ASSIGNMENT 3.
PREDICTIVE MODELLING.
NAME: SAAD SHAIKH.
STUDENT ID: 20070328.

INTRDUCTION.

The aim of this assignment is to understand the predictive modelling for binomial classification using a logistic regression approach. Further to determine the performance on feature-based selection model vs predictors models.

TASKS.

The Assignment is divided into 6 parts. Each of which is in detail below.

TASK 1: Choosing dataset for assignment that contain at-least one column with only binary values (0,1). If the columns are binary, you do not need to change them, otherwise you should recode the values to binaries. Writing a summary of the dataset: stating the number of observations and features, explaining the meaning of the data and role of individual features. Chose the binary columns to predict the outcome.

Selection and inspection of the dataset.

The selected dataset is "PimaIndiansDiabetes2" which have the observations of 8 test done on women and their diabetics test result. The observations are described below:

Pregnant	Number of times pregnant.	Mass	Body mass index (weight in kg/(height in m)\^2)
Glucose	Plasma glucose concentration	Pedigree	Diabetes pedigree function
Pressure	Diastolic blood pressure (mm Hg)	Age	Age (in years)
Triceps	Triceps skin fold thickness (mm)	Diabetes	Test result for diabetes
Insulin	2-Hour serum insulin (mu U/ml)		

. Summarizing Dataset.

```
#loading data set
library(mlbench)
data(PimaIndiansDiabetes2)
a <- PimaIndiansDiabetes2

#summarizing the data set
summary(a)
```

```
> summary(a)
    pregnant         glucose         pressure         triceps
 Min.   : 0.000   Min.   : 44.0   Min.   : 24.00   Min.   : 7.00
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.00   1st Qu.:22.00
 Median : 3.000   Median :117.0   Median : 72.00   Median :29.00
 Mean   : 3.845   Mean   :121.7   Mean   : 72.41   Mean   :29.15
 3rd Qu.: 6.000   3rd Qu.:141.0   3rd Qu.: 80.00   3rd Qu.:36.00
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
                  NA's   :5       NA's   :35       NA's   :227
    insulin            mass          pedigree          age
 Min.   : 14.00   Min.   :18.20   Min.   :0.0780   Min.   :21.00
 1st Qu.: 76.25   1st Qu.:27.50   1st Qu.:0.2437   1st Qu.:24.00
 Median :125.00   Median :32.30   Median :0.3725   Median :29.00
 Mean   :155.55   Mean   :32.46   Mean   :0.4719   Mean   :33.24
 3rd Qu.:190.00   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
 Max.   :846.00   Max.   :67.10   Max.   :2.4200   Max.   :81.00
 NA's   :374      NA's   :11
 diabetes
 neg:500
 pos:268
```

Handling NA values.

The column having NA values are handled using Mean imputation.

```
#handling NA values in columns by mean imputation
a$glucose[is.na(a$glucose)] <- mean(a$glucose, na.rm = TRUE)
a$pressure[is.na(a$pressure)] <- mean(a$pressure, na.rm = TRUE)
a$triceps[is.na(a$triceps)] <- mean(a$triceps, na.rm = TRUE)
a$insulin[is.na(a$insulin)] <- mean(a$insulin, na.rm = TRUE)
a$mass[is.na(a$mass)] <- mean(a$mass, na.rm = TRUE)
summary(a)
```

```
    pregnant         glucose         pressure         triceps
 Min.   : 0.000   Min.   : 44.00   Min.   : 24.00   Min.   : 7.00
 1st Qu.: 1.000   1st Qu.: 99.75   1st Qu.: 64.00   1st Qu.:25.00
 Median : 3.000   Median :117.00   Median : 72.20   Median :29.15
 Mean   : 3.845   Mean   :121.69   Mean   : 72.41   Mean   :29.15
 3rd Qu.: 6.000   3rd Qu.:140.25   3rd Qu.: 80.00   3rd Qu.:32.00
 Max.   :17.000   Max.   :199.00   Max.   :122.00   Max.   :99.00
    insulin           mass          pedigree          age
 Min.   : 14.0   Min.   :18.20   Min.   :0.0780   Min.   :21.00
 1st Qu.:121.5   1st Qu.:27.50   1st Qu.:0.2437   1st Qu.:24.00
 Median :155.5   Median :32.40   Median :0.3725   Median :29.00
 Mean   :155.5   Mean   :32.46   Mean   :0.4719   Mean   :33.24
 3rd Qu.:155.5   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
 Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00
 diabetes
 neg:500
 pos:268
```

Checking the structure of dataset.

The column we want to predict i.e. (diabetes) is of datatype: 2-level factor which consists of positive and negative.

We need to convert it to binary to read and convert the datatype to numeric for further process.

```
#Modifying the data to convert categorical outcome to binary
levels(a$diabetes) <- c(0,1) # binary outcome
a$diabetes <- as.numeric(a$diabetes)
a$diabetes <- a$diabetes - 1
summary(a)
```

TASK 2: Normalizing the predictors using Z-score transformation and assess the normality using the Shapiro-wilk test.

Applying Z-score to normalize the predictor column: **PREGNANT**.
As, observed Mean = 0, Variance = 1.
After Shapiro-wilk test, we find p < 0.05. Hence, a strong reason to reject normal hypothesis.

```
#Z-score transformation on pregnant column
p_mean <- mean(a$pregnant, na.rm = TRUE)
p_sd <- sqrt(var(a$pregnant, na.rm = TRUE))
a$p_normal <- (a$pregnant-p_mean)/p_sd
summary(a$p_normal)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.1411 -0.8443 -0.2508  0.0000  0.6395  3.9040
```

```
#Shapiro Test on pregnant column
shapiro.test(a$pregnant)#comment:
```

```
        Shapiro-Wilk normality test

data:  a$pregnant
W = 0.90428, p-value < 2.2e-16
```

Applying Z-score to normalize the predictor column: **GLUCOSE**.
As, observed Mean = 0, Variance = 1.
After Shapiro-wilk test, we find p < 0.05. Hence, a strong reason to reject normal hypothesis.

```
#Z-score transformation on glucose column
g_mean <- mean(a$glucose, na.rm = TRUE)
g_sd <- sqrt(var(a$glucose, na.rm = TRUE))
a$g_normal <- (a$glucose-g_mean)/g_sd
summary(a$g_normal)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-2.5525 -0.7208 -0.1540  0.0000  0.6099  2.5402
```

```
> #Shapiro Test on glucose column
> shapiro.test(a$glucose)#comment:
o reject normal hypothesis.

        Shapiro-Wilk normality test

data:  a$glucose
W = 0.9699, p-value = 1.777e-11
```

Applying Z-score to normalize the predictor column: **PRESSURE**.
As, observed Mean = 0, Variance = 1.
After Shapiro-wilk test, we find p < 0.05. Hence, a strong reason to reject normal hypothesis.

```
#Z-score transformation on pressure column
pr_mean <- mean(a$pressure, na.rm = TRUE)
pr_sd <- sqrt(var(a$pressure, na.rm = TRUE))
a$pr_normal <- (a$pressure-pr_mean)/pr_sd
summary(a$pr_normal)
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-4.00164  -0.69485  -0.01675   0.00000   0.62786   4.09998
```

```
>
> #Shapiro Test on pressure column
> shapiro.test(a$pressure)#comment:
 to reject normal hypothesis.

        Shapiro-Wilk normality test

data:  a$pressure
W = 0.98804, p-value = 6.463e-06
```

Applying Z-score to normalize the predictor column: **TRICEPS**.
As, observed Mean = 0, Variance = 1.
After Shapiro-wilk test, we find p < 0.05. Hence, a strong reason to reject normal hypothesis.

```
#Z-score transformation on triceps column
tri_mean <- mean(a$triceps, na.rm = TRUE)
tri_sd <- sqrt(var(a$triceps, na.rm = TRUE))
a$tri_normal <- (a$triceps-tri_mean)/tri_sd
summary(a$tri_normal)
```

```
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-2.5200 -0.4725  0.0000  0.0000  0.3238  7.9453
```

```
> #Shapiro Test on triceps column
> shapiro.test(a$tri_normal)#comment:
n to reject normal hypothesis.

        Shapiro-Wilk normality test

data:  a$tri_normal
W = 0.92808, p-value < 2.2e-16
```

Applying Z-score to normalize the predictor column: **INSULIN**.
As, observed Mean = 0, Variance = 1.
After Shapiro-wilk test, we find p < 0.05. Hence, a strong reason to reject normal hypothesis.

```
#Z-score transformation on insulin column
insulin_mean <- mean(a$insulin, na.rm = TRUE)
insulin_sd <- sqrt(var(a$insulin, na.rm = TRUE))
a$insulin_normal <- (a$insulin-insulin_mean)/insulin_mean
summary(a$insulin_normal)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.9100 -0.2189  0.0000  0.0000  0.0000  4.4388
```

```
>
> #Shapiro Test on insulin column
> shapiro.test(a$insulin_normal)#comment:
eason to reject normal hypothesis.

        Shapiro-Wilk normality test

data:  a$insulin_normal
W = 0.69055, p-value < 2.2e-16
```

Applying Z-score to normalize the predictor column: **MASS**.
As, observed Mean = 0, Variance = 1.
After Shapiro-wilk test, we find p < 0.05. Hence, a strong reason to reject normal hypothesis.

```
#Z-score transformation on mass column
mass_mean <- mean(a$mass, na.rm = TRUE)
mass_sd <- sqrt(var(a$mass, na.rm = TRUE))
a$mass_normal <- (a$mass-mass_mean)/mass_sd
summary(a$mass_normal)
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-2.073767 -0.721070 -0.008358  0.000000  0.602537  5.038803
>
```

```
>
> #Shapiro Test on mass column
> shapiro.test(a$mass_normal)#comment:
on to reject normal hypothesis.

        Shapiro-Wilk normality test

data:  a$mass_normal
W = 0.97946, p-value = 6.526e-09
```

Applying Z-score to normalize the predictor column: **PEDIGREE**.
As, observed Mean = 0, Variance = 1.
After Shapiro-wilk test, we find p < 0.05. Hence, a strong reason to reject normal hypothesis.

```
#Z-score transformation on pedigree column
pedigree_mean <- mean(a$pedigree, na.rm = TRUE)
pedigree_sd <- sqrt(var(a$pedigree, na.rm = TRUE))
a$pedigree_normal <- (a$pedigree-pedigree_mean)/pedigree_sd
summary(a$pedigree_normal)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.1888 -0.6885 -0.2999  0.0000  0.4659  5.8797
>
```

```
>
> #Shapiro Test on pedigree column
> shapiro.test(a$pedigree_normal)#comment:
 reason to reject normal hypothesis.

        Shapiro-Wilk normality test

data:  a$pedigree_normal
W = 0.83652, p-value < 2.2e-16
```

Applying Z-score to normalize the predictor column: **AGE**.
As, observed Mean = 0, Variance = 1.
After Shapiro-wilk test, we find $p < 0.05$. Hence, a strong reason to reject normal hypothesis.

```
#Z-score transformation on age column
mean_age <- mean(a$age, na.rm = TRUE)
sd_age <- sqrt(var(a$age, na.rm = TRUE))
a$age_normal <- (a$age-mean_age)/sd_age
summary(a$age_normal)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.0409 -0.7858 -0.3606  0.0000  0.6598  4.0611
```

```
>
> #Shapiro Test on age column
> shapiro.test(a$age_normal)#comment:
n to reject normal hypothesis.

        Shapiro-Wilk normality test

data:  a$age_normal
W = 0.87477, p-value < 2.2e-16
```

TASK 3: Split your full data into a training (70%) and a test(30%) subsets.

Selecting only Normalized columns from the dataset.

```
> # selecting only normalized columns
> a <- a[c(9:17)]
> str(a)
'data.frame':   768 obs. of  9 variables:
 $ diabetes       : num  1 0 1 0 1 0 1 0 1 1 ...
 $ p_normal       : num  0.64 -0.844 1.233 -0.844 -1.141 ...
 $ g_normal       : num  0.865 -1.205 2.015 -1.074 0.503 ...
 $ pr_normal      : num  -0.0335 -0.5295 -0.6949 -0.5295 -2.6789 ...
 $ tri_normal     : num  0.6651 -0.0175 0 -0.7 0.6651 ...
 $ insulin_normal : num  0 0 0 -0.3957 0.0801 ...
 $ mass_normal    : num  0.166 -0.852 -1.332 -0.634 1.548 ...
 $ pedigree_normal: num  0.468 -0.365 0.604 -0.92 5.481 ...
 $ age_normal     : num  1.4251 -0.1905 -0.1055 -1.0409 -0.0205 ...
> set.seed(123)
> sample <- sample(seq_len(nrow(a)), size = 0.7*nrow(a))
> a_training <- a[sample,]
> a_test <- a[-sample,]
```

Task4: In the Training set perform feature selection using the T, F and Wilcoxon scores. Then create new subsets, the Reduced Training, and the Reduced Test subsets, of the Training and Test sets, respectively, that contain only those features that ended up in Top 3 for at least one of the feature selection approaches used.

- Defining function for feature selection: **FSCR**
- Defining variables for the function:

X= predictor columns; Y= response column; k= number of features

- Top 3 features: g normal, mass normal, age normal

```
FSCR = function(X, Y, k) # X - matrix with predictors, Y - binary outcome, k top candidates
{
  J<- rep(NA, ncol(X))
  names(J)<- colnames(X)
  for (i in 1:ncol(X))
  {
    X1<- X[which(Y==0),i]
    X2<- X[which(Y==1),i]
    mu1<- mean(X1); mu2<- mean(X2); mu<- mean(X[,i])
    var1<- var(X1); var2<- var(X2)
    n1<- length(X1); n2<- length(X2)
    J[i]<- (n1*(mu1-mu)^2+n2*(mu2-mu)^2)/(n1*var1+n2*var2)
  }
  J<- sort(J, decreasing=TRUE)[1:k]
  return(list(score=J))
}
```

```
#Function variables
X <- a[c(2:9)] #predictors
Y <- a[c(1)] #outcomes
k <- 3 #top-3 features
> #Looking for top-3 features
> FSCR(X, Y, k) #Top-3: g_normal, mass_normal, age_normal
$score
   g_normal mass_normal  age_normal
0.32008212  0.10750199  0.06008179
```

Defining function for feature selection: **TSCR**

• Defining variables for the function:

X= predictor columns; Y= response column ; k= number of features

• Top 3 features: pedigree normal, pr_normal, insulin normal

```
TSCR = function(X, Y, k) # X - matrix with predictors, Y - binary outcome, k top candidates
{
  J<- rep(NA, ncol(X))
  names(J)<- colnames(X)
  for (i in 1:ncol(X))
  {
    X1<- X[which(Y==0),i]
    X2<- X[which(Y==1),i]
    mu1<- mean(X1); mu2<- mean(X2)
    var1<- var(X1); var2<- var(X2)
    n1<- length(X1); n2<- length(X2)
    J[i]<- (mu1-mu2)/sqrt(var1/n1+var2/n2)
  }
  J<- sort(J, decreasing=TRUE)[1:k]
  return(list(score=J))
}
```

```
> #Function variables
> X <- a[c(2:9)] #predictors
> Y <- a[c(1)] #outcomes
> k <- 3 #top-3 features
>
> #Looking for top-3 features
> TSCR(X, Y, k) #Top-3: pedigree_normal, pr_normal, insulin_normal
$score
pedigree_normal       pr_normal  insulin_normal
      -4.576812       -4.659366       -5.660196
```

Defining Function for feature selection: **WLCX**

• Defining variables for the function:

X= predictor columns; Y= response column ; k= number of features

• Top 3 features: g_normal, age_normal, mass_normal

```
WLCX = function(X, Y, k) # X - matrix with predictors, Y - binary outcome, k top candidates
{
  J<- rep(NA, ncol(X))
  names(J)<- colnames(X)
  for (i in 1:ncol(X))
  {
    X_rank<- apply(data.matrix(X[,i]), 2, function(c) rank(c))
    X1_rank<- X_rank[which(Y==0)]
    X2_rank<- X_rank[which(Y==1)]
    mu1<- mean(X1_rank); mu2<- mean(X2_rank); mu<- mean(X_rank)
    n1<- length(X1_rank); n2<- length(X2_rank); N<- length(X_rank)
    num<- (n1*(mu1-mu)^2+ n2*(mu2-mu)^2)
    denom<- 0
    for (j in 1:n1)
      denom<- denom+(X1_rank[j]-mu)^2
    for (j in 1:n2)
      denom<- denom+(X2_rank[j]-mu)^2
    J[i]<- (N-1)*num/denom
  }
  J<- sort(J, decreasing=TRUE)[1:k]
  return(list(score=J))
}
```

```
> #Function variables
> X <- a[c(2:9)] #predictors
> Y <- a[c(1)] #outcomes
> k <- 3 #top-3 features
>
> #Looking for top-3 features
> WLCX(X, Y, k) #Top-3: g_normal, age_normal, mass_normal
$score
   g_normal  age_normal mass_normal
  177.91876    73.25301    72.08987
```

Defining Reduced Training Dataset and Reduced Test Dataset

In the above 3 selection models, we select g_normal, mass_normal, age_normal.
We then define reduced training set (a_red_training) by selecting only above-mentioned columns along with the training dataset (a).

```
> #defining reduced training and reduced test dataset
> a_red_training <- a_training[c("g_normal", "mass_normal", "age_normal", "diabetes")]
> a_red_test <- a_test[c("g_normal", "mass_normal", "age_normal", "diabetes")]
```

Task5: Train the logistic regression approach on both the Training set and the Reduced Training set.

Training the logistic regression on training dataset

• lgr1 is the logistic regression model trained for response variable diabetes on all other columns in the training dataset (a_training)

Deviance Residuals:

• Deviance residuals min: -2.46; Deviance residuals max: 2.38

Coefficients:

• We look at coefficients, to get the columns with the highest influence on diabetes (rated: *, **,***)

• For a unit increase, in the 3-star(***) rated glucose_normal variable, the log odds of the patient to be diabetic is 1.18

• For a unit increase, in the 3-star(***) rated mass_normal variable, the log odds of the patient to be diabetic is 0.63

For a unit increase, in the 2-star(**) rated pregnant_normal variable, the log odds of the patient to be diabetic is 0.35

```
> # Logistic regression with training dataset
> model1<- glm(diabetes~., data=a_training, family="binomial")
> summary(model1)

Call:
glm(formula = diabetes ~ ., family = "binomial", data = a_training)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4674  -0.7183  -0.4039   0.7125   2.3879

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     -0.89855    0.11727  -7.662 1.83e-14 ***
p_normal         0.35122    0.12991   2.704  0.00686 **
g_normal         1.18066    0.14731   8.015 1.10e-15 ***
pr_normal       -0.08389    0.12497  -0.671  0.50204
tri_normal       0.08440    0.13841   0.610  0.54200
insulin_normal  -0.19361    0.21573  -0.897  0.36947
mass_normal      0.63562    0.14837   4.284 1.84e-05 ***
pedigree_normal  0.21670    0.11335   1.912  0.05590 .
age_normal       0.15550    0.13401   1.160  0.24589
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 694.17  on 536  degrees of freedom
Residual deviance: 495.77  on 528  degrees of freedom
AIC: 513.77

Number of Fisher Scoring iterations: 5
```

Training the logistic regression on reduced training dataset

• lgr2 is the logistic regression model trained for response variable diabetes on all other columns on the reduced training dataset (a_red_training)

Deviance Residuals:

• Deviance residuals min: -2.48; Deviance residuals max: 2.39

Coefficients:

• We look at coefficients, to get the columns with the highest influence on diabetes (rated: ****)

• For a unit increase, in the 3-star(***) rated g_normal variable, the log odds of the patient to be diabetic is 1.11

• For a unit increase, in the 3-star(***) rated mass_normal variable, the log odds of the patient to be diabetic is 0.65

• For a unit increase, in the 3-star(**) rated age_normal variable, the log odds of the patient to be diabetic is 0.33

```
> #Logistic regression with reduced training dataset
> model2<- glm(diabetes~., data=a_red_training, family="binomial")
> summary(model2)

Call:
glm(formula = diabetes ~ ., family = "binomial", data = a_red_training)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4854  -0.7194  -0.4145   0.7117   2.3988

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.8744     0.1148  -7.620 2.54e-14 ***
g_normal      1.1155     0.1313   8.494  < 2e-16 ***
mass_normal   0.6532     0.1216   5.374 7.71e-08 ***
age_normal    0.3332     0.1085   3.070  0.00214 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 694.17  on 536  degrees of freedom
Residual deviance: 508.02  on 533  degrees of freedom
AIC: 516.02

Number of Fisher Scoring iterations: 4
```

Task6: Apply the trained models to both the Test set and the Reduced Test set. Then evaluate the Area under ROC-curve (AUC) for both cases and comment on whether feature reduction has heavily decreased the AUC for the Training set compared to the Reduced Training dataset.

Predicting response 1 (pr_lgr1): with training model to test dataset

• Predicting response 2 (pr_lgr2): with reduced training model to reduced test dataset

• AUC (auc1) on training model to test dataset: **0.829**

• AUC (auc2) on reduced training model to reduced test dataset: **0.821**

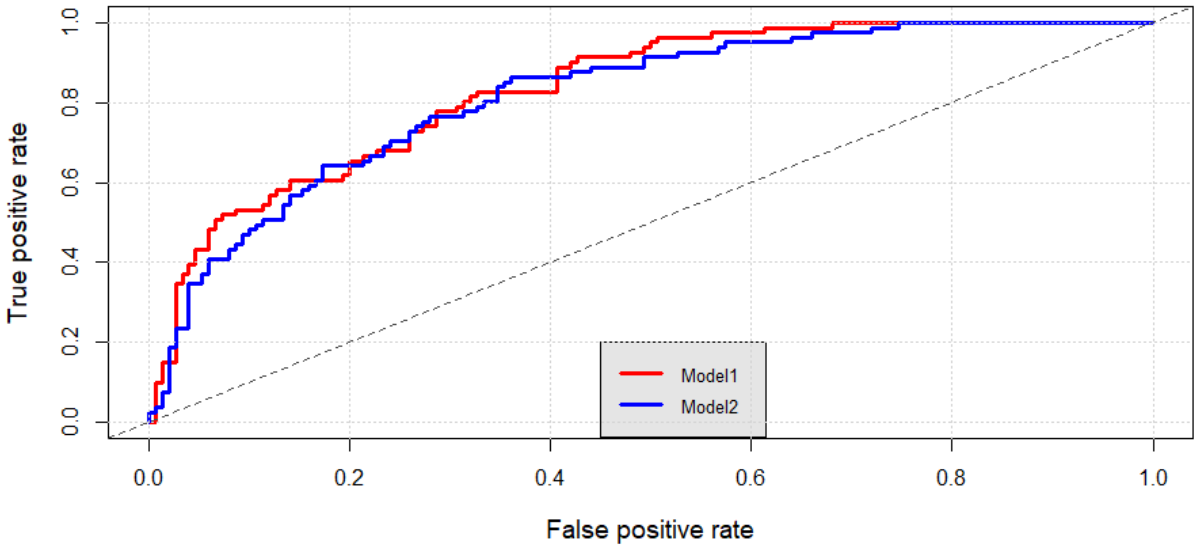
```
#Discrimination
library(Hmisc); library(ggplot2); library(gridExtra)
# Predicting response: training model to test dataset
pr_model1<- predict(model1, a_test, type="response")
#Predicting response: reduced training model to reduced test dataset
pr_model2<- predict(model2, a_red_test, type="response")

# ROC-analysis
library(ROCR)
pred1.obj <- prediction(predictions = pr_model1, labels = a_test$diabetes)
pred2.obj  <- prediction(predictions = pr_model2,  labels = a_red_test$diabetes)

perf1 <- performance(pred1.obj, measure="tpr", x.measure="fpr")
perf2 <- performance(pred2.obj, measure="tpr", x.measure="fpr")
```

```
>
> auc1<- somers2(pr_model1, a_test$diabetes)[1]; auc1
        C
0.8333333
> auc2<- somers2(pr_model2, a_red_test$diabetes)[1]; auc2
        C
0.8160494
```

Plotting Area under ROC curve:

In below graph, we can see that models are inclined towards the true positive rate and the area under the ROC curve is high.

```
plot(perf1, lty = 1, col = "red", lwd = 3, cex.lab = 1.2)
plot(perf2,  lty = 1, col = "blue",  lwd = 3, add = T)
legend(0.45, 0.2, c("Model1", "Model2"),
       lty = c(1,1), col = c("red","blue"),
       lwd = c(3,3), cex = 0.8, bg = "gray90")
grid(nx = NULL, ny = NULL, col = "lightgray", lty = "dotted",
     lwd = par("lwd"), equilogs = TRUE)
abline(0, 1, col = "gray30", lwd = 1, lty = 2)
```


True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
Model1
Model2