Assignment 05

Shubham Maheshwari - 8894405

2024-12-01

# 1. Preliminary Data Preparation

setwd('C:/PROG8435 Data Analytics/Assignment 5')  
  
df <- read.table("PROG8435-24F-Assign05.txt", sep = ",", header = TRUE)  
  
df <- as.data.frame(df)  
head(df)

## Prc Bed floor TotFloor Bath Sqft City Comp Dist  
## 1 1590 3 3 4 4 501 Blossomville Leaseflow 10.6  
## 2 2460 2 5 7 2 1275 Terranova Rentopia 1.4  
## 3 2630 1 4 6 2 982 Riverport Leaseflow 6.8  
## 4 2820 1 2 4 1 2418 Terranova Leaseflow 0.4  
## 5 2740 4 1 2 3 1655 Terranova Rentopia 5.3  
## 6 2370 3 5 5 2 1024 Riverport Leaseflow 0.5

# Rename All Variables to SM   
colnames(df) <- paste(colnames(df), "SM", sep = "\_")  
head(df)

## Prc\_SM Bed\_SM floor\_SM TotFloor\_SM Bath\_SM Sqft\_SM City\_SM Comp\_SM  
## 1 1590 3 3 4 4 501 Blossomville Leaseflow  
## 2 2460 2 5 7 2 1275 Terranova Rentopia  
## 3 2630 1 4 6 2 982 Riverport Leaseflow  
## 4 2820 1 2 4 1 2418 Terranova Leaseflow  
## 5 2740 4 1 2 3 1655 Terranova Rentopia  
## 6 2370 3 5 5 2 1024 Riverport Leaseflow  
## Dist\_SM  
## 1 10.6  
## 2 1.4  
## 3 6.8  
## 4 0.4  
## 5 5.3  
## 6 0.5

df <- as.data.frame(unclass(df), stringsAsFactors=TRUE)  
head(df)

## Prc\_SM Bed\_SM floor\_SM TotFloor\_SM Bath\_SM Sqft\_SM City\_SM Comp\_SM  
## 1 1590 3 3 4 4 501 Blossomville Leaseflow  
## 2 2460 2 5 7 2 1275 Terranova Rentopia  
## 3 2630 1 4 6 2 982 Riverport Leaseflow  
## 4 2820 1 2 4 1 2418 Terranova Leaseflow  
## 5 2740 4 1 2 3 1655 Terranova Rentopia  
## 6 2370 3 5 5 2 1024 Riverport Leaseflow  
## Dist\_SM  
## 1 10.6  
## 2 1.4  
## 3 6.8  
## 4 0.4  
## 5 5.3  
## 6 0.5

# Creating a new variable PC\_SM   
df$PC\_SM <- as.factor(ifelse(df$Prc\_SM < 2251,"L","H"))  
head(df)

## Prc\_SM Bed\_SM floor\_SM TotFloor\_SM Bath\_SM Sqft\_SM City\_SM Comp\_SM  
## 1 1590 3 3 4 4 501 Blossomville Leaseflow  
## 2 2460 2 5 7 2 1275 Terranova Rentopia  
## 3 2630 1 4 6 2 982 Riverport Leaseflow  
## 4 2820 1 2 4 1 2418 Terranova Leaseflow  
## 5 2740 4 1 2 3 1655 Terranova Rentopia  
## 6 2370 3 5 5 2 1024 Riverport Leaseflow  
## Dist\_SM PC\_SM  
## 1 10.6 L  
## 2 1.4 H  
## 3 6.8 H  
## 4 0.4 H  
## 5 5.3 H  
## 6 0.5 H

# Removing PRC Variable   
df <- df[,-c(1)]  
head(df)

## Bed\_SM floor\_SM TotFloor\_SM Bath\_SM Sqft\_SM City\_SM Comp\_SM Dist\_SM  
## 1 3 3 4 4 501 Blossomville Leaseflow 10.6  
## 2 2 5 7 2 1275 Terranova Rentopia 1.4  
## 3 1 4 6 2 982 Riverport Leaseflow 6.8  
## 4 1 2 4 1 2418 Terranova Leaseflow 0.4  
## 5 4 1 2 3 1655 Terranova Rentopia 5.3  
## 6 3 5 5 2 1024 Riverport Leaseflow 0.5  
## PC\_SM  
## 1 L  
## 2 H  
## 3 H  
## 4 H  
## 5 H  
## 6 H

summary(df)

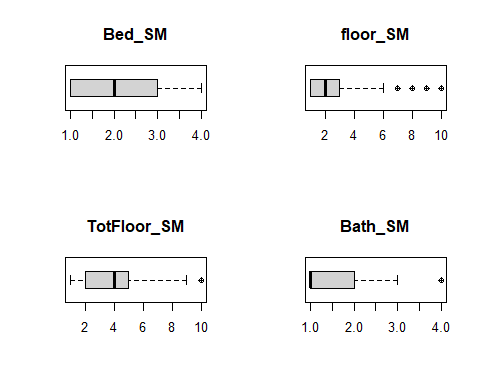
## Bed\_SM floor\_SM TotFloor\_SM Bath\_SM   
## Min. :1.000 Min. : 1.000 Min. : 1.000 Min. :1.000   
## 1st Qu.:1.000 1st Qu.: 1.000 1st Qu.: 2.000 1st Qu.:1.000   
## Median :2.000 Median : 2.000 Median : 4.000 Median :1.000   
## Mean :2.127 Mean : 2.272 Mean : 3.898 Mean :1.618   
## 3rd Qu.:3.000 3rd Qu.: 3.000 3rd Qu.: 5.000 3rd Qu.:2.000   
## Max. :4.000 Max. :10.000 Max. :10.000 Max. :4.000   
## Sqft\_SM City\_SM Comp\_SM Dist\_SM PC\_SM   
## Min. : 501.0 Blossomville:365 Leaseflow:362 Min. : 0.00 H:450   
## 1st Qu.: 781.5 Riverport :355 Rentopia :689 1st Qu.: 1.90 L:601   
## Median :1231.0 Terranova :331 Median : 3.50   
## Mean :1289.7 Mean : 4.11   
## 3rd Qu.:1645.0 3rd Qu.: 5.50   
## Max. :3254.0 Max. :21.00

table(df$PC\_gp)

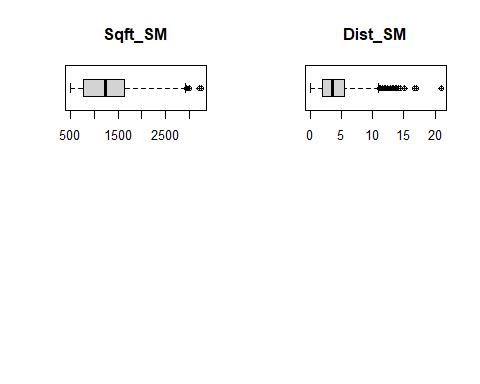
## < table of extent 0 >

# 2. Exploratory Analysis

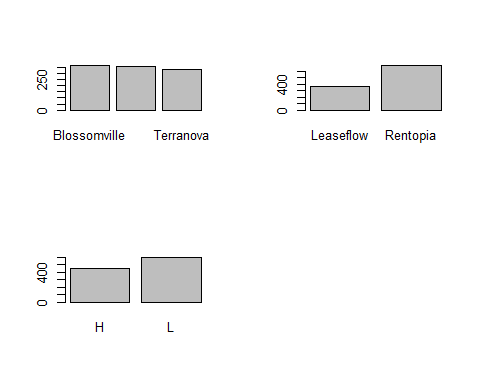
# Boxplots   
par(mfrow=c(2,2))  
for (i in 1:ncol(df)) {  
 if (is.numeric(df[,i])) {  
 boxplot(df[i], main=names(df)[i],  
 horizontal=TRUE, pch=10)  
 }  
}



par(mfrow=c(1,1))



# Barplots for Factor Columns  
par(mfrow=c(2,2))  
  
for(i in 1:ncol(df)) {  
 if(is.factor(df[,i])){  
 ct <- table(df[i])  
 barplot(ct, main=names(df[,i]))  
 }  
}  
  
par(mfrow=c(1,1))



***Observation:***   
There are clear outliers in several numeric variables such as (floor\_SM, Sqft\_SM, Dist\_SM). These might need further investigation to determine their validity apart from that the categorical data seems fairly balanced overall, except for the notable difference in the H and L categories.

hetcor(df$Bed\_SM, df$PC\_SM)

##   
## Two-Step Estimates  
##   
## Correlations/Type of Correlation:  
## df$Bed\_SM df.PC\_SM  
## df$Bed\_SM 1 Polyserial  
## df.PC\_SM 0.4096 1  
##   
## Standard Errors:  
## [1] "" "0.03177"  
##   
## n = 1051   
##   
## P-values for Tests of Bivariate Normality:  
## [1] "" "7.281e-158"

hetcor(df$floor\_SM, df$PC\_SM)

##   
## Two-Step Estimates  
##   
## Correlations/Type of Correlation:  
## df$floor\_SM df.PC\_SM  
## df$floor\_SM 1 Polyserial  
## df.PC\_SM -0.09431 1  
##   
## Standard Errors:  
## [1] "" "0.03819"  
##   
## n = 1051   
##   
## P-values for Tests of Bivariate Normality:  
## [1] "" "1.556e-48"

hetcor(df$TotFloor\_SM, df$PC\_SM)

##   
## Two-Step Estimates  
##   
## Correlations/Type of Correlation:  
## df$TotFloor\_SM df.PC\_SM  
## df$TotFloor\_SM 1 Polyserial  
## df.PC\_SM -0.04169 1  
##   
## Standard Errors:  
## [1] "" "0.03876"  
##   
## n = 1051   
##   
## P-values for Tests of Bivariate Normality:  
## [1] "" "7.366e-24"

hetcor(df$Bath\_SM, df$PC\_SM)

##   
## Two-Step Estimates  
##   
## Correlations/Type of Correlation:  
## df$Bath\_SM df.PC\_SM  
## df$Bath\_SM 1 Polyserial  
## df.PC\_SM -0.3601 1  
##   
## Standard Errors:  
## [1] "" "0.03325"  
##   
## n = 1051   
##   
## P-values for Tests of Bivariate Normality:  
## [1] "" "4.751e-227"

hetcor(df$Sqft\_SM, df$PC\_SM)

##   
## Two-Step Estimates  
##   
## Correlations/Type of Correlation:  
## df$Sqft\_SM df.PC\_SM  
## df$Sqft\_SM 1 Polyserial  
## df.PC\_SM -0.2325 1  
##   
## Standard Errors:  
## [1] "" "0.03648"  
##   
## n = 1051   
##   
## P-values for Tests of Bivariate Normality:  
## [1] "" "0.0003202"

hetcor(df$City\_SM, df$PC\_SM)

##   
## Two-Step Estimates  
##   
## Correlations/Type of Correlation:  
## df$City\_SM df.PC\_SM  
## df$City\_SM 1 Polychoric  
## df.PC\_SM 0.07736 1  
##   
## Standard Errors:  
## [1] "" "0.04351"  
##   
## n = 1051   
##   
## P-values for Tests of Bivariate Normality:  
## [1] "" "0.00000002072"

hetcor(df$Comp\_SM, df$PC\_SM)

##   
## Two-Step Estimates  
##   
## Correlations/Type of Correlation:  
## df$Comp\_SM df.PC\_SM  
## df$Comp\_SM 1 Polychoric  
## df.PC\_SM -0.02633 1  
##   
## Standard Errors:  
## [1] "" "0.05018"  
##   
## n = 1051

hetcor(df$Dist\_SM, df$PC\_SM)

##   
## Two-Step Estimates  
##   
## Correlations/Type of Correlation:  
## df$Dist\_SM df.PC\_SM  
## df$Dist\_SM 1 Polyserial  
## df.PC\_SM 0.2783 1  
##   
## Standard Errors:  
## [1] "" "0.03807"  
##   
## n = 1051   
##   
## P-values for Tests of Bivariate Normality:  
## [1] "" "0.0000000004374"

**Observation:**  
Distance and Price have strong positive correlation (0.2783) while Bath and Price have strong negative correlation ( -0.3601)

# Choosing the sampling rate  
sr <- 0.75  
  
  
n.row <- nrow(df)  
  
set.seed(4405)  
  
training.rows <- sample(1:n.row, sr\*n.row, replace=FALSE)  
  
train <- subset(df[training.rows,])  
  
test <- subset(df[-c(training.rows),])  
  
summary(train)

## Bed\_SM floor\_SM TotFloor\_SM Bath\_SM   
## Min. :1.000 Min. : 1.000 Min. : 1.000 Min. :1.000   
## 1st Qu.:1.000 1st Qu.: 1.000 1st Qu.: 2.000 1st Qu.:1.000   
## Median :2.000 Median : 2.000 Median : 4.000 Median :1.000   
## Mean :2.124 Mean : 2.272 Mean : 3.906 Mean :1.604   
## 3rd Qu.:3.000 3rd Qu.: 3.000 3rd Qu.: 5.000 3rd Qu.:2.000   
## Max. :4.000 Max. :10.000 Max. :10.000 Max. :4.000   
## Sqft\_SM City\_SM Comp\_SM Dist\_SM PC\_SM   
## Min. : 501.0 Blossomville:266 Leaseflow:270 Min. : 0.000 H:336   
## 1st Qu.: 764.2 Riverport :253 Rentopia :518 1st Qu.: 2.000 L:452   
## Median :1206.5 Terranova :269 Median : 3.500   
## Mean :1273.6 Mean : 4.128   
## 3rd Qu.:1634.5 3rd Qu.: 5.500   
## Max. :3254.0 Max. :21.000

summary(test)

## Bed\_SM floor\_SM TotFloor\_SM Bath\_SM   
## Min. :1.000 Min. :1.000 Min. : 1.000 Min. :1.000   
## 1st Qu.:1.000 1st Qu.:1.000 1st Qu.: 2.000 1st Qu.:1.000   
## Median :2.000 Median :2.000 Median : 4.000 Median :1.000   
## Mean :2.137 Mean :2.274 Mean : 3.875 Mean :1.662   
## 3rd Qu.:3.000 3rd Qu.:3.000 3rd Qu.: 5.000 3rd Qu.:2.000   
## Max. :4.000 Max. :9.000 Max. :10.000 Max. :4.000   
## Sqft\_SM City\_SM Comp\_SM Dist\_SM PC\_SM   
## Min. : 501.0 Blossomville: 99 Leaseflow: 92 Min. : 0.200 H:114   
## 1st Qu.: 905.5 Riverport :102 Rentopia :171 1st Qu.: 1.900 L:149   
## Median :1310.0 Terranova : 62 Median : 3.300   
## Mean :1338.1 Mean : 4.055   
## 3rd Qu.:1707.5 3rd Qu.: 5.400   
## Max. :2861.0 Max. :16.700

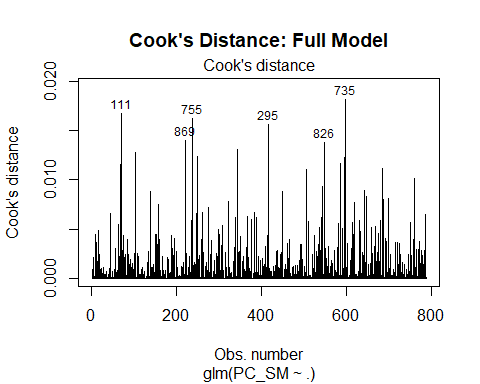
# 3. Model Development

### 1. Full Model

glm.full <- glm(PC\_SM ~ ., family = "binomial", data = train)  
  
# Summarize the model  
summary(glm.full)

##   
## Call:  
## glm(formula = PC\_SM ~ ., family = "binomial", data = train)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.7464780 0.4065176 1.836 0.0663 .   
## Bed\_SM 0.8637998 0.0875599 9.865 < 2e-16 \*\*\*  
## floor\_SM -0.1613477 0.0731850 -2.205 0.0275 \*   
## TotFloor\_SM 0.0734090 0.0605194 1.213 0.2251   
## Bath\_SM -0.9083733 0.1045431 -8.689 < 2e-16 \*\*\*  
## Sqft\_SM -0.0010931 0.0001687 -6.478 0.000000000092758 \*\*\*  
## City\_SMRiverport -1.0919106 0.2249409 -4.854 0.000001208672198 \*\*\*  
## City\_SMTerranova 0.5079386 0.2203052 2.306 0.0211 \*   
## Comp\_SMRentopia -0.2258041 0.1885548 -1.198 0.2311   
## Dist\_SM 0.2703968 0.0378800 7.138 0.000000000000945 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1075.26 on 787 degrees of freedom  
## Residual deviance: 759.93 on 778 degrees of freedom  
## AIC: 779.93  
##   
## Number of Fisher Scoring iterations: 5

# Plot Cook's distance for influential points  
plot(glm.full, which = 4, id.n = 6, main = "Cook's Distance: Full Model")



**Observations:**

Fisher iterations - 5

AIC - 779.93

Residual Deviance - 759.93

z-values - 7/10 variables are significant however it fails at the intercept

Parameter Co-Efficients - Some variables have positive correlation such as Bedroom and Distance , while others are negatively correlated.

There are some influential points but nothing of great significance as they are way under the 0.5 mark.

### 2. Backward Model

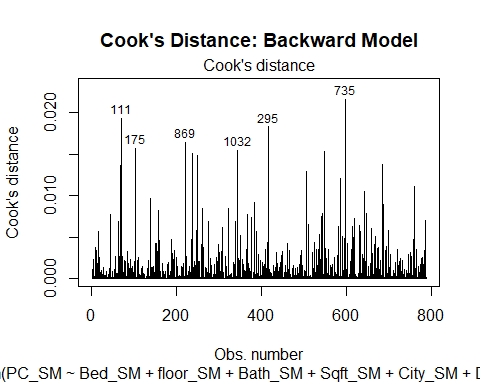
# Perform backward selection  
glm.back <- step(glm.full, direction = "backward", trace = TRUE)

## Start: AIC=779.93  
## PC\_SM ~ Bed\_SM + floor\_SM + TotFloor\_SM + Bath\_SM + Sqft\_SM +   
## City\_SM + Comp\_SM + Dist\_SM  
##   
## Df Deviance AIC  
## - Comp\_SM 1 761.37 779.37  
## - TotFloor\_SM 1 761.41 779.41  
## <none> 759.93 779.93  
## - floor\_SM 1 764.85 782.85  
## - Sqft\_SM 1 806.32 824.32  
## - City\_SM 2 813.04 829.04  
## - Dist\_SM 1 823.02 841.02  
## - Bath\_SM 1 853.83 871.83  
## - Bed\_SM 1 880.92 898.92  
##   
## Step: AIC=779.37  
## PC\_SM ~ Bed\_SM + floor\_SM + TotFloor\_SM + Bath\_SM + Sqft\_SM +   
## City\_SM + Dist\_SM  
##   
## Df Deviance AIC  
## - TotFloor\_SM 1 762.82 778.82  
## <none> 761.37 779.37  
## - floor\_SM 1 766.04 782.04  
## - Sqft\_SM 1 807.83 823.83  
## - City\_SM 2 815.11 829.11  
## - Dist\_SM 1 824.34 840.34  
## - Bath\_SM 1 855.73 871.73  
## - Bed\_SM 1 881.85 897.85  
##   
## Step: AIC=778.82  
## PC\_SM ~ Bed\_SM + floor\_SM + Bath\_SM + Sqft\_SM + City\_SM + Dist\_SM  
##   
## Df Deviance AIC  
## <none> 762.82 778.82  
## - floor\_SM 1 766.22 780.22  
## - Sqft\_SM 1 809.16 823.16  
## - City\_SM 2 816.75 828.75  
## - Dist\_SM 1 825.92 839.92  
## - Bath\_SM 1 856.85 870.85  
## - Bed\_SM 1 882.63 896.63

# Display the summary of the final model  
summary(glm.back)

##   
## Call:  
## glm(formula = PC\_SM ~ Bed\_SM + floor\_SM + Bath\_SM + Sqft\_SM +   
## City\_SM + Dist\_SM, family = "binomial", data = train)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.7493667 0.3669944 2.042 0.0412 \*   
## Bed\_SM 0.8569534 0.0871535 9.833 < 2e-16 \*\*\*  
## floor\_SM -0.0951892 0.0514932 -1.849 0.0645 .   
## Bath\_SM -0.9082320 0.1044389 -8.696 < 2e-16 \*\*\*  
## Sqft\_SM -0.0010896 0.0001681 -6.481 0.000000000090843 \*\*\*  
## City\_SMRiverport -1.1039158 0.2243257 -4.921 0.000000860849669 \*\*\*  
## City\_SMTerranova 0.5045146 0.2195030 2.298 0.0215 \*   
## Dist\_SM 0.2704390 0.0378840 7.139 0.000000000000943 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1075.26 on 787 degrees of freedom  
## Residual deviance: 762.82 on 780 degrees of freedom  
## AIC: 778.82  
##   
## Number of Fisher Scoring iterations: 5

# Plot Cook's distance for the backward selection model  
plot(glm.back, which = 4, id.n = 6, main = "Cook's Distance: Backward Model")



**Observations:**

Fisher iterations - 5

AIC - 778.82

Residual Deviance - 762.82

z-values - 6/8 variables passes the hypothesis

Parameter Co-Efficients - Bed and Distance are positively correlated, while other variables are not. We could see city Terranova positively correlated as well which will require further investigation.

### 4. Based on your preceding analysis, recommend which model should be selected and explain why.

From the above analysis we can conclude that the Backward Model is better than the Full Model

Both the models have no issues fitting the models, as showcased by fischer of 5

Back model has Lower AIC - 779.82 vs 779.93

Residual is slightly higher than that of full model, suggesting minimal loss and also due to having less variables which is explanatory

Removal of non significant values such as totfloor where the full model gave a positive correlation which was not reciprocated by the floor, indicating errors in the model.

# Part- B

## 1. Logistic Regression - Stepwise

start\_time <- Sys.time()  
step.model <- step(glm.full, direction = "both", trace = TRUE)

## Start: AIC=779.93  
## PC\_SM ~ Bed\_SM + floor\_SM + TotFloor\_SM + Bath\_SM + Sqft\_SM +   
## City\_SM + Comp\_SM + Dist\_SM  
##   
## Df Deviance AIC  
## - Comp\_SM 1 761.37 779.37  
## - TotFloor\_SM 1 761.41 779.41  
## <none> 759.93 779.93  
## - floor\_SM 1 764.85 782.85  
## - Sqft\_SM 1 806.32 824.32  
## - City\_SM 2 813.04 829.04  
## - Dist\_SM 1 823.02 841.02  
## - Bath\_SM 1 853.83 871.83  
## - Bed\_SM 1 880.92 898.92  
##   
## Step: AIC=779.37  
## PC\_SM ~ Bed\_SM + floor\_SM + TotFloor\_SM + Bath\_SM + Sqft\_SM +   
## City\_SM + Dist\_SM  
##   
## Df Deviance AIC  
## - TotFloor\_SM 1 762.82 778.82  
## <none> 761.37 779.37  
## + Comp\_SM 1 759.93 779.93  
## - floor\_SM 1 766.04 782.04  
## - Sqft\_SM 1 807.83 823.83  
## - City\_SM 2 815.11 829.11  
## - Dist\_SM 1 824.34 840.34  
## - Bath\_SM 1 855.73 871.73  
## - Bed\_SM 1 881.85 897.85  
##   
## Step: AIC=778.82  
## PC\_SM ~ Bed\_SM + floor\_SM + Bath\_SM + Sqft\_SM + City\_SM + Dist\_SM  
##   
## Df Deviance AIC  
## <none> 762.82 778.82  
## + TotFloor\_SM 1 761.37 779.37  
## + Comp\_SM 1 761.41 779.41  
## - floor\_SM 1 766.22 780.22  
## - Sqft\_SM 1 809.16 823.16  
## - City\_SM 2 816.75 828.75  
## - Dist\_SM 1 825.92 839.92  
## - Bath\_SM 1 856.85 870.85  
## - Bed\_SM 1 882.63 896.63

end\_time <- Sys.time()  
  
Time <- end\_time - start\_time  
  
summary(step.model)

##   
## Call:  
## glm(formula = PC\_SM ~ Bed\_SM + floor\_SM + Bath\_SM + Sqft\_SM +   
## City\_SM + Dist\_SM, family = "binomial", data = train)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.7493667 0.3669944 2.042 0.0412 \*   
## Bed\_SM 0.8569534 0.0871535 9.833 < 2e-16 \*\*\*  
## floor\_SM -0.0951892 0.0514932 -1.849 0.0645 .   
## Bath\_SM -0.9082320 0.1044389 -8.696 < 2e-16 \*\*\*  
## Sqft\_SM -0.0010896 0.0001681 -6.481 0.000000000090843 \*\*\*  
## City\_SMRiverport -1.1039158 0.2243257 -4.921 0.000000860849669 \*\*\*  
## City\_SMTerranova 0.5045146 0.2195030 2.298 0.0215 \*   
## Dist\_SM 0.2704390 0.0378840 7.139 0.000000000000943 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1075.26 on 787 degrees of freedom  
## Residual deviance: 762.82 on 780 degrees of freedom  
## AIC: 778.82  
##   
## Number of Fisher Scoring iterations: 5

Time

## Time difference of 0.1581888 secs

# Predict probabilities for training data  
response\_tr <- predict(step.model, newdata = train, type = "response")  
  
# Classify predictions based on the threshold of 0.5  
class\_tr <- ifelse(response\_tr > 0.5, "H", "L")  
  
# Create confusion matrix for training data  
conf\_tr <- table(train$PC\_SM, class\_tr,  
 dnn = list("Actual", "Predicted"))  
conf\_tr

## Predicted  
## Actual H L  
## H 110 226  
## L 372 80

# Calculate accuracy for training data  
accuracy\_tr <- (conf\_tr[2, 2] + conf\_tr[1, 1]) / sum(conf\_tr)  
  
# Display rounded accuracy  
round(accuracy\_tr, 3)

## [1] 0.241

# Predict probabilities for testing data  
response\_te <- predict(step.model, newdata = test, type = "response")  
  
# Classify predictions based on the threshold of 0.5  
class\_te <- ifelse(response\_te > 0.5, "H", "L")  
  
# Create confusion matrix for testing data  
conf\_te <- table(test$PC\_SM, class\_te,  
 dnn = list("Actual", "Predicted"))  
conf\_te

## Predicted  
## Actual H L  
## H 29 85  
## L 112 37

# Calculate accuracy for testing data  
accuracy\_te <- (conf\_te[2, 2] + conf\_te[1, 1]) / sum(conf\_te)  
  
# Display rounded accuracy  
round(accuracy\_te, 3)

## [1] 0.251

## 2. Naïve-Bayes Classification

# Recording the start time  
start\_tm <- Sys.time()  
  
# Fit the Naive Bayes model  
nb.mod <- NaiveBayes(PC\_SM ~ ., data = train, na.action = na.omit)  
  
# Record end time  
end\_tm <- Sys.time()  
  
# Calculate time taken for model fitting  
time\_taken <- end\_tm - start\_tm  
time\_taken

## Time difference of 0.006692171 secs

# Train Data  
pred\_NB <- predict(nb.mod, newdata=train)  
  
Conf\_NB <- table(Actual=train$PC\_SM , Predicted=pred\_NB$class)  
Conf\_NB

## Predicted  
## Actual H L  
## H 215 121  
## L 74 378

Accuracy <- (Conf\_NB[1,1] + Conf\_NB[2,2])/sum(Conf\_NB)  
  
round(Accuracy,4)

## [1] 0.7525

# Test Data  
  
pred\_NB <- predict(nb.mod, newdata=test)  
  
Conf\_NB <- table(Actual=test$PC\_SM , Predicted=pred\_NB$class)  
Conf\_NB

## Predicted  
## Actual H L  
## H 79 35  
## L 34 115

Accuracy <- (Conf\_NB[1,1] + Conf\_NB[2,2]/sum(Conf\_NB))  
  
round(Accuracy,4)

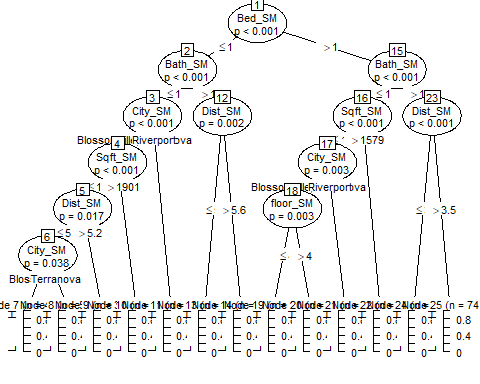
## [1] 79.4373

## 3. Recursive Partitioning Analysis

start\_time <- Sys.time()  
  
RP.mod <- ctree(PC\_SM ~ ., data= train)  
  
end\_time <- Sys.time()  
   
Time <- end\_time - start\_time  
Time

## Time difference of 0.08562088 secs

plot(RP.mod, gp=gpar(fontsize=8))



RP.mod

##   
## Model formula:  
## PC\_SM ~ Bed\_SM + floor\_SM + TotFloor\_SM + Bath\_SM + Sqft\_SM +   
## City\_SM + Comp\_SM + Dist\_SM  
##   
## Fitted party:  
## [1] root  
## | [2] Bed\_SM <= 1  
## | | [3] Bath\_SM <= 1  
## | | | [4] City\_SM in Blossomville, Terranova  
## | | | | [5] Sqft\_SM <= 1901  
## | | | | | [6] Dist\_SM <= 5.2  
## | | | | | | [7] City\_SM in Blossomville: H (n = 45, err = 44.4%)  
## | | | | | | [8] City\_SM in Terranova: L (n = 53, err = 28.3%)  
## | | | | | [9] Dist\_SM > 5.2: L (n = 32, err = 6.2%)  
## | | | | [10] Sqft\_SM > 1901: H (n = 30, err = 26.7%)  
## | | | [11] City\_SM in Riverport: H (n = 61, err = 27.9%)  
## | | [12] Bath\_SM > 1  
## | | | [13] Dist\_SM <= 5.6: H (n = 96, err = 10.4%)  
## | | | [14] Dist\_SM > 5.6: H (n = 34, err = 41.2%)  
## | [15] Bed\_SM > 1  
## | | [16] Bath\_SM <= 1  
## | | | [17] Sqft\_SM <= 1579  
## | | | | [18] City\_SM in Blossomville, Terranova  
## | | | | | [19] floor\_SM <= 4: L (n = 129, err = 2.3%)  
## | | | | | [20] floor\_SM > 4: L (n = 14, err = 35.7%)  
## | | | | [21] City\_SM in Riverport: L (n = 66, err = 24.2%)  
## | | | [22] Sqft\_SM > 1579: L (n = 91, err = 38.5%)  
## | | [23] Bath\_SM > 1  
## | | | [24] Dist\_SM <= 3.5: H (n = 63, err = 33.3%)  
## | | | [25] Dist\_SM > 3.5: L (n = 74, err = 28.4%)  
##   
## Number of inner nodes: 12  
## Number of terminal nodes: 13

# Predict on training data  
pred.RP <- predict(RP.mod, newdata= train)  
  
# Confusion matrix  
Conf\_RP <- table(Actual = train$PC\_SM, Predicted = pred.RP)  
  
# Print confusion matrix  
Conf\_RP

## Predicted  
## Actual H L  
## H 239 97  
## L 90 362

# Calculate accuracy for training data   
Accuracy <- (Conf\_RP[1,1] + Conf\_RP[2,2]) / sum(Conf\_RP)  
round(Accuracy, 4)

## [1] 0.7627

# Predict on testing data  
pred.RP <- predict(RP.mod, newdata= test)  
  
# Confusion matrix  
Conf\_RP <- table(Actual = test$PC\_SM, Predicted = pred.RP)  
  
# Print confusion matrix  
Conf\_RP

## Predicted  
## Actual H L  
## H 78 36  
## L 44 105

# Calculate accuracy for testing data   
Accuracy <- (Conf\_RP[1,1] + Conf\_RP[2,2]) / sum(Conf\_RP)  
round(Accuracy, 4)

## [1] 0.6958

## 4.Neural Network Fitting

# Install and load the nnet package  
if (!require("nnet")) {  
 install.packages("nnet")  
}

## Loading required package: nnet

library("nnet")  
  
# Record start time  
start\_tm <- Sys.time()  
  
# Fit the neural network model  
nn.mod <- nnet(  
 PC\_SM ~ ., # Response variable and predictors  
 data = train, # Dataset  
 size = 4, # Number of nodes in the hidden layer  
 rang = 0.0001, # Initial random weights range  
 maxit = 1200, # Maximum number of iterations  
 trace = FALSE # Suppress detailed trace output  
)  
  
# Record end time  
end\_tm <- Sys.time()  
  
# Calculate time taken for model fitting  
time <- end\_tm - start\_tm  
time

## Time difference of 0.060112 secs

nn.mod

## a 9-4-1 network with 45 weights  
## inputs: Bed\_SM floor\_SM TotFloor\_SM Bath\_SM Sqft\_SM City\_SMRiverport City\_SMTerranova Comp\_SMRentopia Dist\_SM   
## output(s): PC\_SM   
## options were - entropy fitting

summary(nn.mod)

## a 9-4-1 network with 45 weights  
## options were - entropy fitting   
## b->h1 i1->h1 i2->h1 i3->h1 i4->h1 i5->h1 i6->h1 i7->h1   
## 0.00 0.00 0.00 0.00 0.00 -0.04 0.00 0.00   
## i8->h1 i9->h1   
## 0.00 0.00   
## b->h2 i1->h2 i2->h2 i3->h2 i4->h2 i5->h2 i6->h2 i7->h2   
## 0.00 0.00 0.00 0.00 0.00 0.07 0.00 0.00   
## i8->h2 i9->h2   
## 0.00 0.00   
## b->h3 i1->h3 i2->h3 i3->h3 i4->h3 i5->h3 i6->h3 i7->h3   
## -481.99 1515.78 3564.80 398.45 -772.59 -3.41 -1485.28 -113.70   
## i8->h3 i9->h3   
## -1951.04 386.23   
## b->h4 i1->h4 i2->h4 i3->h4 i4->h4 i5->h4 i6->h4 i7->h4   
## 0.00 0.00 0.00 0.00 0.00 0.03 0.00 0.00   
## i8->h4 i9->h4   
## 0.00 0.00   
## b->o h1->o h2->o h3->o h4->o   
## -0.67 0.07 -0.76 2.82 -0.77

nn.mod$wts

## [1] 0.00002117856 0.00107111817 -0.00021931569 0.00004010136  
## [5] -0.00073038153 -0.04326680220 -0.00020079291 0.00022890363  
## [9] 0.00001728673 0.00171545987 -0.00006648911 -0.00086527741  
## [13] 0.00004488806 -0.00007699143 0.00025277547 0.06739214735  
## [17] 0.00009094115 -0.00014728294 0.00005011235 -0.00117195191  
## [21] -481.98791086054 1515.78354175498 3564.80071465022 398.45403241885  
## [25] -772.58819036037 -3.40552491183 -1485.27782063718 -113.70424230677  
## [29] -1951.04026410999 386.22736143107 0.00001313792 0.00013789595  
## [33] 0.00002721834 -0.00005825219 -0.00001542321 0.03076401896  
## [37] -0.00007761934 0.00015091648 0.00002431594 0.00040409096  
## [41] -0.66756417875 0.07363176256 -0.76305677047 2.81820935023  
## [45] -0.76660343321

# Predict on training data using the neural network mode  
pred.nn <- predict(nn.mod, newdata = train, type = "class")  
  
# Confusion matrix  
Conf\_NN <- table(Actual = train$PC\_SM, Predicted = pred.nn)  
  
# Print confusion matrix  
Conf\_NN

## Predicted  
## Actual H L  
## H 99 237  
## L 11 441

# Calculate accuracy  
Accuracy <- (Conf\_NN[1,1] + Conf\_NN[2,2]) / sum(Conf\_NN)  
round(Accuracy, 4)

## [1] 0.6853

# Predict on testing data using the neural network mode  
pred.nn <- predict(nn.mod, newdata = test, type = "class")  
  
  
Conf\_NN <- table(Actual = test$PC\_SM, Predicted = pred.nn)  
  
#Print confusion matrix  
Conf\_NN

## Predicted  
## Actual H L  
## H 35 79  
## L 17 132

Accuracy<- (Conf\_NN[1,1] + Conf\_NN[2,2]) / sum(Conf\_NN)  
round(Accuracy, 4)

## [1] 0.635

## 5. Compare All Classifiers

### 1.Which classifier is most accurate?

Logistic Regression : 25.1% Naive Bayes : 79.44% Recursive Partition: 69.58% Neural Network: 63.5%

Naïve Bayes is the most accurate classifier with a testing accuracy of 79.44%, outperforming all other models.

### 2.Which classifier seems most consistent (think train and test)?

We measured consistency by taking the difference between the training and testing accuracy, these were the results:

Logistic Regression : ~ 1% Naive Bayes : ~ 4.2% Recursive Partition: ~ 6.7% Neural Network: ~ 5.03%

Naïve Bayes is the most consistent classifier, with only a 4.2% difference between training and testing accuracy.

### 3.Which classifier is most suitable when processing speed is most important?

Here are the results of different models

Logistic Regression : 0.2357 seconds Naive Bayes: 0.0472 seconds Recursive Partioning: 0.1705 seconds Neural Network: 0.1065 seconds

Naïve Bayes is the most suitable when processing speed, as it trained in just 0.0472 seconds.

### 4.Which classifier minimizes false positives?

Here are the results of different models

Logistic Regression : 112 Naive Bayes : 34 Recursive Partioning: 44 Neural Network: 17

Neural Network minimizes false positives, with only 17 false positives in the testing set

### 5.In your opinion, which classifier is best overall? Make sure you state why.

In my opinion Naiive Bayes is the best overall classifier for the following reasons:

Highest Accuracy: 79.44%. Most Consistent: Smallest train-test gap (4.2%). Fastest Training: 0.0472 seconds, suitable for scenarios requiring quick results. Balanced Performance: While it does not minimize false positives, its overall performance (accuracy and consistency) outweighs this limitation.

# References

All the code was taken from the course material provided by the professor in class and Econestoga Learning.

Would like to mention Tony and Shubham for their help in clearing my doubts.