# DIABETES PREDICTION

Langat Erick

## Contents

### STEP-1 : Importing Diabetes Data

```
require(tidyverse)
diabetes <- read.csv("D:/DATASETS/diabetes.csv")
head(diabetes)
```

```
##   Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 1           6     148            72            35       0 33.6
## 2           1      85            66            29       0 26.6
## 3           8     183            64             0       0 23.3
## 4           1      89            66            23      94 28.1
## 5           0     137            40            35     168 43.1
## 6           5     116            74             0       0 25.6
##   DiabetesPedigreeFunction Age Outcome
## 1                    0.627  50       1
## 2                    0.351  31       0
## 3                    0.672  32       1
## 4                    0.167  21       0
## 5                    2.288  33       1
## 6                    0.201  30       0
```

## STEP-2 : Exploratory Data Analysis on Diabetes Data

```
#structure
str(diabetes)
```

```
## 'data.frame':    768 obs. of  9 variables:
##  $ Pregnancies             : int  6 1 8 1 0 5 3 10 2 8 ...
##  $ Glucose                 : int  148 85 183 89 137 116 78 115 197 125 ...
##  $ BloodPressure           : int  72 66 64 66 40 74 50 0 70 96 ...
##  $ SkinThickness           : int  35 29 0 23 35 0 32 0 45 0 ...
##  $ Insulin                 : int  0 0 0 94 168 0 88 0 543 0 ...
##  $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
##  $ Age                     : int  50 31 32 21 33 30 26 29 53 54 ...
##  $ Outcome                 : int  1 0 1 0 1 0 1 0 1 1 ...
```

```r
sum(is.na(diabetes))
```

```
## [1] 0
```

```r
colSums(is.na(diabetes)) %>% as.data.frame()
```

```
##                               .
## Pregnancies                   0
## Glucose                       0
## BloodPressure                 0
## SkinThickness                 0
## Insulin                       0
## BMI                           0
## DiabetesPedigreeFunction      0
## Age                           0
## Outcome                       0
```

```r
colnames(diabetes) %>% as.data.frame()
```

```
##                               .
## 1                   Pregnancies
## 2                       Glucose
## 3                 BloodPressure
## 4                 SkinThickness
## 5                       Insulin
## 6                           BMI
## 7 DiabetesPedigreeFunction
## 8                           Age
## 9                       Outcome
```

```r
require(psych)
describe(diabetes)
```

```
##                           vars   n   mean     sd median trimmed   mad   min
## Pregnancies                  1 768   3.85   3.37   3.00    3.46  2.97  0.00
## Glucose                      2 768 120.89  31.97 117.00  119.38 29.65  0.00
## BloodPressure                3 768  69.11  19.36  72.00   71.36 11.86  0.00
## SkinThickness                4 768  20.54  15.95  23.00   19.94 17.79  0.00
## Insulin                      5 768  79.80 115.24  30.50   56.75 45.22  0.00
## BMI                          6 768  31.99   7.88  32.00   31.96  6.82  0.00
## DiabetesPedigreeFunction     7 768   0.47   0.33   0.37    0.42  0.25  0.08
## Age                          8 768  33.24  11.76  29.00   31.54 10.38 21.00
## Outcome                      9 768   0.35   0.48   0.00    0.31  0.00  0.00
##                             max  range  skew kurtosis   se
## Pregnancies               17.00  17.00  0.90     0.14 0.12
## Glucose                  199.00 199.00  0.17     0.62 1.15
## BloodPressure            122.00 122.00 -1.84     5.12 0.70
## SkinThickness             99.00  99.00  0.11    -0.53 0.58
## Insulin                  846.00 846.00  2.26     7.13 4.16
## BMI                       67.10  67.10 -0.43     3.24 0.28
## DiabetesPedigreeFunction   2.42   2.34  1.91     5.53 0.01
## Age                       81.00  60.00  1.13     0.62 0.42
## Outcome                    1.00   1.00  0.63    -1.60 0.02
```

```r
dim(diabetes)
```

```
## [1] 768   9
```

```r
#summary data
summary(diabetes) %>% as.data.frame()
```

```
##    Var1                     Var2             Freq
## 1                   Pregnancies Min.   : 0.000
## 2                   Pregnancies 1st Qu.: 1.000
## 3                   Pregnancies Median : 3.000
## 4                   Pregnancies Mean   : 3.845
## 5                   Pregnancies 3rd Qu.: 6.000
## 6                   Pregnancies Max.   :17.000
## 7                       Glucose  Min.   :  0.0
## 8                       Glucose  1st Qu.: 99.0
## 9                       Glucose  Median :117.0
## 10                      Glucose  Mean   :120.9
## 11                      Glucose  3rd Qu.:140.2
## 12                      Glucose  Max.   :199.0
## 13                BloodPressure Min.   :  0.00
## 14                BloodPressure 1st Qu.: 62.00
## 15                BloodPressure Median : 72.00
## 16                BloodPressure Mean   : 69.11
## 17                BloodPressure 3rd Qu.: 80.00
## 18                BloodPressure Max.   :122.00
## 19                SkinThickness  Min.   : 0.00
## 20                SkinThickness  1st Qu.: 0.00
## 21                SkinThickness  Median :23.00
## 22                SkinThickness  Mean   :20.54
## 23                SkinThickness  3rd Qu.:32.00
## 24                SkinThickness  Max.   :99.00
## 25                      Insulin  Min.   :  0.0
## 26                      Insulin  1st Qu.:  0.0
## 27                      Insulin  Median : 30.5
## 28                      Insulin  Mean   : 79.8
## 29                      Insulin  3rd Qu.:127.2
## 30                      Insulin  Max.   :846.0
## 31                          BMI  Min.   : 0.00
## 32                          BMI  1st Qu.:27.30
## 33                          BMI  Median :32.00
## 34                          BMI  Mean   :31.99
## 35                          BMI  3rd Qu.:36.60
## 36                          BMI  Max.   :67.10
## 37      DiabetesPedigreeFunction Min.   :0.0780
## 38      DiabetesPedigreeFunction 1st Qu.:0.2437
## 39      DiabetesPedigreeFunction Median :0.3725
## 40      DiabetesPedigreeFunction Mean   :0.4719
## 41      DiabetesPedigreeFunction 3rd Qu.:0.6262
## 42      DiabetesPedigreeFunction Max.   :2.4200
## 43                           Age  Min.   :21.00
## 44                           Age  1st Qu.:24.00
## 45                           Age  Median :29.00
```

```
## 46                            Age  Mean   :33.24
## 47                            Age  3rd Qu.:41.00
## 48                            Age  Max.   :81.00
## 49                        Outcome  Min.   :0.000
## 50                        Outcome  1st Qu.:0.000
## 51                        Outcome  Median :0.000
## 52                        Outcome  Mean   :0.349
## 53                        Outcome  3rd Qu.:1.000
## 54                        Outcome  Max.   :1.000
```

## STEP-3 : Predicting Diabetes

```
#DO NOT MODIFY THIS CODE
knitr::opts_chunk$set(echo = TRUE)
library(ggplot2) #for data visualization
require(tidyquant)
library(grid) # for grids
library(gridExtra) # for arranging the grids
library(corrplot) # for Correlation plot
library(caret) # for confusion matrix
library(e1071) # for naive bayes
```

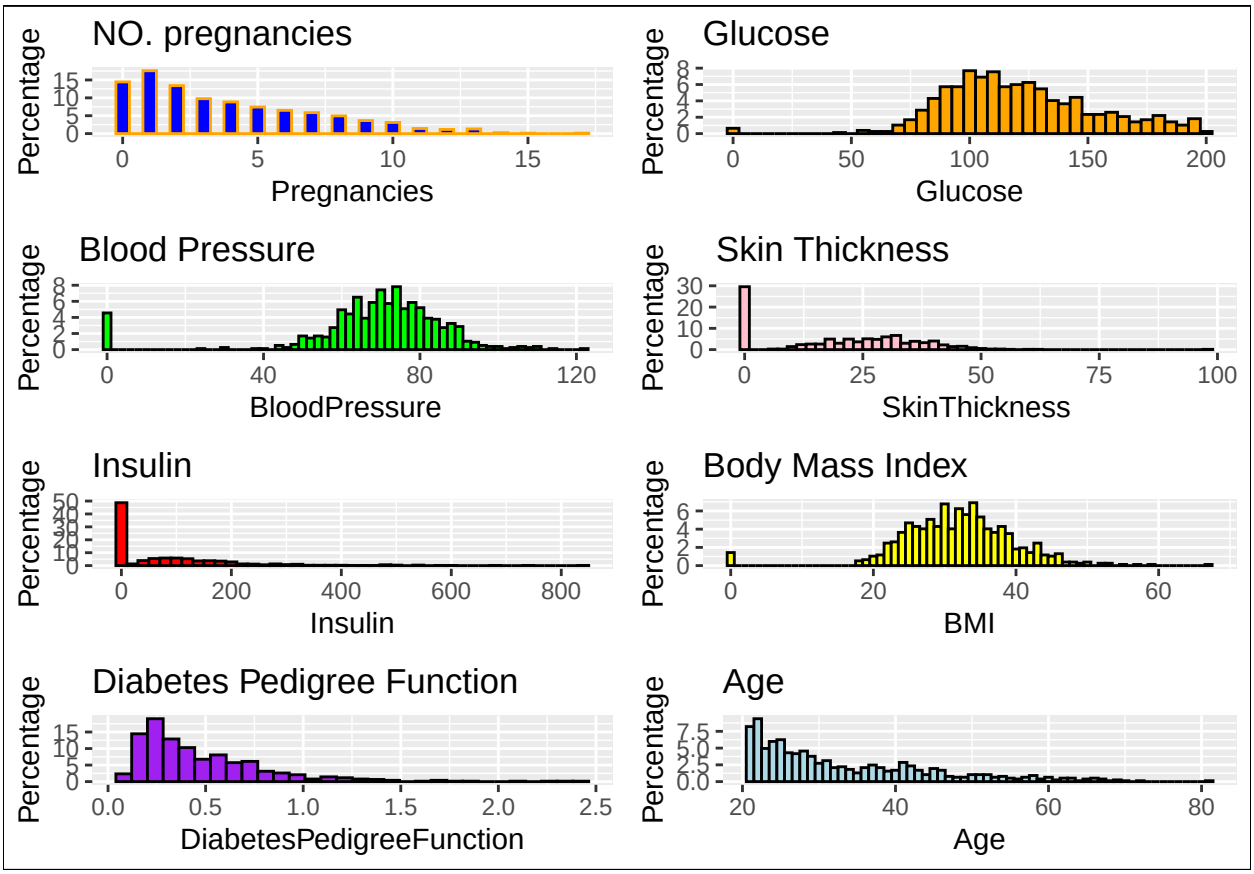
#Plotting Histograms of Numeric Values

```
describe(diabetes)
```

```
##                           vars   n   mean     sd median trimmed   mad   min
## Pregnancies                  1 768   3.85   3.37   3.00    3.46  2.97  0.00
## Glucose                      2 768 120.89  31.97 117.00  119.38 29.65  0.00
## BloodPressure                3 768  69.11  19.36  72.00   71.36 11.86  0.00
## SkinThickness                4 768  20.54  15.95  23.00   19.94 17.79  0.00
## Insulin                      5 768  79.80 115.24  30.50   56.75 45.22  0.00
## BMI                          6 768  31.99   7.88  32.00   31.96  6.82  0.00
## DiabetesPedigreeFunction     7 768   0.47   0.33   0.37    0.42  0.25  0.08
## Age                          8 768  33.24  11.76  29.00   31.54 10.38 21.00
## Outcome                      9 768   0.35   0.48   0.00    0.31  0.00  0.00
##                             max  range  skew kurtosis   se
## Pregnancies               17.00  17.00  0.90     0.14 0.12
## Glucose                  199.00 199.00  0.17     0.62 1.15
## BloodPressure            122.00 122.00 -1.84     5.12 0.70
## SkinThickness             99.00  99.00  0.11    -0.53 0.58
## Insulin                  846.00 846.00  2.26     7.13 4.16
## BMI                       67.10  67.10 -0.43     3.24 0.28
## DiabetesPedigreeFunction   2.42   2.34  1.91     5.53 0.01
## Age                       81.00  60.00  1.13     0.62 0.42
## Outcome                    1.00   1.00  0.63    -1.60 0.02
```

```
p1<-ggplot(diabetes, aes(Pregnancies))+
    geom_histogram(aes(y = 100*(..count..)/sum(..count..)),binwidth=.5,
                   colour="orange", fill="blue")+
```

```
                ggtitle('NO. pregnancies')+ ylab("Percentage")
p2 <- ggplot(diabetes, aes(x=Glucose)) + ggtitle("Glucose") +
  geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 5, colour="black", fill="orange") +
p3 <- ggplot(diabetes, aes(x=BloodPressure)) + ggtitle("Blood Pressure") +
  geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 2, colour="black", fill="green") +
p4 <- ggplot(diabetes, aes(x=SkinThickness)) + ggtitle("Skin Thickness") +
  geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 2, colour="black", fill="pink") + y
p5 <- ggplot(diabetes, aes(x=Insulin)) + ggtitle("Insulin") +
  geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 20, colour="black", fill="red") + y
p6 <- ggplot(diabetes, aes(x=BMI)) + ggtitle("Body Mass Index") +
  geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth = 1, colour="black", fill="yellow") +
p7 <- ggplot(diabetes, aes(x=DiabetesPedigreeFunction)) + ggtitle("Diabetes Pedigree Function") +
  geom_histogram(aes(y = 100*(..count..)/sum(..count..)), colour="black", fill="purple") + ylab("Percen
p8 <- ggplot(diabetes, aes(x=Age)) + ggtitle("Age") +
  geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth=1, colour="black", fill="lightblue")
grid.arrange(p1, p2, p3, p4, p5, p6, p7, p8, ncol=2)
grid.rect(width = 1, height = 1, gp = gpar(lwd = 1, col = "black", fill = NA))
```

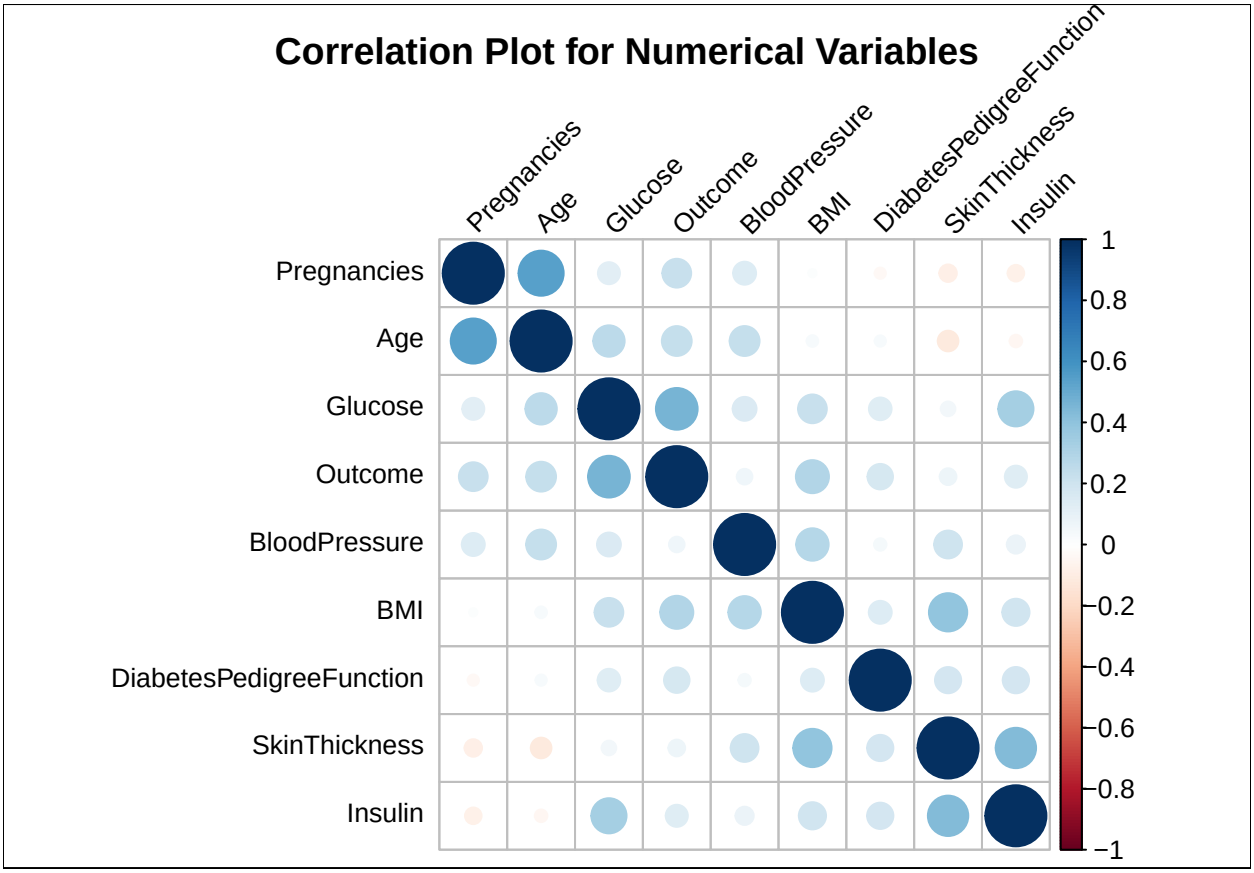



**All the variables seem to have reasonable broad distribution, therefore, will be kept for the regression analysis.**

### Correlation between Numeric Variables

Here, **sapply()** function will return the columns from the diabetes dataset which have numeric values. **cor()** function will produce correlation matrix of all those numeric columns returned by sapply(). **corrplot()** provides a visual representation of correlation matrix that supports automatic variable reordering to help detect hidden patterns among variables.

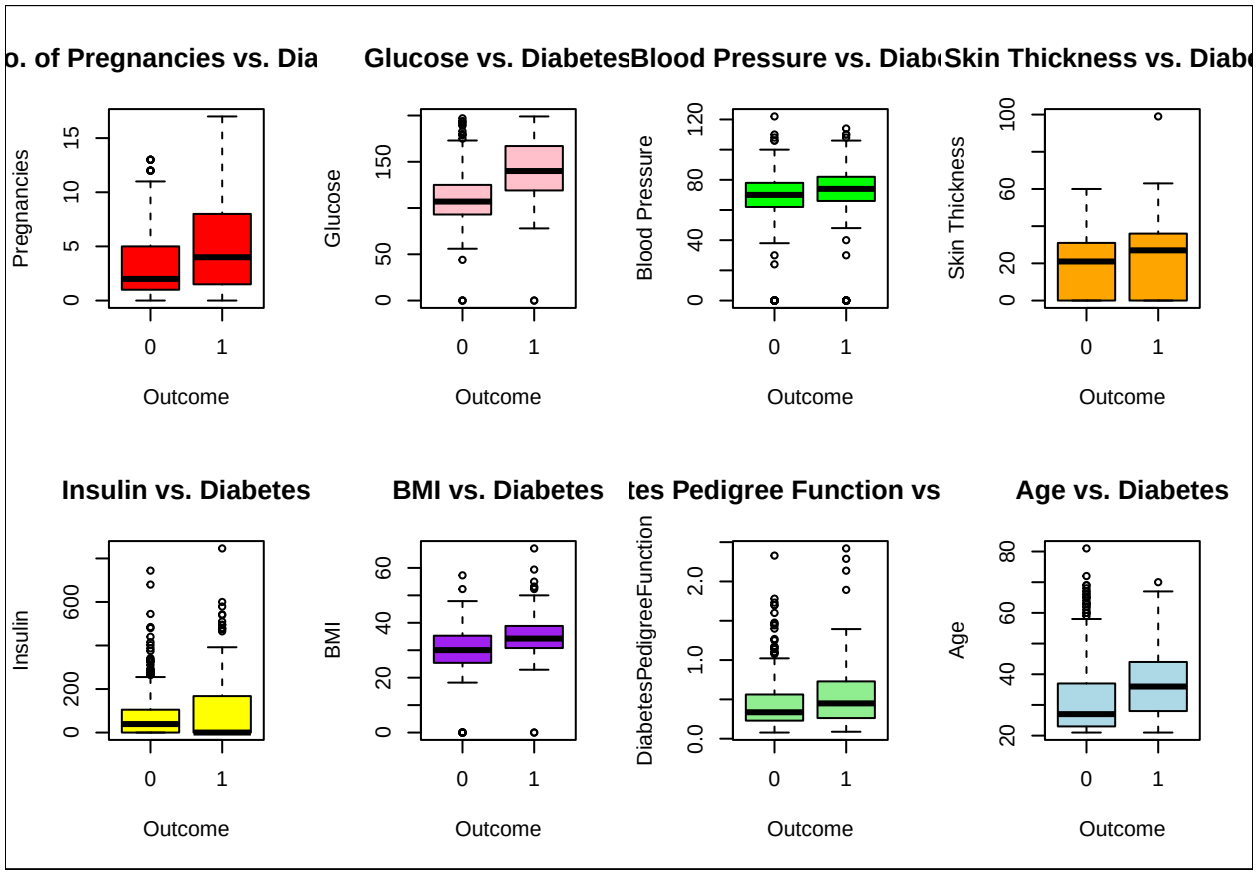
```
numeric.var <- sapply(diabetes, is.numeric)
corr.matrix <- cor(diabetes[,numeric.var])
corrplot(corr.matrix, main="\n\nCorrelation Plot for Numerical Variables", order = "hclust", tl.col = "
box(which = "outer", lty = "solid")
```




**The numeric variables are almost not correlated.**

```
attach(diabetes)
par(mfrow=c(2,4))
boxplot(Pregnancies~Outcome, main="No. of Pregnancies vs. Diabetes",
        xlab="Outcome", ylab="Pregnancies",col="red")
boxplot(Glucose~Outcome, main="Glucose vs. Diabetes",
        xlab="Outcome", ylab="Glucose",col="pink")
boxplot(BloodPressure~Outcome, main="Blood Pressure vs. Diabetes",
        xlab="Outcome", ylab="Blood Pressure",col="green")
boxplot(SkinThickness~Outcome, main="Skin Thickness vs. Diabetes",
        xlab="Outcome", ylab="Skin Thickness",col="orange")
boxplot(Insulin~Outcome, main="Insulin vs. Diabetes",
        xlab="Outcome", ylab="Insulin",col="yellow")
```

```
boxplot(BMI~Outcome, main="BMI vs. Diabetes",
        xlab="Outcome", ylab="BMI",col="purple")
boxplot(DiabetesPedigreeFunction~Outcome, main="Diabetes Pedigree Function vs. Diabetes", xlab="Outcome
boxplot(Age~Outcome, main="Age vs. Diabetes",
        xlab="Outcome", ylab="Age",col="lightblue")
box(which = "outer", lty = "solid")
```




**Blood pressure and skin thickness show little variation with diabetes, they will be excluded from the model. Other variables show more or less correlation with diabetes, so will be kept.**

### Logistic Regression

```
diabetes$BloodPressure <- NULL
diabetes$SkinThickness <- NULL
train <- diabetes[1:540,]
test <- diabetes[541:768,]
model <-glm(Outcome ~.,family=binomial,data=train)
summary(model)
```

```
##
## Call:
## glm(formula = Outcome ~ ., family = binomial, data = train)
##
```

```
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -8.3461752  0.8157916 -10.231  < 2e-16 ***
## Pregnancies               0.1246856  0.0373214   3.341 0.000835 ***
## Glucose                   0.0315778  0.0042497   7.431 1.08e-13 ***
## Insulin                  -0.0013400  0.0009441  -1.419 0.155781
## BMI                       0.0881521  0.0164090   5.372 7.78e-08 ***
## DiabetesPedigreeFunction  0.9642132  0.3430094   2.811 0.004938 **
## Age                       0.0018904  0.0107225   0.176 0.860053
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 700.47  on 539  degrees of freedom
## Residual deviance: 526.56  on 533  degrees of freedom
## AIC: 540.56
##
## Number of Fisher Scoring iterations: 5
```

The top three most relevant features are "Glucose", "BMI" and "Number of times pregnant" because of the low p-values. "Insulin" and "Age" appear not statistically significant.

```
anova(model, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Outcome
##
## Terms added sequentially (first to last)
##
##
##                           Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                        539     700.47
## Pregnancies                1   26.314       538     674.16 2.901e-07 ***
## Glucose                    1  102.960       537     571.20 < 2.2e-16 ***
## Insulin                    1    0.062       536     571.14  0.803341
## BMI                        1   36.135       535     535.00 1.841e-09 ***
## DiabetesPedigreeFunction   1    8.414       534     526.59  0.003723 **
## Age                        1    0.031       533     526.56  0.860201
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

**From the table of deviance, we can see that adding insulin and age have little effect on the residual deviance.**

### Cross Validation

```
fitted.results <- predict(model,newdata=test,type='response')
fitted.results <- ifelse(fitted.results > 0.5,1,0)
(conf_matrix_logi<-table(fitted.results, test$Outcome))
```

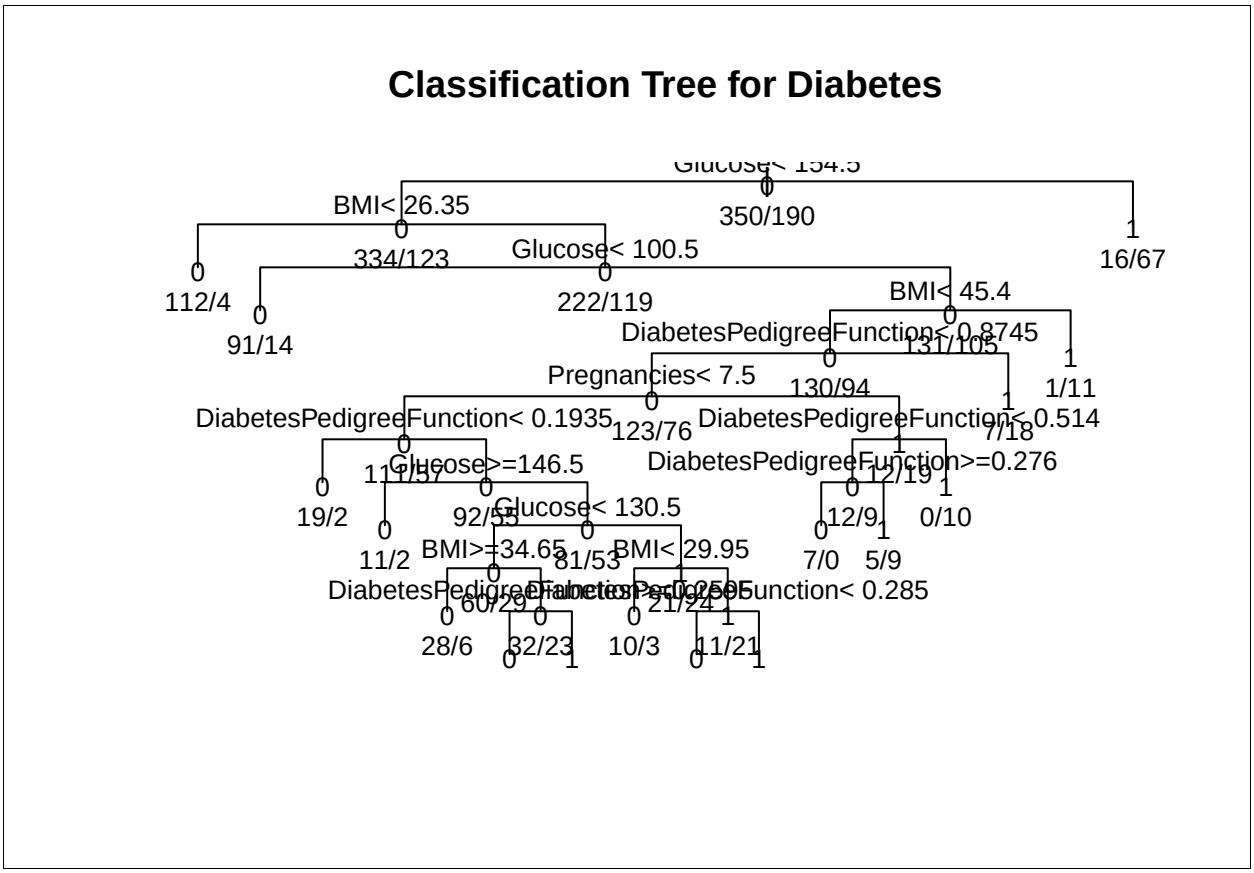
```
##
## fitted.results   0   1
##              0 136  34
##              1  14  44
```

```
# misClasificError <- mean(fitted.results == test$Outcome)
misClasificError <- mean(fitted.results != test$Outcome)
print(paste('Accuracy',1-misClasificError))
```

```
## [1] "Accuracy 0.789473684210526"
```

### Decision Tree

```
library(rpart)
model2 <- rpart(Outcome ~ Pregnancies + Glucose + BMI + DiabetesPedigreeFunction, data=train,method="cla
plot(model2, uniform=TRUE,
    main="Classification Tree for Diabetes")
text(model2, use.n=TRUE, all=TRUE, cex=.8)
box(which = "outer", lty = "solid")
```



**This means if a person's BMI less than 45.4 and his/her Diabetes Pedigree function less than 0.8745, then the person is more likely to have diabetes.**

### ###Confusion Table and Accuracy

```
treePred <- predict(model2, test, type = 'class')
(conf_matrix_dtree<-table(treePred, test$Outcome))
```

```
##
## treePred   0   1
##        0 121  29
##        1  29  49
```

```
accuracy2 <- mean(treePred==test$Outcome)
print(paste('accuracy',accuracy2))
```

```
## [1] "accuracy 0.745614035087719"
```

### Naive Bayes

```
# creating Naive Bayes model
model_naive <- naiveBayes(Outcome ~., data = train)
```

### Confusion Table and Accuracy

```
# predicting target
toppredict_set <- test[1:6]
dim(toppredict_set)
```

```
## [1] 228   6
```

```
preds_naive <- predict(model_naive, newdata = toppredict_set)
(conf_matrix_naive <- table(preds_naive, test$Outcome))
```

```
##
## preds_naive   0   1
##           0 129  29
##           1  21  49
```

```
accuracy3 <- mean(preds_naive==test$Outcome)
print(paste('accuracy',accuracy3))
```

```
## [1] "accuracy 0.780701754385965"
```

**Conclusion**

**If we compare accuracy and sensitivity level of our models to see the highest value, we can summarise as followed :**

```
confusionMatrix(conf_matrix_logi)
```

```
## Confusion Matrix and Statistics
##
##
## fitted.results   0   1
##              0 136  34
##              1  14  44
##
##                Accuracy : 0.7895
##                  95% CI : (0.7307, 0.8405)
##     No Information Rate : 0.6579
##     P-Value [Acc > NIR] : 9.506e-06
##
##                   Kappa : 0.5016
##
##  Mcnemar's Test P-Value : 0.006099
##
##             Sensitivity : 0.9067
##             Specificity : 0.5641
##          Pos Pred Value : 0.8000
##          Neg Pred Value : 0.7586
##              Prevalence : 0.6579
##          Detection Rate : 0.5965
##    Detection Prevalence : 0.7456
##       Balanced Accuracy : 0.7354
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(conf_matrix_dtree)
```

```
## Confusion Matrix and Statistics
##
##
## treePred   0   1
##        0 121  29
##        1  29  49
##
##                Accuracy : 0.7456
##                  95% CI : (0.6839, 0.8008)
##     No Information Rate : 0.6579
##     P-Value [Acc > NIR] : 0.002723
##
##                   Kappa : 0.4349
##
##  Mcnemar's Test P-Value : 1.000000
##
```

```
##             Sensitivity : 0.8067
##             Specificity : 0.6282
##          Pos Pred Value : 0.8067
##          Neg Pred Value : 0.6282
##              Prevalence : 0.6579
##          Detection Rate : 0.5307
##    Detection Prevalence : 0.6579
##       Balanced Accuracy : 0.7174
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(conf_matrix_naive)
```

```
## Confusion Matrix and Statistics
##
##
## preds_naive   0   1
##           0 129  29
##           1  21  49
##
##                Accuracy : 0.7807
##                  95% CI : (0.7213, 0.8326)
##     No Information Rate : 0.6579
##     P-Value [Acc > NIR] : 3.562e-05
##
##                   Kappa : 0.5005
##
##  Mcnemar's Test P-Value : 0.3222
##
##             Sensitivity : 0.8600
##             Specificity : 0.6282
##          Pos Pred Value : 0.8165
##          Neg Pred Value : 0.7000
##              Prevalence : 0.6579
##          Detection Rate : 0.5658
##    Detection Prevalence : 0.6930
##       Balanced Accuracy : 0.7441
##
##        'Positive' Class : 0
##
```

**In this project, we compared the performance of Linear Regression, Decision Tree and Naive Bayes algorithms and found that Logistic Regression performed better on this standard, unaltered dataset. After, Logistic Regression there comes Naive Bayes algorithm with more accuracy than the Decision Tree. Accuracy given by Logistic Regression was 79%, Decision Tree was 74% and Naive Bayes was 78%.**