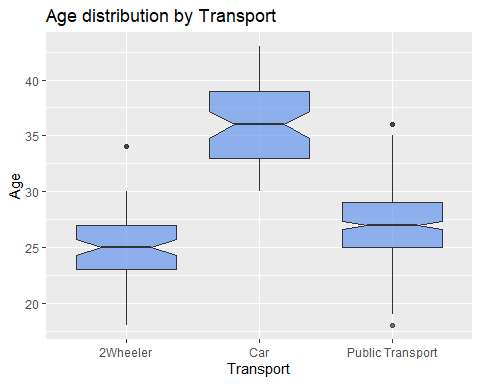
ggplot(imputed,   
 aes(x = Salary,   
 fill = Transport)) +  
 geom\_density(alpha = 0.4) +  
 labs(title = "Salary distribution by means of Transport")

 Observation: # HIgh Salried prople use casr # 2 wheeler and public transport is mostly used by low income salary group

# Transport vs Distance  
ggplot(imputed,   
 aes(x = Distance,   
 fill = Transport)) +  
 geom\_density(alpha = 0.4) +  
 labs(title = "Distance travelled by means of Transport")

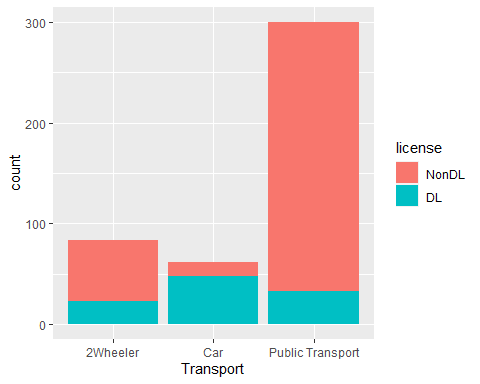
 Observation: # Short distance is mostly uses public transport # For long distnce car is used by employees

# Transport vs Age  
ggplot(imputed, aes(x = Transport,y = Age)) +  
 geom\_boxplot(notch = TRUE, fill = "cornflowerblue", alpha = .7) +  
 labs(title = "Age distribution by Transport")



# Younger age group uses 2 wheeler and employee aged above 30 mostly uses car.

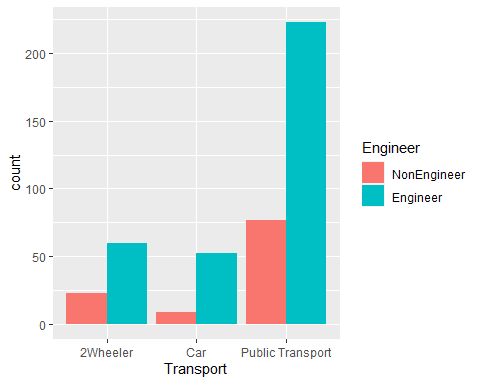
# Transport vs License  
ggplot(imputed,   
 aes(x = Transport,   
 fill = license)) +   
 geom\_bar(position = "stack")



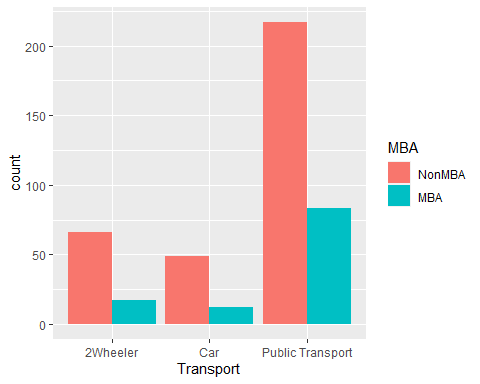
# Non Driving license people are using the puiblic transport .  
# people with license are using Car  
# Most 2 wheeler user are not having Driving lIcense

Engineer Vs Transport

ggplot(imputed,   
 aes(x = Transport,   
 fill = Engineer)) +   
 geom\_bar(position = position\_dodge(preserve = "single"))

 MBA vs Transport

ggplot(imputed,   
 aes(x = Transport,   
 fill = MBA)) +   
 geom\_bar(position = position\_dodge(preserve = "single")) #+

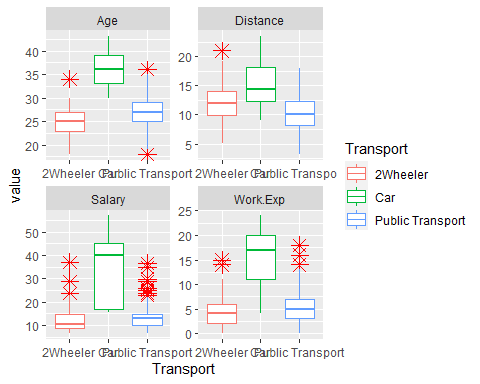


#geom\_text(count)

Observation: 1. There are more Non Mba persons in all three transport categories.

Outlier Analysis

imputed %>%  
 dplyr::select("Age","Work.Exp","Salary","Distance", "Transport") %>%   
 tidyr::gather(Measure, value, -Transport) %>% # Convert to key-value pairs  
 ggplot(aes(x = Transport, y = value, color=Transport)) + # Plot the values  
 geom\_boxplot(outlier.colour="red", outlier.shape=8,  
 outlier.size=4)+  
 facet\_wrap(~ Measure, scales = "free") # In separate panels



Multicolinearity Check

vif(data\_numeric)

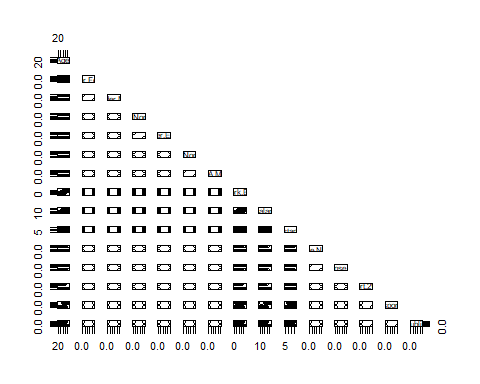
## Variables VIF  
## 1 Age 8.191802  
## 2 Gender 1.128719  
## 3 Engineer 1.015725  
## 4 MBA 1.039560  
## 5 Work.Exp 15.818172  
## 6 Salary 8.887503  
## 7 Distance 1.349063  
## 8 license 1.563280  
## 9 Transport 1.268938

# Age Work Exp and salar have high multicolinearity. That was also evident by the corplot

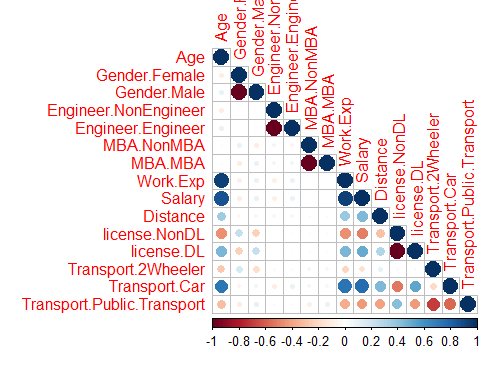
# Check for multicolinearity  
# One hot encoded data  
d2 <- data.frame(predict(dummyVars(~., imputed), imputed))  
str(d2)

## 'data.frame': 444 obs. of 15 variables:  
## $ Age : num 28 23 29 28 27 26 28 26 22 27 ...  
## $ Gender.Female : num 0 1 0 1 0 0 0 1 0 0 ...  
## $ Gender.Male : num 1 0 1 0 1 1 1 0 1 1 ...  
## $ Engineer.NonEngineer : num 1 0 0 0 0 0 0 0 0 0 ...  
## $ Engineer.Engineer : num 0 1 1 1 1 1 1 1 1 1 ...  
## $ MBA.NonMBA : num 1 1 1 0 1 1 1 1 1 1 ...  
## $ MBA.MBA : num 0 0 0 1 0 0 0 0 0 0 ...  
## $ Work.Exp : num 4 4 7 5 4 4 5 3 1 4 ...  
## $ Salary : num 14.3 8.3 13.4 13.4 13.4 12.3 14.4 10.5 7.5 13.5 ...  
## $ Distance : num 3.2 3.3 4.1 4.5 4.6 4.8 5.1 5.1 5.1 5.2 ...  
## $ license.NonDL : num 1 1 1 1 1 0 1 1 1 1 ...  
## $ license.DL : num 0 0 0 0 0 1 0 0 0 0 ...  
## $ Transport.2Wheeler : num 0 0 0 0 0 0 1 0 0 0 ...  
## $ Transport.Car : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Transport.Public.Transport: num 1 1 1 1 1 1 0 1 1 1 ...

# 15 columns not one hot encoded  
  
pairs(d2, upper.panel = NULL)



corrplot(cor(d2), type = 'lower')



pcs <- prcomp(d2, center = T, scale. = T, tol = 0.8)  
print(pcs)

## Standard deviations (1, .., p=15):  
## [1] 2.217036e+00 1.538006e+00 1.431043e+00 1.342214e+00 1.296373e+00  
## [6] 9.834883e-01 8.188768e-01 6.198273e-01 3.571386e-01 1.993525e-01  
## [11] 1.060846e-15 5.846732e-16 5.164963e-16 3.882144e-16 3.491764e-16  
##   
## Rotation (n x k) = (15 x 1):  
## PC1  
## Age -0.38484360  
## Gender.Female 0.09916395  
## Gender.Male -0.09916395  
## Engineer.NonEngineer 0.06162668  
## Engineer.Engineer -0.06162668  
## MBA.NonMBA -0.01105180  
## MBA.MBA 0.01105180  
## Work.Exp -0.39440790  
## Salary -0.40596240  
## Distance -0.24178447  
## license.NonDL 0.33848902  
## license.DL -0.33848902  
## Transport.2Wheeler 0.03421775  
## Transport.Car -0.38690267  
## Transport.Public.Transport 0.25603024

# PLot for Work exp and Salary wrt to transport  
library(ggpubr)

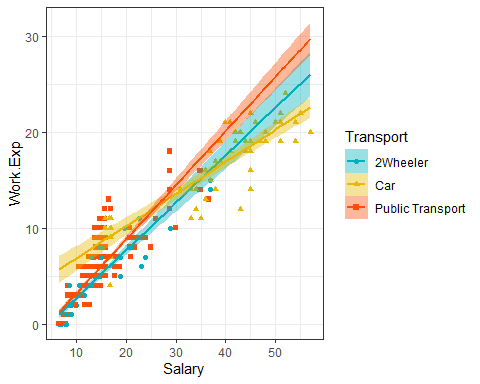
##   
## Attaching package: 'ggpubr'

## The following object is masked from 'package:plyr':  
##   
## mutate

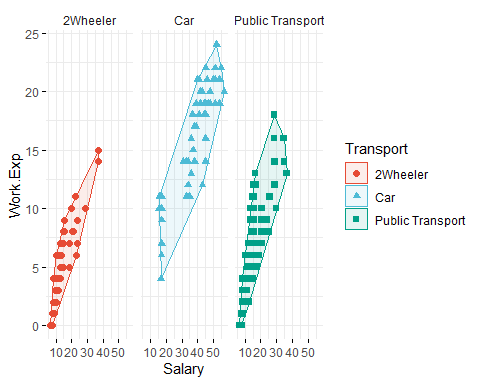
## The following object is masked from 'package:raster':  
##   
## rotate

b <- ggplot(data, aes(x = Salary, y = Work.Exp))  
b + geom\_point(aes(color = Transport, shape = Transport))+  
 geom\_smooth(aes(color = Transport, fill = Transport),  
 method = "lm", fullrange = TRUE) +  
 #facet\_wrap(~Transport) +  
 scale\_color\_manual(values = c("#00AFBB", "#E7B800", "#FC4E07"))+  
 scale\_fill\_manual(values = c("#00AFBB", "#E7B800", "#FC4E07")) +  
 theme\_bw()

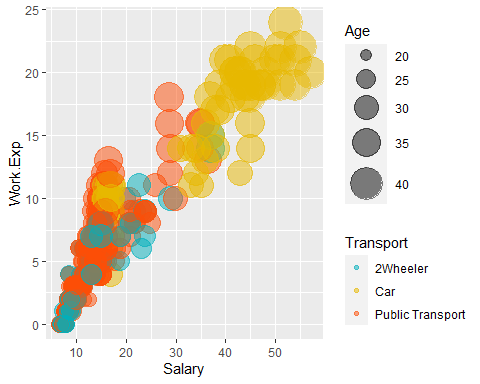
## `geom\_smooth()` using formula 'y ~ x'



# Change the ellipse type to 'convex'  
ggscatter(data, x = "Salary", y = "Work.Exp",  
 color = "Transport", palette = "npg",  
 shape = "Transport",  
 ellipse = TRUE, ellipse.type = "convex",  
 ggtheme = theme\_minimal())+  
 facet\_wrap(~Transport)



# Bubble chart ( Salary , Work Exp, Age and Transport)  
b + geom\_point(aes(color = Transport, size = Age), alpha = 0.5) +  
 scale\_color\_manual(values = c("#00AFBB", "#E7B800", "#FC4E07")) +  
 scale\_size(range = c(0.5, 12)) # Adjust the range of points size

 Business Requirement as we understand is need to find out employees who would use car as a means of transport and factor s impacting in it. Observation: 1. From the bubble we can understand that People with High exp prefer car as mode of transport. 2. From the 1st chart we can see that work exp and salary . We can see that for car users slope is least which means. With less work exp they are getting more salary which enable them to have a car.

Data Spliting and ploting

# Data prepration as Train and Test  
split <- sample.split(imputed$Transport, SplitRatio = .75)  
  
cartrain <- subset( imputed, split==T)  
cartest<- subset(imputed, split==F)  
dim(cartrain) # 111 rows

## [1] 333 9

dim(cartest) # 333 row

## [1] 111 9

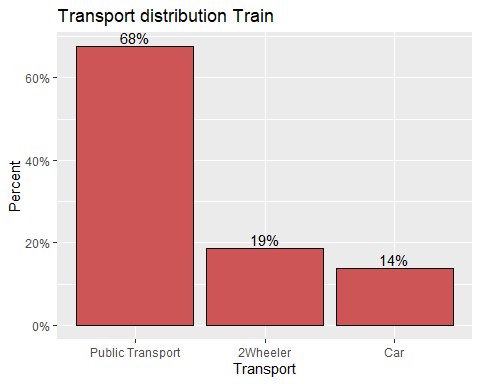
prop.table(table(cartrain$Transport))

##   
## 2Wheeler Car Public Transport   
## 0.1861862 0.1381381 0.6756757

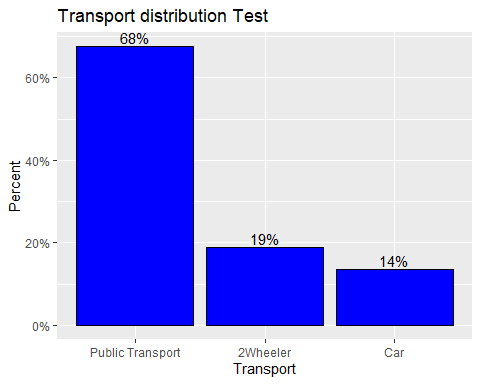
prop.table(table(cartest$Transport))

##   
## 2Wheeler Car Public Transport   
## 0.1891892 0.1351351 0.6756757

#par(mfrow=c(1,2))  
plotdata1 <- cartrain %>% count("Transport") %>% mutate(pct =freq / sum(freq),pctlabel = paste0(round(pct\*100), "%"))  
plotdata2 <- cartest %>% count("Transport") %>% mutate(pct =freq / sum(freq),pctlabel = paste0(round(pct\*100), "%"))  
# plot the bars as percentages,   
# in decending order with bar labels  
ggplot(plotdata1, aes(x = reorder(Transport, -pct),y = pct)) +   
 geom\_bar(stat = "identity", fill = "indianred3", color = "black") +  
 geom\_text(aes(label = pctlabel), vjust = -0.25) +  
 scale\_y\_continuous(labels = percent) +labs(x = "Transport", y = "Percent", title = "Transport distribution Train")



ggplot(plotdata2, aes(x = reorder(Transport, -pct),y = pct)) +   
 geom\_bar(stat = "identity", fill = "blue", color = "black") +  
 geom\_text(aes(label = pctlabel), vjust = -0.25) +  
 scale\_y\_continuous(labels = percent) +labs(x = "Transport", y = "Percent", title = "Transport distribution Test")



#grid.arrange(plot1, plot2, ncol=2)  
# distribution is same for both test and train data

# Normalising the data  
normalize <- function(x) {  
 return ((x - min(x)) / (max(x) - min(x))) }  
cartest<- cartest %>% mutate\_if(is.numeric, normalize)  
cartrain<- cartrain %>% mutate\_if(is.numeric, normalize)  
summary(cartest)

## Age Gender Engineer MBA Work.Exp   
## Min. :0.0000 Female:34 NonEngineer:29 NonMBA:83 Min. :0.00000   
## 1st Qu.:0.2727 Male :77 Engineer :82 MBA :28 1st Qu.:0.09091   
## Median :0.3636 Median :0.18182   
## Mean :0.4369 Mean :0.28624   
## 3rd Qu.:0.5455 3rd Qu.:0.38636   
## Max. :1.0000 Max. :1.00000   
## Salary Distance license Transport   
## Min. :0.00000 Min. :0.0000 NonDL:85 2Wheeler :21   
## 1st Qu.:0.04183 1st Qu.:0.2169 DL :26 Car :15   
## Median :0.11952 Median :0.3675 Public Transport:75   
## Mean :0.19086 Mean :0.3884   
## 3rd Qu.:0.17729 3rd Qu.:0.5181   
## Max. :1.00000 Max. :1.0000

summary(cartrain)

## Age Gender Engineer MBA Work.Exp   
## Min. :0.0000 Female: 94 NonEngineer: 80 NonMBA:249 Min. :0.0000   
## 1st Qu.:0.2800 Male :239 Engineer :253 MBA : 84 1st Qu.:0.1250   
## Median :0.3600 Median :0.2083   
## Mean :0.3917 Mean :0.2625   
## 3rd Qu.:0.4800 3rd Qu.:0.3333   
## Max. :1.0000 Max. :1.0000   
## Salary Distance license Transport   
## Min. :0.0000 Min. :0.0000 NonDL:255 2Wheeler : 62   
## 1st Qu.:0.0701 1st Qu.:0.2822 DL : 78 Car : 46   
## Median :0.1485 Median :0.3911 Public Transport:225   
## Mean :0.1998 Mean :0.4034   
## 3rd Qu.:0.1918 3rd Qu.:0.5050   
## Max. :1.0000 Max. :1.0000

Naive Bayes

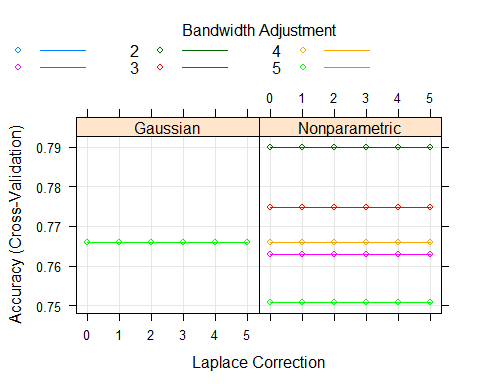
1. The main limitation of Naive Bayes is the assumption of independent predictor features. Naive Bayes implicitly assumes that all the attributes are mutually independent. In real life, it’s almost impossible that we get a set of predictors that are completely independent or one another.
2. If a categorical variable has a category in the test dataset, which was not observed in training dataset, then the model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as Zero Frequency. To solve this, we can use a smoothing technique. Details on additive smoothing or laplace smoothing can be found here.
3. Cannot incorporate feature interactions.
4. For regression problems, i.e. continuous real-valued data, there may not be a good way to calculate likelihoods. Binning the data and assigning discrete classes to the bins is sub-optimal since it throws away information. Assuming each feature is normally distributed is workable, but could impact performance if features are not normally distributed. On the other hand, with enough training data in each class, you could estimate the likelihood densities directly, permitting accurate likelihood calculations for new data.
5. Performance is sensitive to skewed data — that is, when the training data is not representative of the class distributions in the overall population. In this case, the prior estimates will be incorrect.

# So, multi collinearity does not affect the Naive Bayes  
# MOdel 1 Naive Bayes  
train\_control <- trainControl(  
 method = "cv",   
 number = 3 )  
search\_grid <- expand.grid(  
 usekernel = c(TRUE, FALSE),  
 fL = 0:5,  
 adjust = seq(0, 5, by = 1) )  
  
library(caret)  
  
nb.m1 <- caret::train(  
 Transport ~ ., data = cartrain,  
 method = "nb",  
 trControl = train\_control,  
 tuneGrid = search\_grid,  
 preProc = c("BoxCox", "center", "scale", "pca")  
)

#warnings()  
# top 5 modesl  
nb.m1$results %>%   
 top\_n(5, wt = Accuracy) %>%  
 arrange(desc(Accuracy))

## usekernel fL adjust Accuracy Kappa AccuracySD KappaSD  
## 1 TRUE 0 2 0.7899136 0.4947807 0.02076712 0.03870763  
## 2 TRUE 1 2 0.7899136 0.4947807 0.02076712 0.03870763  
## 3 TRUE 2 2 0.7899136 0.4947807 0.02076712 0.03870763  
## 4 TRUE 3 2 0.7899136 0.4947807 0.02076712 0.03870763  
## 5 TRUE 4 2 0.7899136 0.4947807 0.02076712 0.03870763  
## 6 TRUE 5 2 0.7899136 0.4947807 0.02076712 0.03870763

# plot search grid results  
plot(nb.m1)



confusionMatrix(nb.m1) # train accuracy # 78.38

## Cross-Validated (3 fold) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 3.0 0.6 1.5  
## Car 0.6 10.8 0.9  
## Public Transport 15.0 2.4 65.2  
##   
## Accuracy (average) : 0.7898

# Test prediction  
data\_pred <- predict(nb.m1,cartest[,-9],type="raw")  
confusionMatrix(data\_pred,cartest[,9], positive = "a", mode="everything") # test accuracy 83.78

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 3 1 0  
## Car 2 12 2  
## Public Transport 16 2 73  
##   
## Overall Statistics  
##   
## Accuracy : 0.7928   
## 95% CI : (0.7055, 0.8639)  
## No Information Rate : 0.6757   
## P-Value [Acc > NIR] : 0.0044228   
##   
## Kappa : 0.5064   
##   
## Mcnemar's Test P-Value : 0.0009688   
##   
## Statistics by Class:  
##   
## Class: 2Wheeler Class: Car Class: Public Transport  
## Sensitivity 0.14286 0.8000 0.9733  
## Specificity 0.98889 0.9583 0.5000  
## Pos Pred Value 0.75000 0.7500 0.8022  
## Neg Pred Value 0.83178 0.9684 0.9000  
## Precision 0.75000 0.7500 0.8022  
## Recall 0.14286 0.8000 0.9733  
## F1 0.24000 0.7742 0.8795  
## Prevalence 0.18919 0.1351 0.6757  
## Detection Rate 0.02703 0.1081 0.6577  
## Detection Prevalence 0.03604 0.1441 0.8198  
## Balanced Accuracy 0.56587 0.8792 0.7367

###################################################################################################

KNN Algorithm

source("Confusion matrix plot.r")

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

set.seed(400)  
ctrl <- trainControl(method="repeatedcv",repeats = 3) #,classProbs=TRUE,summaryFunction = twoClassSummary)  
knnFit <- caret::train(Transport ~ ., data = cartrain, method = "knn", trControl = ctrl, preProcess = c("center","scale"), tuneLength = 20)  
data\_pred <- predict(knnFit,cartest[,-9],type="raw")  
confusionMatrix(data\_pred,cartest[,9], positive = "a", mode="everything") # 78.38

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 4 1 0  
## Car 2 11 2  
## Public Transport 15 3 73  
##   
## Overall Statistics  
##   
## Accuracy : 0.7928   
## 95% CI : (0.7055, 0.8639)  
## No Information Rate : 0.6757   
## P-Value [Acc > NIR] : 0.004423   
##   
## Kappa : 0.5058   
##   
## Mcnemar's Test P-Value : 0.001413   
##   
## Statistics by Class:  
##   
## Class: 2Wheeler Class: Car Class: Public Transport  
## Sensitivity 0.19048 0.7333 0.9733  
## Specificity 0.98889 0.9583 0.5000  
## Pos Pred Value 0.80000 0.7333 0.8022  
## Neg Pred Value 0.83962 0.9583 0.9000  
## Precision 0.80000 0.7333 0.8022  
## Recall 0.19048 0.7333 0.9733  
## F1 0.30769 0.7333 0.8795  
## Prevalence 0.18919 0.1351 0.6757  
## Detection Rate 0.03604 0.0991 0.6577  
## Detection Prevalence 0.04505 0.1351 0.8198  
## Balanced Accuracy 0.58968 0.8458 0.7367

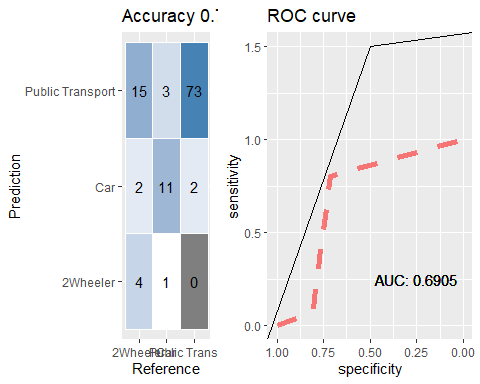
cfmROC(data\_pred,cartest[,9])

## Warning in roc.default(as.numeric(y\_true) - 1, as.numeric(y\_pred) - 1):  
## 'response' has more than two levels. Consider setting 'levels' explicitly or  
## using 'multiclass.roc' instead

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

## Setting direction: controls > cases

## Warning: Removed 1 row(s) containing missing values (geom\_path).



ctrl <- trainControl(method="repeatedcv",repeats = 3,classProbs=FALSE,summaryFunction = multiClassSummary)  
knnFit <- caret::train(Transport ~ ., data = cartrain, method = "knn", trControl = ctrl, preProcess = c("center","scale"), tuneLength = 20)

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :  
## There were missing values in resampled performance measures.

data\_pred <- predict(knnFit,cartest[,-9],type="raw")  
confusionMatrix(data\_pred,cartest[,9], positive = "a", mode="everything") # 81.98

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 6 1 0  
## Car 2 12 3  
## Public Transport 13 2 72  
##   
## Overall Statistics  
##   
## Accuracy : 0.8108   
## 95% CI : (0.7255, 0.8789)  
## No Information Rate : 0.6757   
## P-Value [Acc > NIR] : 0.001096   
##   
## Kappa : 0.5679   
##   
## Mcnemar's Test P-Value : 0.003614   
##   
## Statistics by Class:  
##   
## Class: 2Wheeler Class: Car Class: Public Transport  
## Sensitivity 0.28571 0.8000 0.9600  
## Specificity 0.98889 0.9479 0.5833  
## Pos Pred Value 0.85714 0.7059 0.8276  
## Neg Pred Value 0.85577 0.9681 0.8750  
## Precision 0.85714 0.7059 0.8276  
## Recall 0.28571 0.8000 0.9600  
## F1 0.42857 0.7500 0.8889  
## Prevalence 0.18919 0.1351 0.6757  
## Detection Rate 0.05405 0.1081 0.6486  
## Detection Prevalence 0.06306 0.1532 0.7838  
## Balanced Accuracy 0.63730 0.8740 0.7717

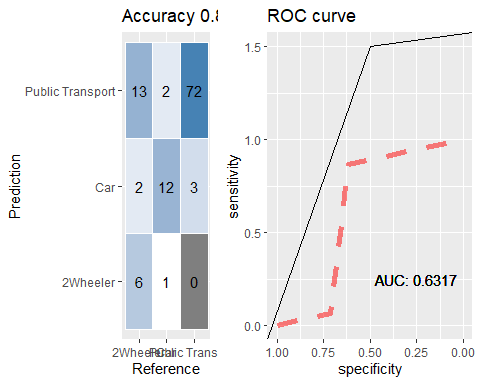
cfmROC(data\_pred,cartest[,9])

## Warning in roc.default(as.numeric(y\_true) - 1, as.numeric(y\_pred) - 1):  
## 'response' has more than two levels. Consider setting 'levels' explicitly or  
## using 'multiclass.roc' instead

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

## Setting direction: controls > cases

## Warning: Removed 1 row(s) containing missing values (geom\_path).



# with normalized data  
ctrl <- trainControl(method="repeatedcv",repeats = 3,classProbs=FALSE,summaryFunction = multiClassSummary)  
knnFit <- caret::train(Transport ~ ., data = cartrain, method = "knn", trControl = ctrl, preProcess = c("center","scale"), tuneLength = 20)

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :  
## There were missing values in resampled performance measures.

data\_pred <- predict(knnFit,cartest[,-9],type="raw")  
confusionMatrix(data\_pred,cartest[,9], positive = "a", mode="everything") # 84.68

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 3 1 1  
## Car 2 11 1  
## Public Transport 16 3 73  
##   
## Overall Statistics  
##   
## Accuracy : 0.7838   
## 95% CI : (0.6956, 0.8563)  
## No Information Rate : 0.6757   
## P-Value [Acc > NIR] : 0.008229   
##   
## Kappa : 0.4783   
##   
## Mcnemar's Test P-Value : 0.002225   
##   
## Statistics by Class:  
##   
## Class: 2Wheeler Class: Car Class: Public Transport  
## Sensitivity 0.14286 0.7333 0.9733  
## Specificity 0.97778 0.9688 0.4722  
## Pos Pred Value 0.60000 0.7857 0.7935  
## Neg Pred Value 0.83019 0.9588 0.8947  
## Precision 0.60000 0.7857 0.7935  
## Recall 0.14286 0.7333 0.9733  
## F1 0.23077 0.7586 0.8743  
## Prevalence 0.18919 0.1351 0.6757  
## Detection Rate 0.02703 0.0991 0.6577  
## Detection Prevalence 0.04505 0.1261 0.8288  
## Balanced Accuracy 0.56032 0.8510 0.7228

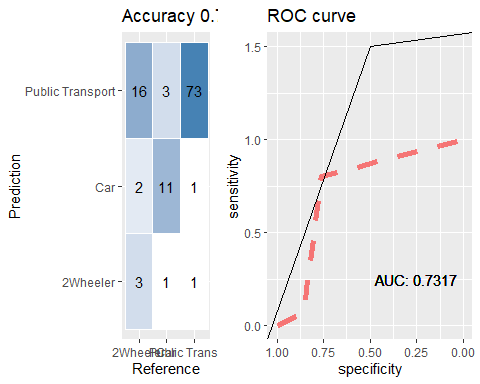
cfmROC(data\_pred,cartest[,9])

## Warning in roc.default(as.numeric(y\_true) - 1, as.numeric(y\_pred) - 1):  
## 'response' has more than two levels. Consider setting 'levels' explicitly or  
## using 'multiclass.roc' instead

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

## Setting direction: controls > cases

## Warning: Removed 1 row(s) containing missing values (geom\_path).



Logistic Regression

# Logistic Regression 1.1 without smote  
trControl <- trainControl(method = "cv", number = 10,verboseIter = FALSE,  
 summaryFunction = multiClassSummary)  
#trControl$sampling<- 'smote'  
fit\_glm = caret::train(  
 Transport ~ .,  
 data = cartrain,  
 method = "multinom",  
 trControl = trControl,  
 preProcess = c("center", "scale"),  
 trace = FALSE  
)

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :  
## There were missing values in resampled performance measures.

data\_pred <- predict(fit\_glm,cartest[,-9],type="raw")  
confusionMatrix(data\_pred,cartest[,9], positive = "a", mode="everything") # 84.68 accuracy increased

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 5 0 1  
## Car 2 15 6  
## Public Transport 14 0 68  
##   
## Overall Statistics  
##   
## Accuracy : 0.7928   
## 95% CI : (0.7055, 0.8639)  
## No Information Rate : 0.6757   
## P-Value [Acc > NIR] : 0.0044228   
##   
## Kappa : 0.5521   
##   
## Mcnemar's Test P-Value : 0.0002408   
##   
## Statistics by Class:  
##   
## Class: 2Wheeler Class: Car Class: Public Transport  
## Sensitivity 0.23810 1.0000 0.9067  
## Specificity 0.98889 0.9167 0.6111  
## Pos Pred Value 0.83333 0.6522 0.8293  
## Neg Pred Value 0.84762 1.0000 0.7586  
## Precision 0.83333 0.6522 0.8293  
## Recall 0.23810 1.0000 0.9067  
## F1 0.37037 0.7895 0.8662  
## Prevalence 0.18919 0.1351 0.6757  
## Detection Rate 0.04505 0.1351 0.6126  
## Detection Prevalence 0.05405 0.2072 0.7387  
## Balanced Accuracy 0.61349 0.9583 0.7589

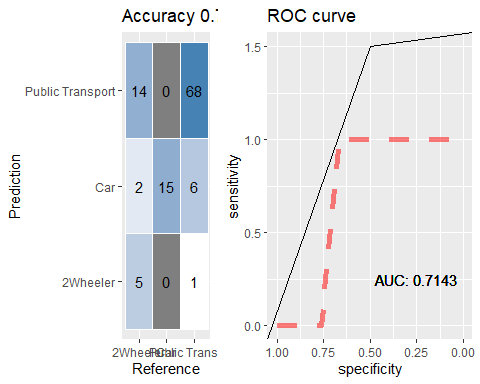
#plot   
cfmROC(data\_pred,cartest[,9])

## Warning in roc.default(as.numeric(y\_true) - 1, as.numeric(y\_pred) - 1):  
## 'response' has more than two levels. Consider setting 'levels' explicitly or  
## using 'multiclass.roc' instead

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

## Setting direction: controls > cases

## Warning: Removed 1 row(s) containing missing values (geom\_path).



# Logistic Regression 1.1 with smote  
trControl <- trainControl(method = "cv", number = 10,verboseIter = FALSE,  
 summaryFunction = multiClassSummary)  
trControl$sampling<- 'smote'  
fit\_glm = caret::train(  
 Transport ~ .,  
 data = cartrain,  
 method = "multinom",  
 trControl = trControl,  
 preProcess = c("center", "scale"),  
 trace = FALSE  
)

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

##   
## Attaching package: 'DMwR'

## The following object is masked from 'package:plyr':  
##   
## join

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :  
## There were missing values in resampled performance measures.

fit\_glm

## Penalized Multinomial Regression   
##   
## 333 samples  
## 8 predictor  
## 3 classes: '2Wheeler', 'Car', 'Public Transport'   
##   
## Pre-processing: centered (8), scaled (8)   
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 299, 300, 299, 299, 300, 300, ...   
## Addtional sampling using SMOTE prior to pre-processing  
##   
## Resampling results across tuning parameters:  
##   
## decay Accuracy Kappa Mean\_F1 Mean\_Sensitivity Mean\_Specificity  
## 0e+00 0.7695744 0.5078045 0.6957321 0.6912400 0.8242359   
## 1e-04 0.7531473 0.4635516 0.6851415 0.6643638 0.8066442   
## 1e-01 0.7953933 0.5605834 0.7163844 0.7191420 0.8416837   
## Mean\_Pos\_Pred\_Value Mean\_Neg\_Pred\_Value Mean\_Precision Mean\_Recall  
## 0.7164381 0.8539557 0.7164381 0.6912400   
## 0.6564562 0.8389998 0.6564562 0.6643638   
## 0.7500161 0.8751070 0.7500161 0.7191420   
## Mean\_Detection\_Rate Mean\_Balanced\_Accuracy  
## 0.2565248 0.7577380   
## 0.2510491 0.7355040   
## 0.2651311 0.7804129   
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was decay = 0.1.

data\_pred <- predict(fit\_glm,cartest[,-9],type="raw")  
confusionMatrix(data\_pred,cartest[,9], positive = "a", mode="everything") # 85.59 accuracy decreasd

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 3 0 1  
## Car 3 15 9  
## Public Transport 15 0 65  
##   
## Overall Statistics  
##   
## Accuracy : 0.7477   
## 95% CI : (0.6565, 0.8254)  
## No Information Rate : 0.6757   
## P-Value [Acc > NIR] : 0.0619   
##   
## Kappa : 0.4671   
##   
## Mcnemar's Test P-Value : 2.215e-05   
##   
## Statistics by Class:  
##   
## Class: 2Wheeler Class: Car Class: Public Transport  
## Sensitivity 0.14286 1.0000 0.8667  
## Specificity 0.98889 0.8750 0.5833  
## Pos Pred Value 0.75000 0.5556 0.8125  
## Neg Pred Value 0.83178 1.0000 0.6774  
## Precision 0.75000 0.5556 0.8125  
## Recall 0.14286 1.0000 0.8667  
## F1 0.24000 0.7143 0.8387  
## Prevalence 0.18919 0.1351 0.6757  
## Detection Rate 0.02703 0.1351 0.5856  
## Detection Prevalence 0.03604 0.2432 0.7207  
## Balanced Accuracy 0.56587 0.9375 0.7250

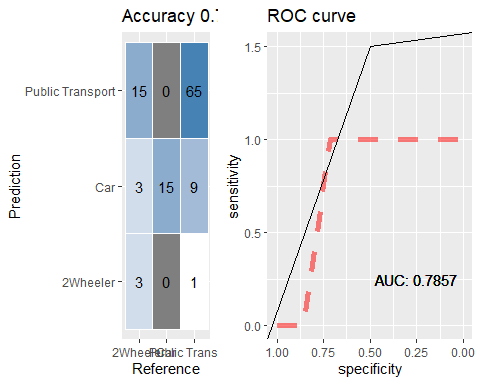
#plot   
cfmROC(data\_pred,cartest[,9])

## Warning in roc.default(as.numeric(y\_true) - 1, as.numeric(y\_pred) - 1):  
## 'response' has more than two levels. Consider setting 'levels' explicitly or  
## using 'multiclass.roc' instead

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

## Setting direction: controls > cases

## Warning: Removed 1 row(s) containing missing values (geom\_path).



# Logistic Regression 1.2 without smote removing age  
trControl <- trainControl(method = "cv", number = 10,verboseIter = FALSE,  
 summaryFunction = multiClassSummary)  
trControl$sampling<- 'smote'  
fit\_glm = caret::train(  
 Transport ~ .-Age,  
 data = cartrain,  
 method = "multinom",  
 trControl = trControl,  
 preProcess = c("center", "scale"),  
 trace = FALSE  
)

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :  
## There were missing values in resampled performance measures.

data\_pred <- predict(fit\_glm,cartest[,-9],type="raw")  
confusionMatrix(data\_pred,cartest[,9], positive = "a", mode="everything") # 81.98 accuracy decreasd

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 8 0 1  
## Car 2 14 5  
## Public Transport 11 1 69  
##   
## Overall Statistics  
##   
## Accuracy : 0.8198   
## 95% CI : (0.7355, 0.8863)  
## No Information Rate : 0.6757   
## P-Value [Acc > NIR] : 0.0005033   
##   
## Kappa : 0.6134   
##   
## Mcnemar's Test P-Value : 0.0046366   
##   
## Statistics by Class:  
##   
## Class: 2Wheeler Class: Car Class: Public Transport  
## Sensitivity 0.38095 0.9333 0.9200  
## Specificity 0.98889 0.9271 0.6667  
## Pos Pred Value 0.88889 0.6667 0.8519  
## Neg Pred Value 0.87255 0.9889 0.8000  
## Precision 0.88889 0.6667 0.8519  
## Recall 0.38095 0.9333 0.9200  
## F1 0.53333 0.7778 0.8846  
## Prevalence 0.18919 0.1351 0.6757  
## Detection Rate 0.07207 0.1261 0.6216  
## Detection Prevalence 0.08108 0.1892 0.7297  
## Balanced Accuracy 0.68492 0.9302 0.7933

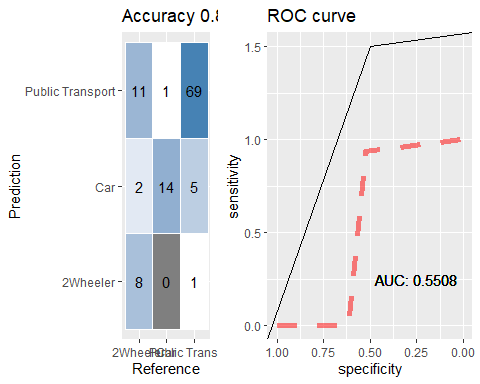
#plot   
cfmROC(data\_pred,cartest[,9])

## Warning in roc.default(as.numeric(y\_true) - 1, as.numeric(y\_pred) - 1):  
## 'response' has more than two levels. Consider setting 'levels' explicitly or  
## using 'multiclass.roc' instead

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

## Setting direction: controls > cases

## Warning: Removed 1 row(s) containing missing values (geom\_path).



# Logistic Regression 1.2 without smote removing Work.exp  
trControl <- trainControl(method = "cv", number = 10,verboseIter = FALSE,  
 summaryFunction = multiClassSummary)  
trControl$sampling<- 'smote'  
fit\_glm = caret::train(  
 Transport ~ .-Work.Exp-Age,  
 data = cartrain,  
 method = "multinom",  
 trControl = trControl,  
 preProcess = c("center", "scale"),  
 trace = FALSE  
)

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :  
## There were missing values in resampled performance measures.

data\_pred <- predict(fit\_glm,cartest[,-9],type="raw")  
confusionMatrix(data\_pred,cartest[,9], positive = "a", mode="everything") # 82.88 accuracy decreasd

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 7 0 4  
## Car 4 13 5  
## Public Transport 10 2 66  
##   
## Overall Statistics  
##   
## Accuracy : 0.7748   
## 95% CI : (0.6857, 0.8486)  
## No Information Rate : 0.6757   
## P-Value [Acc > NIR] : 0.01459   
##   
## Kappa : 0.5305   
##   
## Mcnemar's Test P-Value : 0.04906   
##   
## Statistics by Class:  
##   
## Class: 2Wheeler Class: Car Class: Public Transport  
## Sensitivity 0.33333 0.8667 0.8800  
## Specificity 0.95556 0.9062 0.6667  
## Pos Pred Value 0.63636 0.5909 0.8462  
## Neg Pred Value 0.86000 0.9775 0.7273  
## Precision 0.63636 0.5909 0.8462  
## Recall 0.33333 0.8667 0.8800  
## F1 0.43750 0.7027 0.8627  
## Prevalence 0.18919 0.1351 0.6757  
## Detection Rate 0.06306 0.1171 0.5946  
## Detection Prevalence 0.09910 0.1982 0.7027  
## Balanced Accuracy 0.64444 0.8865 0.7733

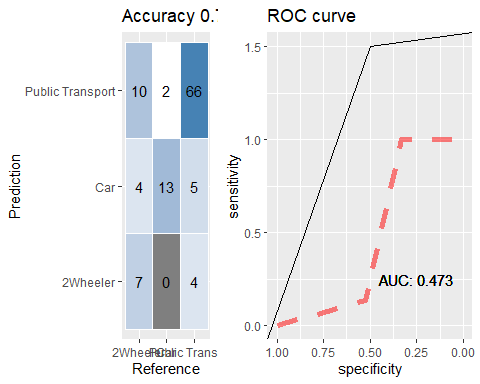
#plot   
cfmROC(data\_pred,cartest[,9])

## Warning in roc.default(as.numeric(y\_true) - 1, as.numeric(y\_pred) - 1):  
## 'response' has more than two levels. Consider setting 'levels' explicitly or  
## using 'multiclass.roc' instead

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

## Setting direction: controls < cases

## Warning: Removed 1 row(s) containing missing values (geom\_path).

 BAgging

#Bagging 1.1  
library(rpart)  
cntrl <- trainControl(method = "cv", number = 10)  
control = rpart.control(minsplit = 2, cp = 0)  
fit\_bag<- caret::train(Transport ~ ., data = cartrain, method = "treebag",nbagg= 200, trControl = cntrl)  
pred<- fit\_bag %>% predict(cartest[,-9])  
confusionMatrix(pred, cartest[,9]) # Accuracy % ovefitting

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 10 0 7  
## Car 2 13 4  
## Public Transport 9 2 64  
##   
## Overall Statistics  
##   
## Accuracy : 0.7838   
## 95% CI : (0.6956, 0.8563)  
## No Information Rate : 0.6757   
## P-Value [Acc > NIR] : 0.008229   
##   
## Kappa : 0.56   
##   
## Mcnemar's Test P-Value : 0.404653   
##   
## Statistics by Class:  
##   
## Class: 2Wheeler Class: Car Class: Public Transport  
## Sensitivity 0.47619 0.8667 0.8533  
## Specificity 0.92222 0.9375 0.6944  
## Pos Pred Value 0.58824 0.6842 0.8533  
## Neg Pred Value 0.88298 0.9783 0.6944  
## Prevalence 0.18919 0.1351 0.6757  
## Detection Rate 0.09009 0.1171 0.5766  
## Detection Prevalence 0.15315 0.1712 0.6757  
## Balanced Accuracy 0.69921 0.9021 0.7739

confusionMatrix(fit\_bag) #81.08 on trin

## Cross-Validated (10 fold) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 8.1 0.0 3.6  
## Car 0.3 12.0 1.8  
## Public Transport 10.2 1.8 62.2  
##   
## Accuracy (average) : 0.8228

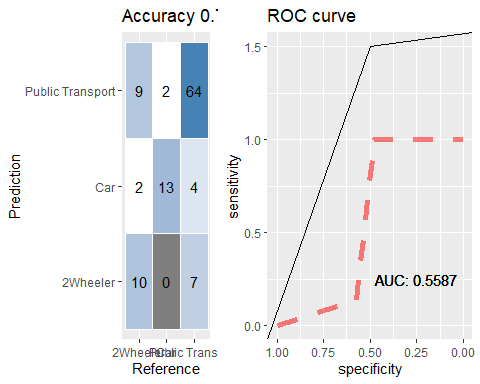
#plot   
cfmROC(pred,cartest[,9])

## Warning in roc.default(as.numeric(y\_true) - 1, as.numeric(y\_pred) - 1):  
## 'response' has more than two levels. Consider setting 'levels' explicitly or  
## using 'multiclass.roc' instead

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

## Setting direction: controls < cases

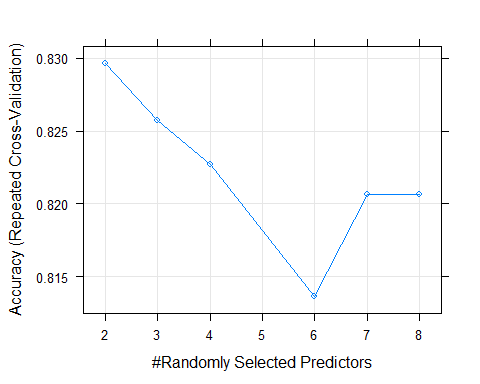
## Warning: Removed 1 row(s) containing missing values (geom\_path).



# Bagging Random Forest 1.2  
# Tune using caret  
# Random Search  
control <- trainControl(method="repeatedcv", number=10, repeats=3, search="random")  
set.seed(7)  
mtry <- sqrt(ncol(cartrain))  
rf\_random <- caret::train(Transport~., data=cartrain, method="rf", metric="Accuracy", tuneLength=15, trControl=control)  
print(rf\_random)

## Random Forest   
##   
## 333 samples  
## 8 predictor  
## 3 classes: '2Wheeler', 'Car', 'Public Transport'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 3 times)   
## Summary of sample sizes: 300, 300, 299, 300, 299, 299, ...   
## Resampling results across tuning parameters:  
##   
## mtry Accuracy Kappa   
## 2 0.8296651 0.6061516  
## 3 0.8257157 0.6075338  
## 4 0.8226835 0.6022571  
## 6 0.8136150 0.5854277  
## 7 0.8206560 0.6023820  
## 8 0.8206614 0.6022243  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was mtry = 2.

plot(rf\_random)



pred<- rf\_random %>% predict(cartest[,-9])  
confusionMatrix(pred, cartest[,9]) # Accuracy 81.08% ovefitting

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 9 0 4  
## Car 2 13 3  
## Public Transport 10 2 68  
##   
## Overall Statistics  
##   
## Accuracy : 0.8108   
## 95% CI : (0.7255, 0.8789)  
## No Information Rate : 0.6757   
## P-Value [Acc > NIR] : 0.001096   
##   
## Kappa : 0.5966   
##   
## Mcnemar's Test P-Value : 0.189320   
##   
## Statistics by Class:  
##   
## Class: 2Wheeler Class: Car Class: Public Transport  
## Sensitivity 0.42857 0.8667 0.9067  
## Specificity 0.95556 0.9479 0.6667  
## Pos Pred Value 0.69231 0.7222 0.8500  
## Neg Pred Value 0.87755 0.9785 0.7742  
## Prevalence 0.18919 0.1351 0.6757  
## Detection Rate 0.08108 0.1171 0.6126  
## Detection Prevalence 0.11712 0.1622 0.7207  
## Balanced Accuracy 0.69206 0.9073 0.7867

confusionMatrix(rf\_random) #81.98 on train

## Cross-Validated (10 fold, repeated 3 times) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 6.3 0.3 1.4  
## Car 0.4 11.9 1.4  
## Public Transport 11.9 1.6 64.8  
##   
## Accuracy (average) : 0.8298

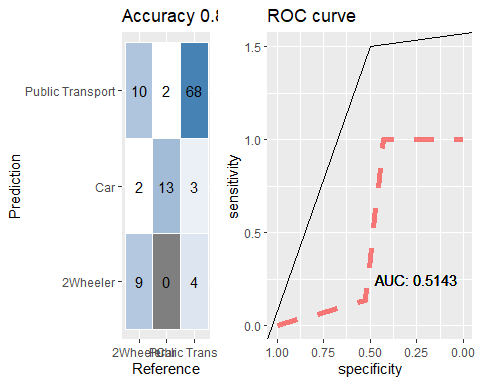
#plot   
cfmROC(pred,cartest[,9])

## Warning in roc.default(as.numeric(y\_true) - 1, as.numeric(y\_pred) - 1):  
## 'response' has more than two levels. Consider setting 'levels' explicitly or  
## using 'multiclass.roc' instead

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

## Setting direction: controls < cases

## Warning: Removed 1 row(s) containing missing values (geom\_path).

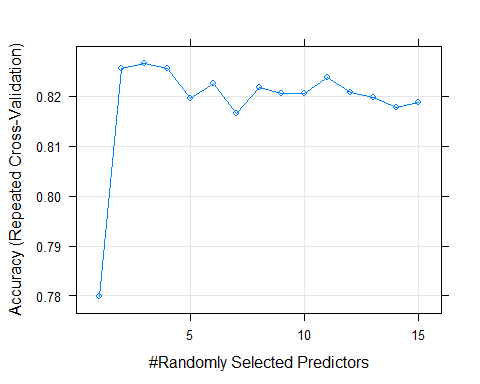


# Bagging Random Forest 1.3  
# Tune using caret  
# Grid Search  
control <- trainControl(method="repeatedcv", number=10, repeats=3, search="grid")  
set.seed(7)  
tunegrid <- expand.grid(.mtry=c(1:15))  
rf\_gridsearch <- caret::train(Transport~., data=cartrain, method="rf", metric="Accuracy",  
 tuneGrid=tunegrid, trControl=control)

print(rf\_gridsearch)

## Random Forest   
##   
## 333 samples  
## 8 predictor  
## 3 classes: '2Wheeler', 'Car', 'Public Transport'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 3 times)   
## Summary of sample sizes: 300, 300, 299, 300, 299, 299, ...   
## Resampling results across tuning parameters:  
##   
## mtry Accuracy Kappa   
## 1 0.7797954 0.4351525  
## 2 0.8256544 0.5981371  
## 3 0.8266961 0.6108113  
## 4 0.8256507 0.6094739  
## 5 0.8195288 0.5962470  
## 6 0.8226519 0.6060451  
## 7 0.8165282 0.5950916  
## 8 0.8216994 0.6041076  
## 9 0.8206299 0.6010880  
## 10 0.8206895 0.6031657  
## 11 0.8237215 0.6111501  
## 12 0.8207192 0.6030377  
## 13 0.8197685 0.5995986  
## 14 0.8176887 0.5963134  
## 15 0.8187287 0.5988751  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was mtry = 3.

plot(rf\_gridsearch)



pred<- rf\_random %>% predict(cartest[,-9])  
confusionMatrix(pred, cartest[,9]) # Accuracy 100% ovefitting

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 9 0 4  
## Car 2 13 3  
## Public Transport 10 2 68  
##   
## Overall Statistics  
##   
## Accuracy : 0.8108   
## 95% CI : (0.7255, 0.8789)  
## No Information Rate : 0.6757   
## P-Value [Acc > NIR] : 0.001096   
##   
## Kappa : 0.5966   
##   
## Mcnemar's Test P-Value : 0.189320   
##   
## Statistics by Class:  
##   
## Class: 2Wheeler Class: Car Class: Public Transport  
## Sensitivity 0.42857 0.8667 0.9067  
## Specificity 0.95556 0.9479 0.6667  
## Pos Pred Value 0.69231 0.7222 0.8500  
## Neg Pred Value 0.87755 0.9785 0.7742  
## Prevalence 0.18919 0.1351 0.6757  
## Detection Rate 0.08108 0.1171 0.6126  
## Detection Prevalence 0.11712 0.1622 0.7207  
## Balanced Accuracy 0.69206 0.9073 0.7867

confusionMatrix(rf\_gridsearch) #81.68 on train

## Cross-Validated (10 fold, repeated 3 times) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 7.5 0.3 2.8  
## Car 0.4 11.9 1.5  
## Public Transport 10.7 1.6 63.3  
##   
## Accuracy (average) : 0.8268

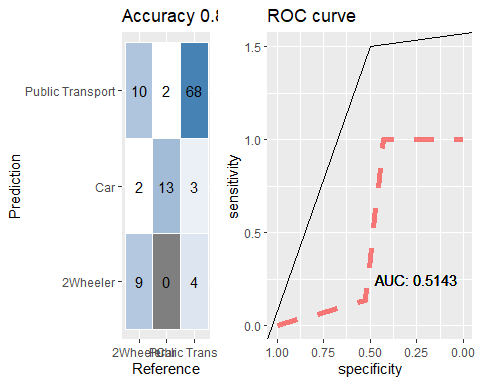
#plot   
cfmROC(pred,cartest[,9])

## Warning in roc.default(as.numeric(y\_true) - 1, as.numeric(y\_pred) - 1):  
## 'response' has more than two levels. Consider setting 'levels' explicitly or  
## using 'multiclass.roc' instead

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

## Setting direction: controls < cases

## Warning: Removed 1 row(s) containing missing values (geom\_path).



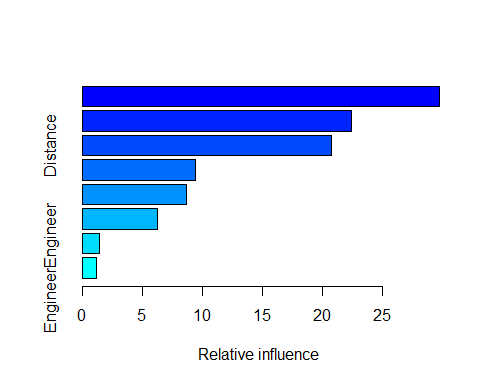
# Boosting 1.1  
Control <- trainControl( method = "repeatedcv", number = 10, repeats = 3)  
set.seed(825)  
gbmFit1 <- caret::train(Transport ~ ., data = cartrain,   
 method = "gbm",   
 trControl = Control,  
 verbose = FALSE)  
pred<- gbmFit1 %>% predict(cartest[,-9])  
confusionMatrix(pred, cartest[,9]) # Accuracy 91.89%

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 9 0 7  
## Car 1 13 2  
## Public Transport 11 2 66  
##   
## Overall Statistics  
##   
## Accuracy : 0.7928   
## 95% CI : (0.7055, 0.8639)  
## No Information Rate : 0.6757   
## P-Value [Acc > NIR] : 0.004423   
##   
## Kappa : 0.5613   
##   
## Mcnemar's Test P-Value : 0.595785   
##   
## Statistics by Class:  
##   
## Class: 2Wheeler Class: Car Class: Public Transport  
## Sensitivity 0.42857 0.8667 0.8800  
## Specificity 0.92222 0.9688 0.6389  
## Pos Pred Value 0.56250 0.8125 0.8354  
## Neg Pred Value 0.87368 0.9789 0.7188  
## Prevalence 0.18919 0.1351 0.6757  
## Detection Rate 0.08108 0.1171 0.5946  
## Detection Prevalence 0.14414 0.1441 0.7117  
## Balanced Accuracy 0.67540 0.9177 0.7594

confusionMatrix(gbmFit1) #81.08 on train

## Cross-Validated (10 fold, repeated 3 times) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction 2Wheeler Car Public Transport  
## 2Wheeler 7.6 0.0 6.5  
## Car 0.3 12.9 1.0  
## Public Transport 10.7 0.9 60.1  
##   
## Accuracy (average) : 0.8058

summary(gbmFit1)



## var rel.inf  
## Salary Salary 29.786892  
## Age Age 22.442780  
## Distance Distance 20.776778  
## Work.Exp Work.Exp 9.405650  
## licenseDL licenseDL 8.680116  
## GenderMale GenderMale 6.293596  
## MBAMBA MBAMBA 1.431116  
## EngineerEngineer EngineerEngineer 1.183072

gbmFit1$bestTune

## n.trees interaction.depth shrinkage n.minobsinnode  
## 8 100 3 0.1 10

#n.trees interaction.depth shrinkage n.minobsinnode  
#4 50 2 0.1 10  
#plot   
cfmROC(pred,cartest[,9])

## Warning in roc.default(as.numeric(y\_true) - 1, as.numeric(y\_pred) - 1):  
## 'response' has more than two levels. Consider setting 'levels' explicitly or  
## using 'multiclass.roc' instead

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

## Setting direction: controls > cases

## Warning: Removed 1 row(s) containing missing values (geom\_path).

