Pima Indians Diabetes Prediction

Anil Raju and Ronald White

January 2022

Abstract:

The main goal of this study was to develop a model that could accurately predict the onset of gestational diabetes given eight health, descriptive, and/or diagnostic predictors. The data was obtained from the National Institute of Diabetes and Digestive and Kidney Diseases. The population included all patients who are women, at least 21, and of Pima Indian Heritage. After pre-processing, the data was split into a training and test set. Data in the training set were used to create linear and nonlinear classification models. The top two performing models were used to examine predictions on the test data and the model that was performed optimally was selected as our recommendation.

Table of Contents

1 Background

Diabetes is an incurable condition in which the body becomes resistant to insulin, a hormone which converts food into glucose needed for energy. Thirty-four million Americans live with Diabetes and almost 100 million are considered pre-diabetic. In addition, diabetes is a leading cause of blindness, amputation and kidney failure. Lack of gravity about diabetes, combined with inadequate access to health services and vital medicines, has caused the condition to become a universal problem with overwhelming human, social, and economic impact. Like other types of diabetes, gestational diabetes influences how the body's cells use glucose. Pregnant women are more inclined to develop this type of diabetes and high blood sugar that can affect the pregnancies negatively. Therefore, the goal of this study is to create a model that would accurately predict the onset of gestational diabetes. The population identified were women of Pima Indian heritage and at least 21 years of age. One study suggested that the highest prevalence of gestational diabetes can be found amongst the Pima Indians.

2 Variable Introduction

Although the data can be found on the Kaggle website, it originates from the National Institute of Diabetes and Digestive and Kidney Diseases. The data set consisted of 768 patients. The response variable was the Type II diabetes diagnosis (Yes=1 or No=0 later renamed as "Diabetic" and "nonDiab"). There were no categorical predictors and eight (8) continuous predictors: Number of pregnancies the patients have had, BMI (kg/sq.meter), insulin level (U/ml), age (years), Glucose level, Diabetes Pedigree Function, and Diastolic Blood pressure (mm Hg) and Skin thickness(mm).

3 Preprocessing of the Predictors

a. Missing Data

   Preprocessing data involves determining whether the data will require additions, deletions and or transformations. The processing began by determining if there were any missing values among the predictors or the response variable. In the event of missing values, imputation would be used. In this set there were no missing values; therefore, no imputation was necessary.

b. Deletions

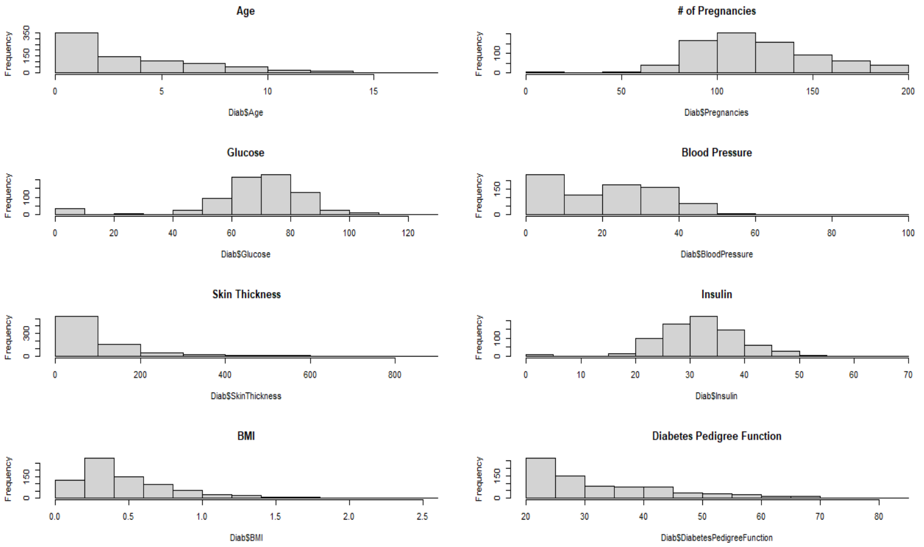
   Investigation of predictors that have zero or near zero variance is a useful procedure to determine if any predictors should be deleted. Because all predictors in this data set were continuous, there were no predictors with variances near zero. Another technique to determine if predictor reduction is necessary, is the application of a Principal Component Analysis (PCA). After the Pima Indian data was subjected to a PCA, the results revealed that 8 components were needed to explain 95% of the variance. For this reason, no predictors were removed.

c. Skewness

One of the assumptions of a linear relationship is that predictors have a symmetric distribution. The table below shows the skewness statistics for the original set:

Predictors	**Skewness**
Pregnancies	0.9
Glucose	0.17
BloodPressure	-1.8
SkinThickness	0.11
Insulin	2.2
BMI	-0.4
DiabetesPedigreeFunction	1.9
Age	1.1

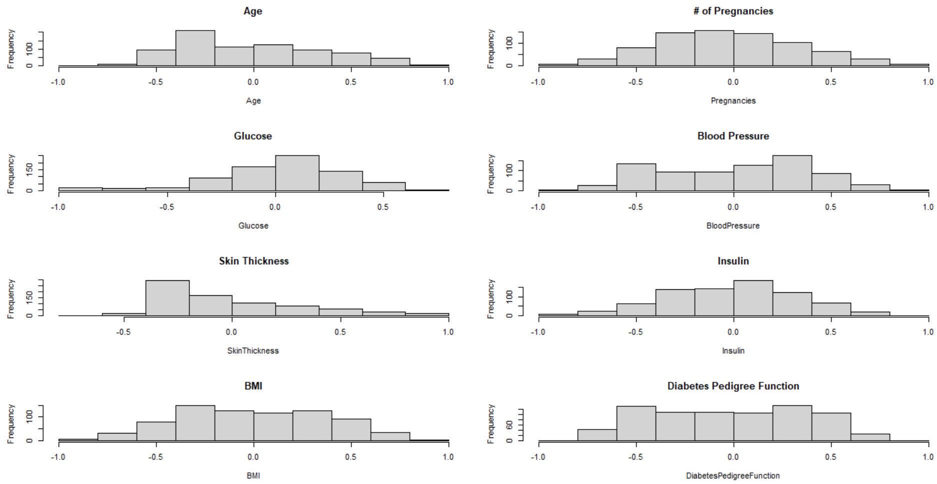
From the table, Glucose, SkinThickness and BMI are approximately symmetric. Pregnancies as a predictor is moderately symmetric. Finally, Blood Pressure is highly skewed to the left while the three variables are highly skewed to the right. Histograms were also created to visualize predictors that were skewed and to understand the distribution.



A Box-Cox transformation method was used to remove the skewness. The following table contains the transformed skewness values.

Predictors	Skewness	Transformed Skewness
Pregnancies	0.9	0.5
Glucose	0.17	0.2
BloodPressure	-1.8	-0.75
SkinThickness	0.11	-0.14
Insulin	2.2	0.96
BMI	-0.4	-0.06
DiabetesPedigreeFunction	1.9	0.02
Age	1.1	0.01

The following histogram after the transformation also shows that the distribution is fairly normal; symmetrical and not skewed.



d. Outliers

For linear models, outliers could bias responses and affect the performance of the model. In this data set, almost all predictors had a few outliers; however, Insulin and pedigree function had the greatest number. Consequently, the spatial sign method was used to remove the outliers. The following figures shows box plots before and after the removal of the outliers:

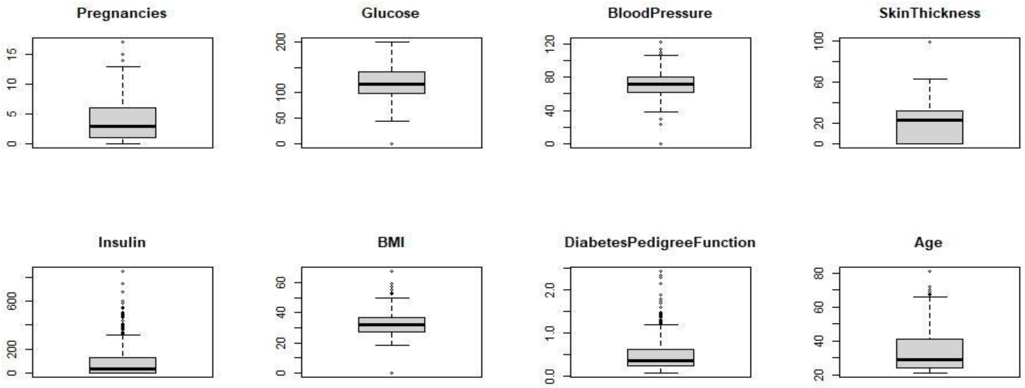


Figure A: Boxplots Before Spatial Sign Transformation

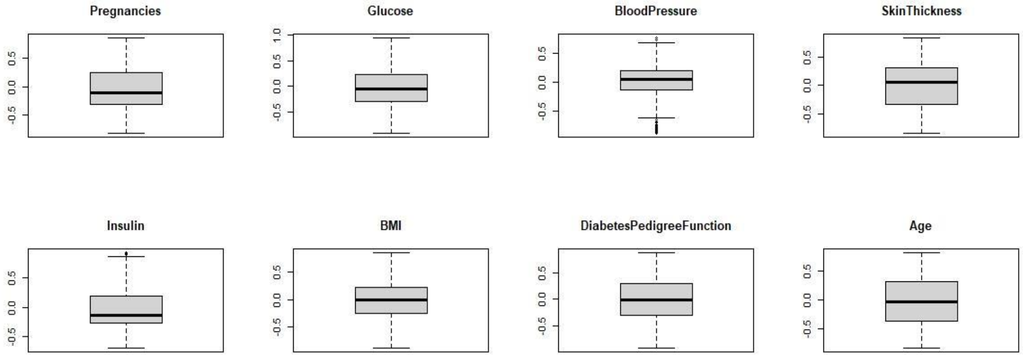
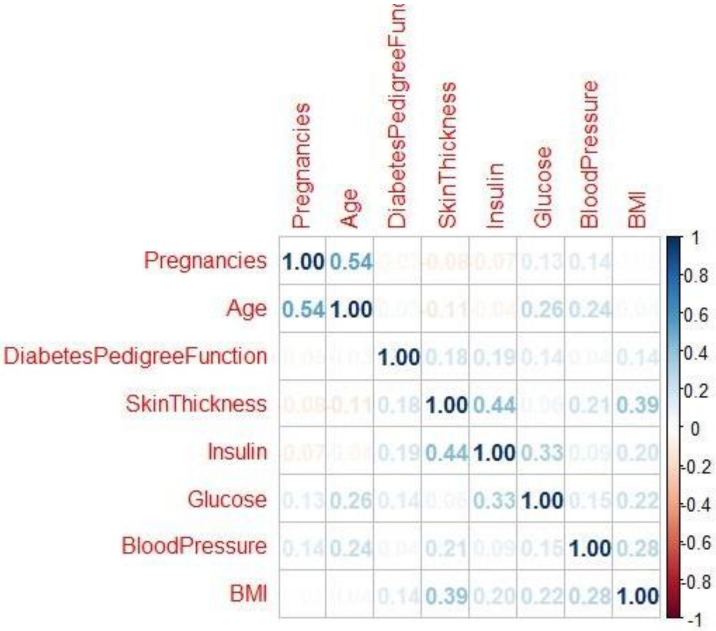


Figure B: Boxplots After Spatial Sign Transformation

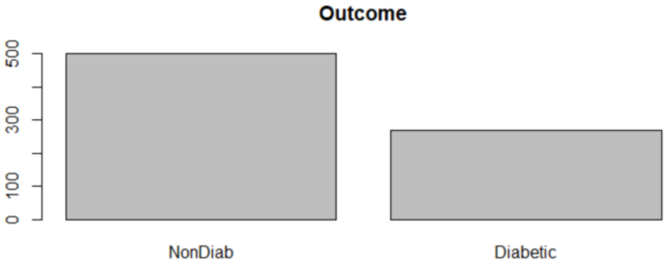
e. Correlations

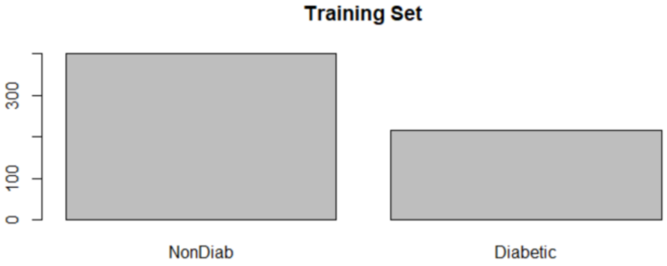
Another assumption of a linear relationship is that there is no multicollinearity. From the correlation matrix below, pairwise there appeared to be no highly correlated variables with *r*=0.85 as the threshold. Hence, no predictors were excluded.



4 Resampling Methodology

Stratified Random Sampling was used based on the response variable (Outcome) as the distribution among the classes in the response variable is imbalanced (shown below). The training set plot below shows that the classes have a similar distribution as the full data set. Since the sample size was small, the “leave-group-out” cross validation choice was made to avoid bias and maintain low variance.





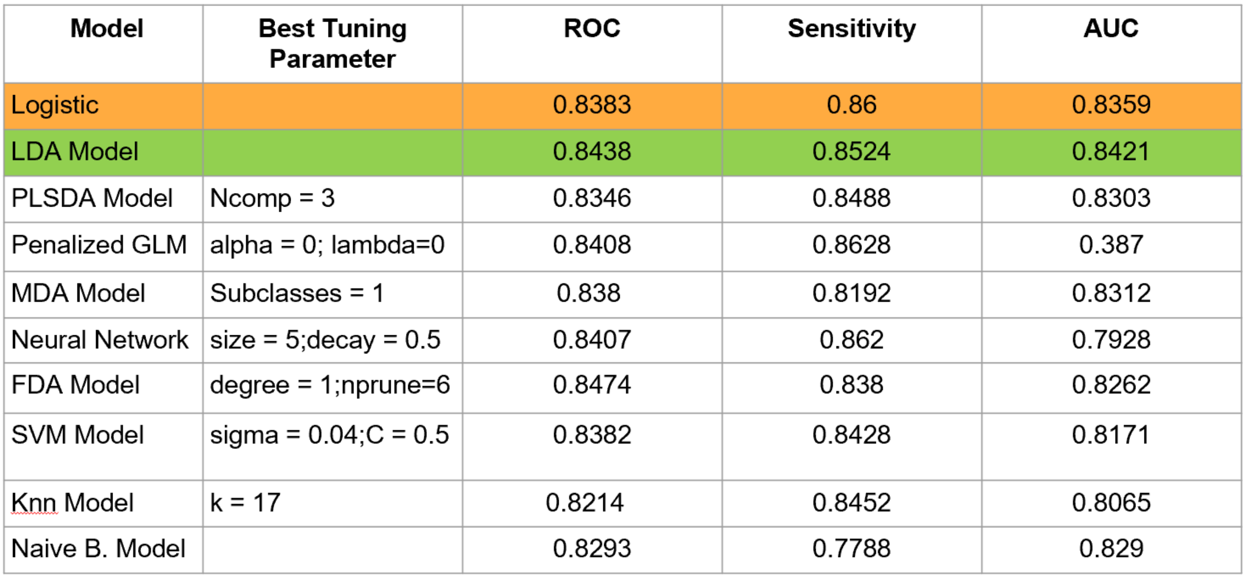
5 Data Splitting

After the data had been cleaned or preprocessed, the next step was to use R Code to place 80% of the data in the training set and 20% of the data in the testing set. The split resulted in 615 observations in the training set.

6 Model Fitting

a. Training Set

The following table summarizes the best tuning parameters and statistics for the corresponding linear and non-linear classification models.



Model	Best Tuning Parameter	ROC	Sensitivity	AUC
Logistic		0.8383	0.86	0.8359
LDA Model		0.8438	0.8524	0.8421
PLSDA Model	Ncomp = 3	0.8346	0.8488	0.8303
Penalized GLM	alpha = 0; lambda=0	0.8408	0.8628	0.387
MDA Model	Subclasses = 1	0.838	0.8192	0.8312
Neural Network	size = 5;decay = 0.5	0.8407	0.862	0.7928
FDA Model	degree = 1;nprune=6	0.8474	0.838	0.8262
SVM Model	sigma = 0.04;C = 0.5	0.8382	0.8428	0.8171
Knn Model	k = 17	0.8214	0.8452	0.8065
Naive B. Model		0.8293	0.7788	0.829

We are using AUC (area under curve) as the statistics to identify the best model as they represent the model that can distinguish Diabetes patients from non-patients. Based upon the metrics (AUC), the logistic and Linear Discriminant Analysis models were the best fit and would be used for further exploration.

b. Testing Data

We selected the Logistic and the LDA models from the list of models we tried on our training set. Both the Logistic and the LDA models were evaluated on the testing set and we would be using the AUC metric again to find the best model among them. Following table shows the results of the testing set evaluation:

Model	AUC	Sensitivity
Logistic	0.806	0.6226
LDA	0.8119	0.6226

Based on the metric (AUC), we would be selecting the LDA model as the best model. The sensitivity is also a good metric to find out if a person has diabetic or not. Although the sensitivity levels are not great, we have about 62% chance of rightly predicting patients with this model.

Confusion Matrix for the LDA model:

Linear Discriminant Analysis Model	**Observed**	
Prediction	NonDiab	Diabetic
NonDiab	84	20
Diabetic	16	33

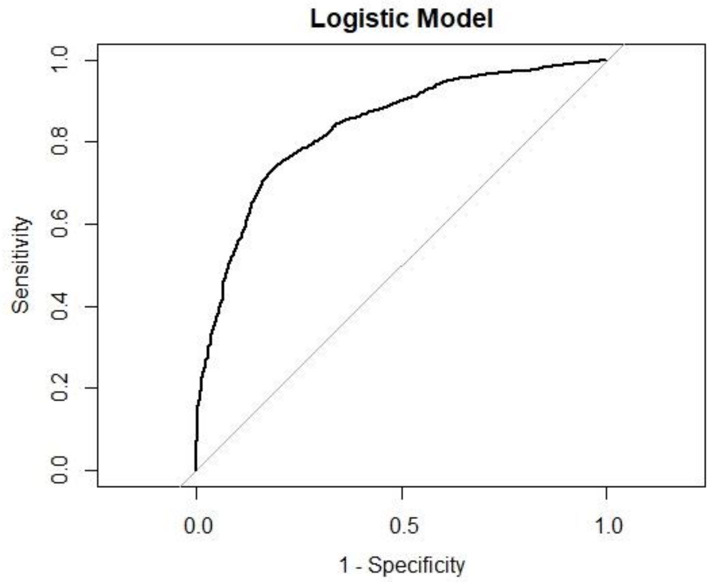
7 Summary

The final model chosen was the Linear Discriminant Analysis Model with an accuracy rate of 0.76, sensitivity of 0.6226, and the specificity of 0.84. Because the sensitivity was relatively low, the model would produce too many false negatives which is not particularly useful. The specificity, however, was considerably higher so the model would produce fewer false positive results which is one of the goals of a useful model.

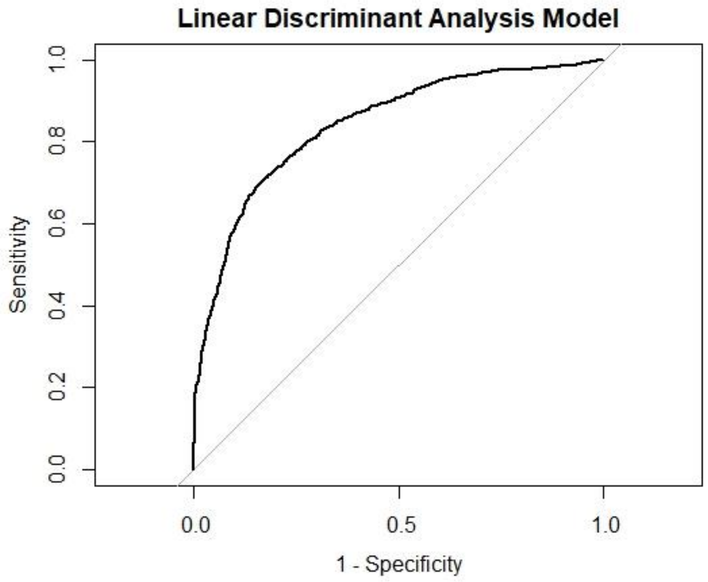
Being able to predict the onset of gestational diabetes among Pima women who were historically the most susceptible is hugely significant. Because at minimum, BMI and glucose levels can be controlled, preventive measures such as nutrition and weight counseling can be made widespread to the population to prevent future high-risk pregnancies.

Appendix 1: Supplemental Material for Linear Classification Models on Training Set

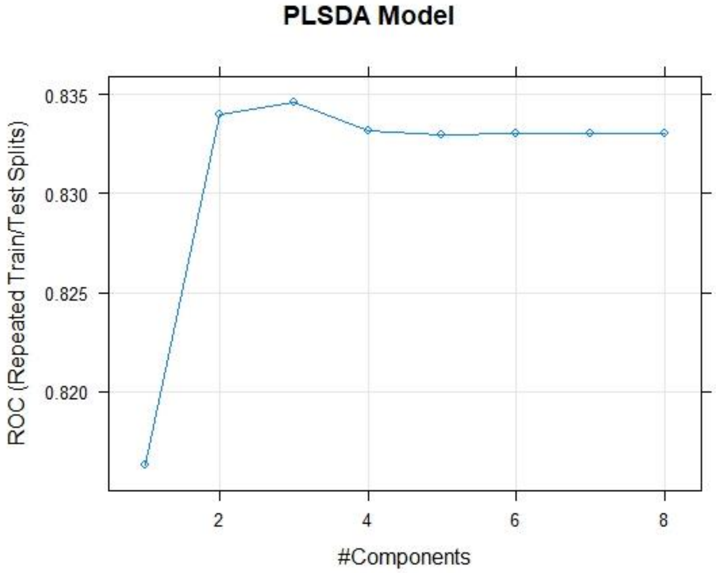
- Logistic Model
  AUC: 0.8359

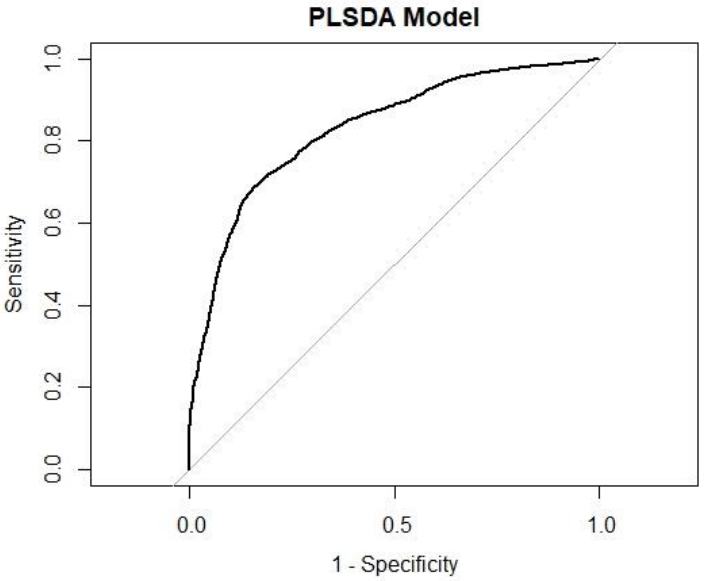


- Linear Discriminant Analysis Model
  AUC: 0.8421

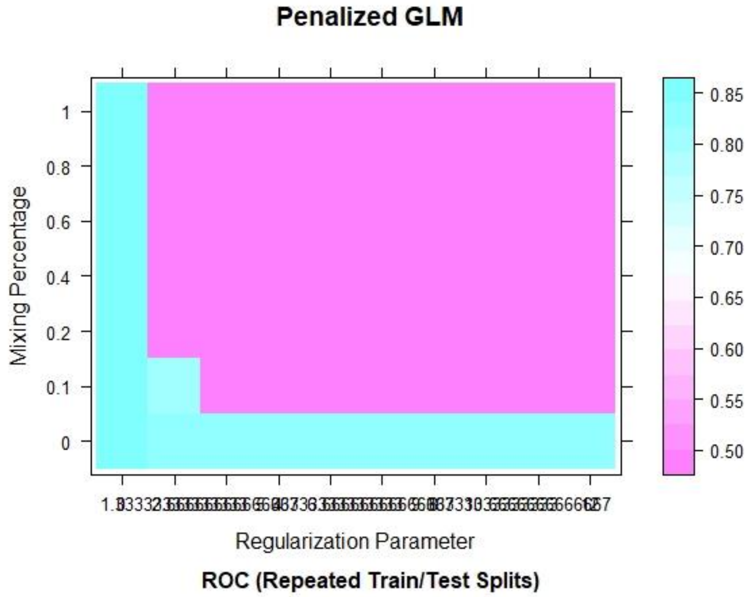


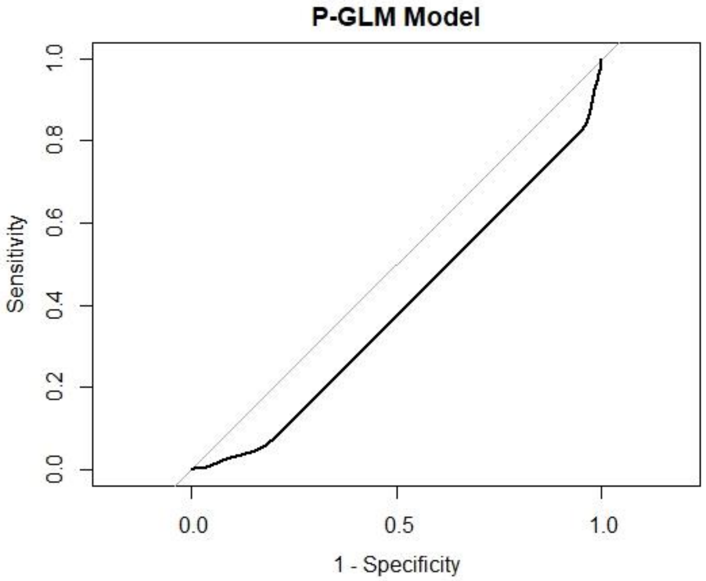
- Partial Least Squares Discriminant Analysis Model
  ncomp: 3
  AUC: 0.8303





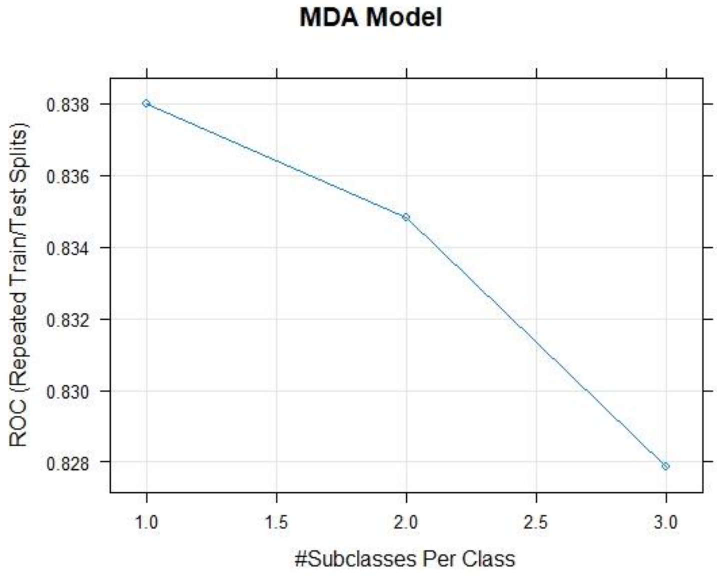
- Penalized Model
  alpha: 0; lambda: 0
  AUC: 0.387

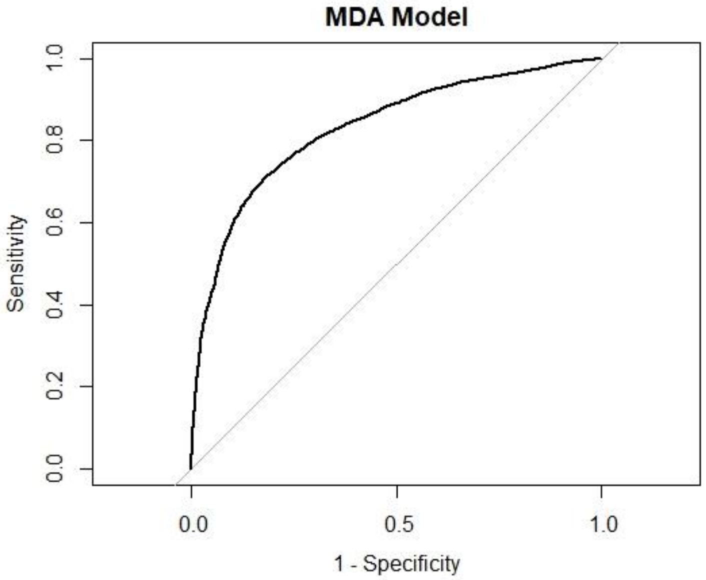




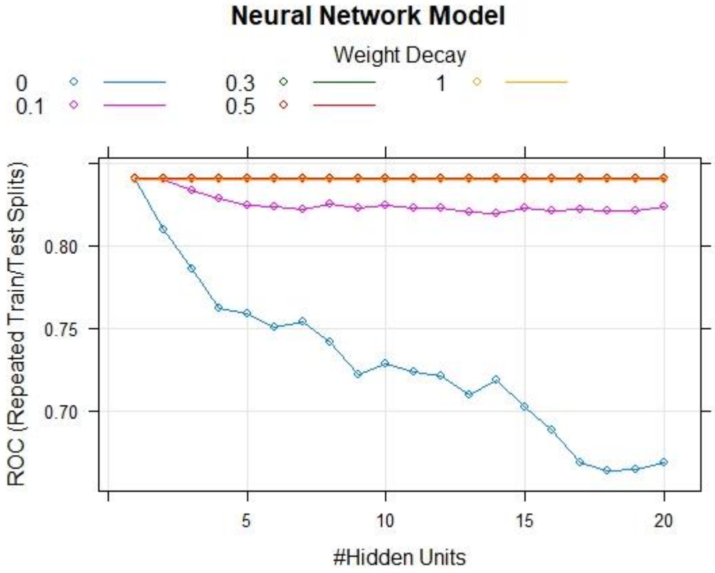
Appendix 2: Supplemental Material for non-Linear Classification Models

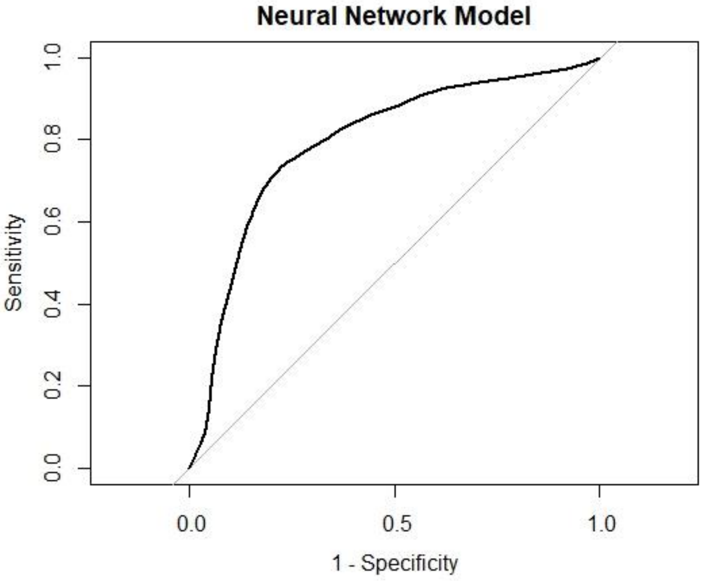
- Nonlinear Mixture Discriminant Analysis Model
  Subclasses: 1
  AUC: 0.8312



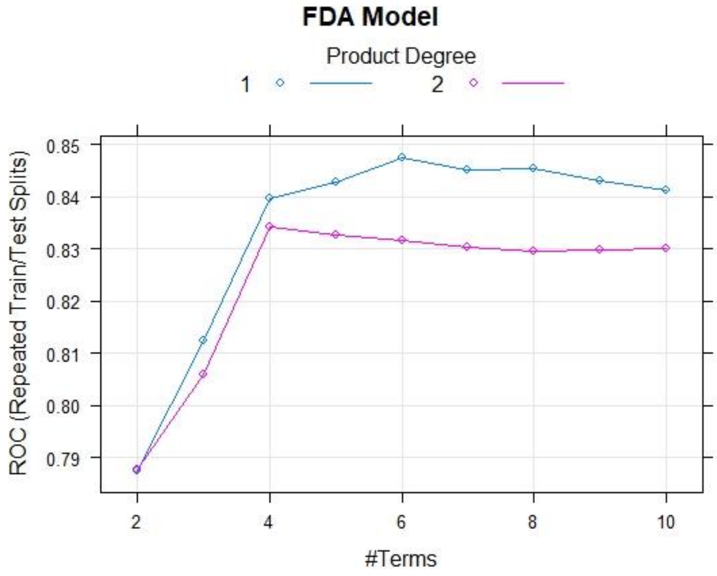


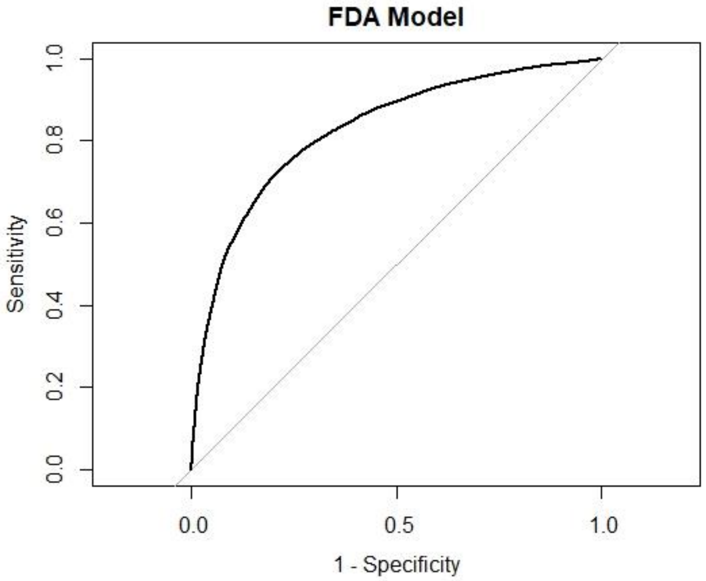
- Neural Network Model
  Size:5; decay:0.5
  AUC: 0.7928



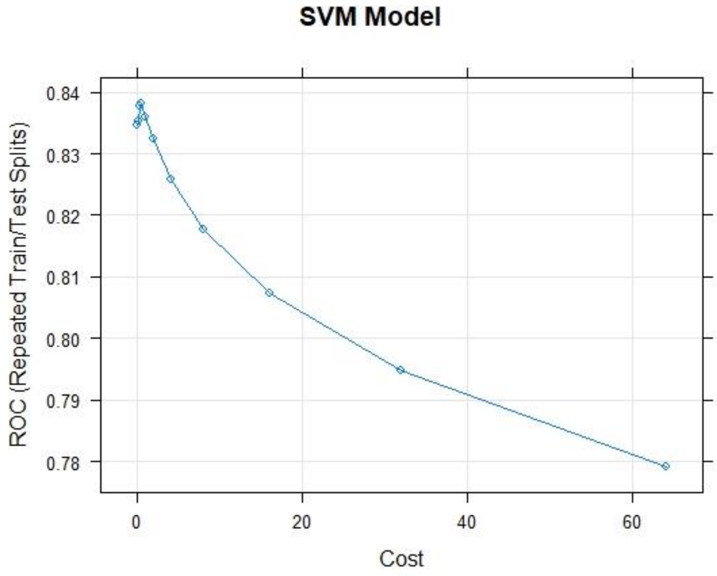


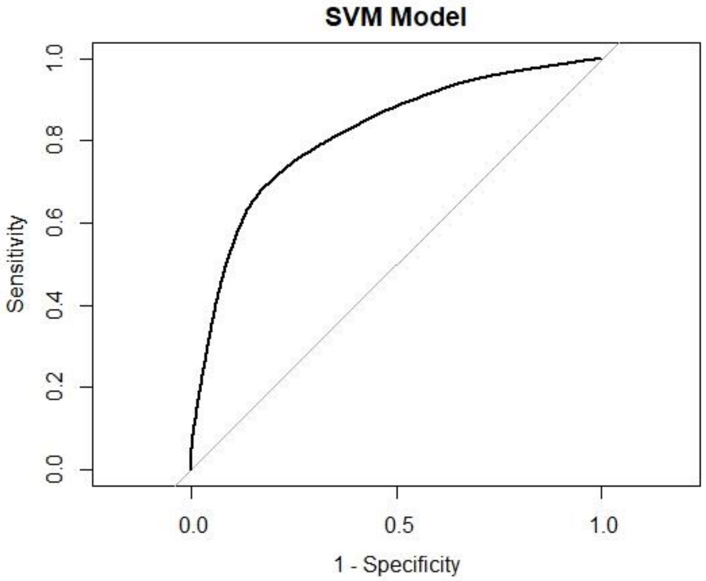
- Flexible Discriminant Analysis Model
  Degree:1; nprune:6
  AUC: 0.8262



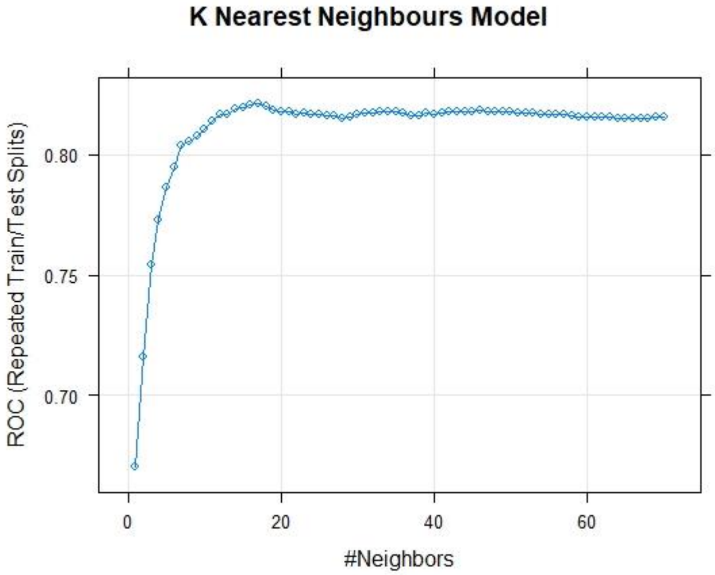


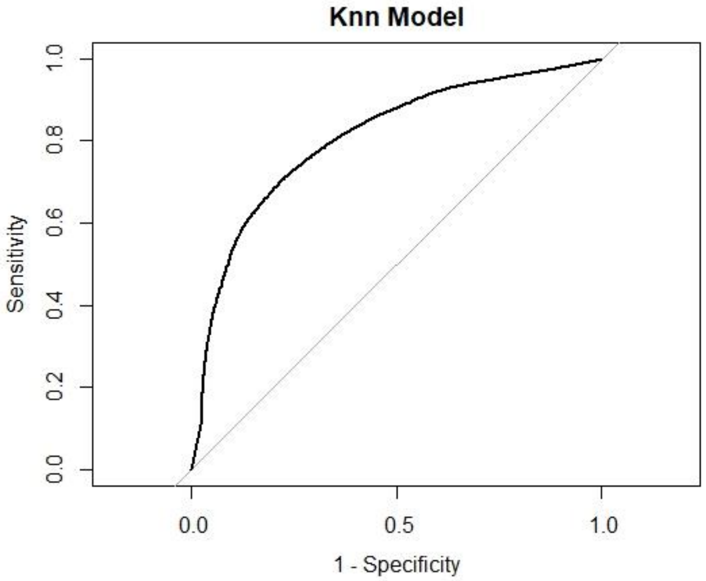
- Support Vector Machines Model
  sigma:0.04, C:0.5
  AUC: 0.8171



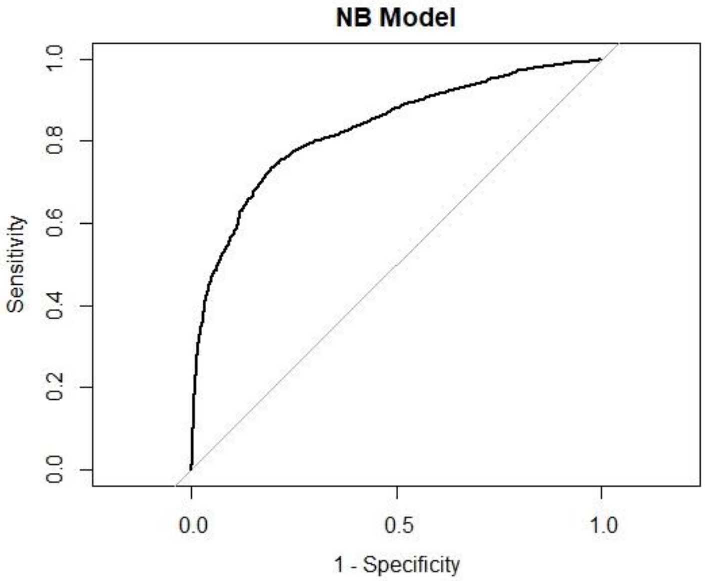


- K-Nearest Neighbors Model
  k:17
  AUC: 0.8065



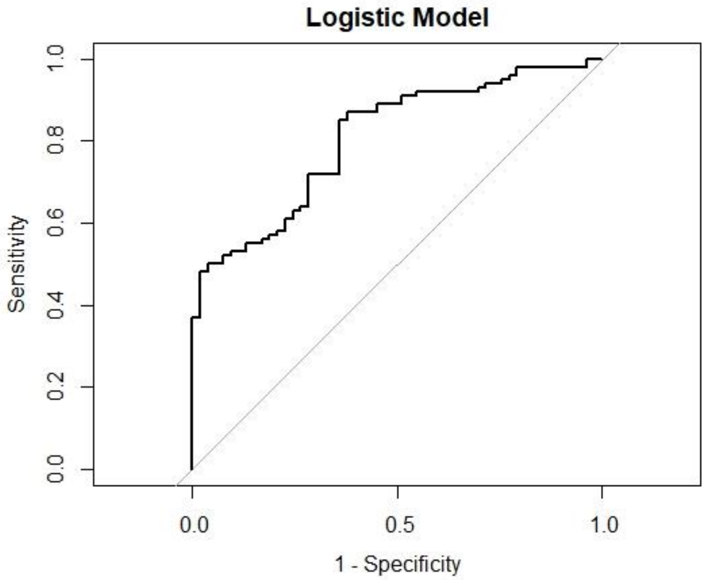


- Naive Bayes
  AUC: 0.829

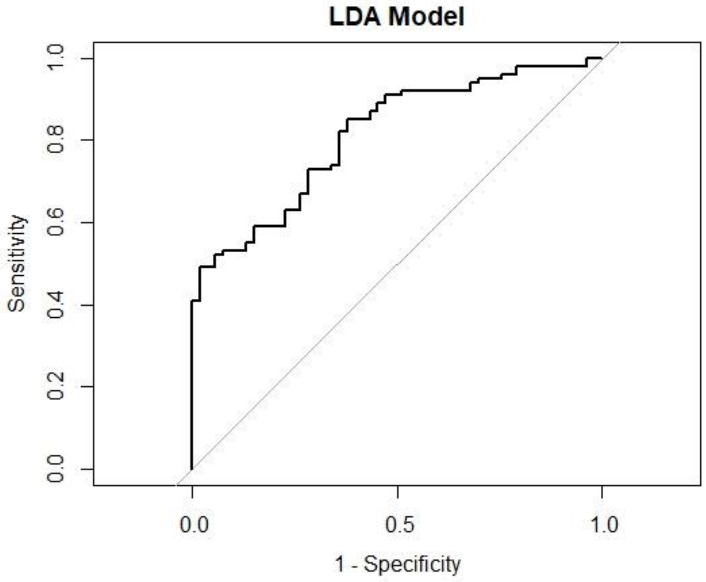


Appendix 3: Supplemental Material for Models on Testing Set

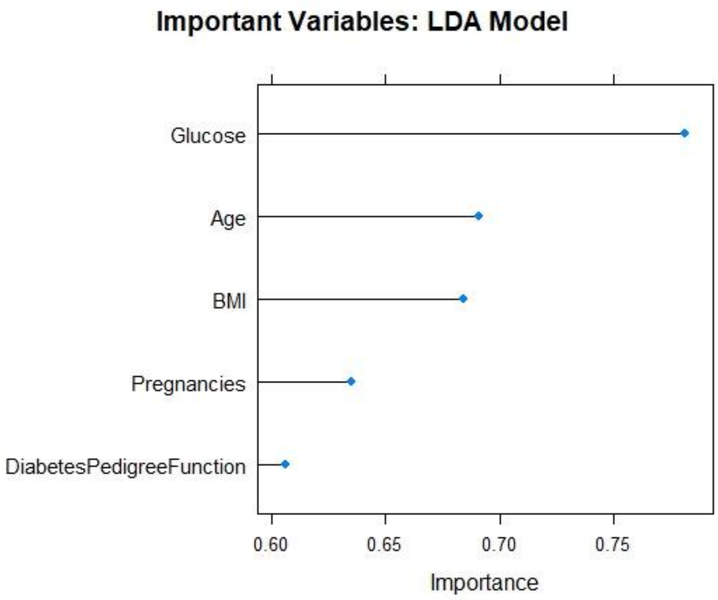
- Logistic Model
  AUC: 0.806



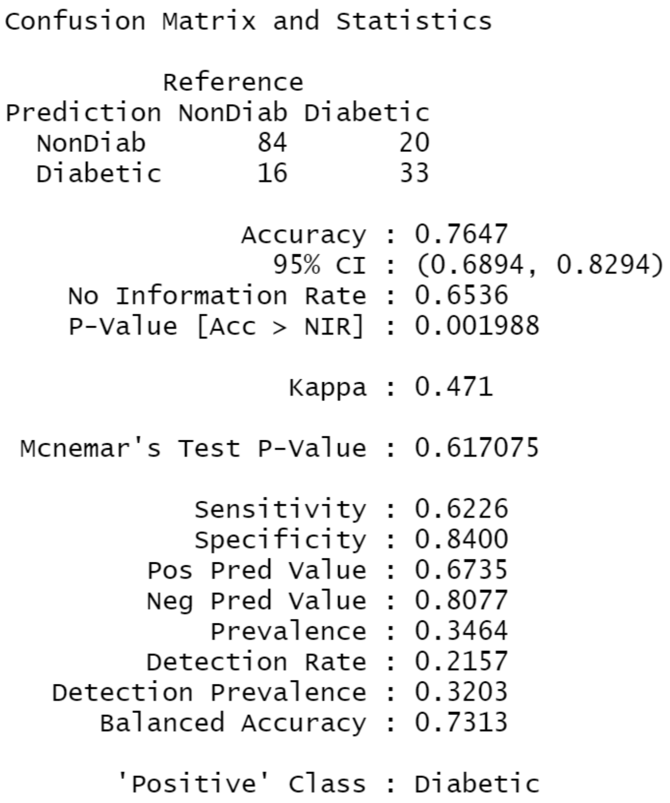
- LDA Model
  AUC: 0.8119



Consequently, it was deemed that the LDA model is the best model. The five best predictors for this model were not unusual:



The confusion matrix revealed a fairly high specificity with a specificity of 0.84. The percentage of false positives produced in the testing set would be 16%.



```
Confusion Matrix and Statistics

          Reference
Prediction NonDiab Diabetic
  NonDiab       84       20
  Diabetic      16       33
                                          
               Accuracy : 0.7647          
                 95% CI : (0.6894, 0.8294)
    No Information Rate : 0.6536          
    P-Value [Acc > NIR] : 0.001988        
                                          
                  Kappa : 0.471           
                                          
 Mcnemar's Test P-Value : 0.617075        
                                          
            Sensitivity : 0.6226          
            Specificity : 0.8400          
         Pos Pred Value : 0.6735          
         Neg Pred Value : 0.8077          
             Prevalence : 0.3464          
         Detection Rate : 0.2157          
   Detection Prevalence : 0.3203          
      Balanced Accuracy : 0.7313          
                                          
       'Positive' Class : Diabetic        
```

R Code

```
##Install and load required packages----
#install.packages(c("caret","e1071"))
library(caret)
library(e1071)
library(corrplot)
library(pls)
library(dplyr)
library(pROC)
library(nnet)
library(mda)
library(earth)
library(MASS)
library(kernlab)
library(klaR)

#Read the data file
#set working directory where the R and csv file is saved
Diab <- read.csv("diabetes.csv")

# Check Data --------------------------------------------------------

#checking for near zero variance:
nearZeroVar(Diab)
# 0 means no variable with near zero variance

#check for any missing values:
sum(is.na(Diab))
#0 means no missing values, so we don't have to impute values

#checking for distribution/center and skewness:
#Histogram plots of the 8 predictors, to show distribution and skewness
par(mfrow=c(4,2))
hist(Diab[,1], xlab = "Age", ylab = "Frequency", main = "Age")
hist(Diab[,2], xlab = "Pregnancies", ylab = "Frequency", main = "# of Pregnancies")
hist(Diab[,3], xlab = "Glucose", ylab = "Frequency", main = "Glucose")
hist(Diab[,4], xlab = "BloodPressure", ylab = "Frequency", main = "Blood Pressure")
hist(Diab[,5], xlab = "SkinThickness", ylab = "Frequency", main = "Skin Thickness")
hist(Diab[,6], xlab = "Insulin", ylab = "Frequency", main = "Insulin")
hist(Diab[,7], xlab = "BMI", ylab = "Frequency", main = "BMI")
hist(Diab[,8], xlab = "DiabetesPedigreeFunction", ylab = "Frequency", main = "Diabetes
Pedigree Function")
#many variables are not centered and has a skew in the distribution

#checking for skewness:
skewValues <- apply(Diab[1:8], 2, skewness)
head(skewValues)  #for all variables
#couple of the variables are extremely skewed

#checking for correlated predictors:
par(mfrow=c(1,1))
segDiab <- Diab[1:8]
correlations <- cor(segDiab)
corrplot(correlations, order = "hclust", method = 'number')
highCorr <- findCorrelation(correlations, cutoff = .85)
length(highCorr) # zero - no correlated predictors
#no two predictors seems to be highly correlated and hence none of the variables are removed

#checking for outliers:
par(mfrow=c(2,4))
for(xx in 1:8){
  boxplot(Diab[xx],main = colnames(Diab[xx]))
}
#many variables shows strong outliers

#check if the distribution among the response variable classes is not vastly different
par(mfrow=c(1,1))
hist(Diab[,9], xlab = "Outcome", ylab = "Frequency", main = "Diabetes Outcome")
#the distribution is unbalanced and hence use stratified sampling with ROC

# Transform and Plot -----------------------------------------------------------

#Transform the data to resolve issues
trans <- preProcess(Diab[,1:8], method = c("nzv", "BoxCox", "center", "scale", "spatialSign"))
## need {caret} package
tDiab <- predict(trans, Diab[,1:8])
XX <- tDiab
YY <- as.factor(Diab[,9])
levels(YY) <- c("NonDiab","Diabetic")
```

```
#checking for distribution/center and skewness after the transformation:
#Histogram plots of the 8 predictors, to show distribution and skewness
par(mfrow=c(4,2))
hist(tDiab[,1], xlab = "Age", ylab = "Frequency", main = "Age")
hist(tDiab[,2], xlab = "Pregnancies", ylab = "Frequency", main = "# of Pregnancies")
hist(tDiab[,3], xlab = "Glucose", ylab = "Frequency", main = "Glucose")
hist(tDiab[,4], xlab = "BloodPressure", ylab = "Frequency", main = "Blood Pressure")
hist(tDiab[,5], xlab = "SkinThickness", ylab = "Frequency", main = "Skin Thickness")
hist(tDiab[,6], xlab = "Insulin", ylab = "Frequency", main = "Insulin")
hist(tDiab[,7], xlab = "BMI", ylab = "Frequency", main = "BMI")
hist(tDiab[,8], xlab = "DiabetesPedigreeFunction", ylab = "Frequency", main = "Diabetes
Pedigree Function")

#checking for skewness after the transformation:
skewValues <- apply(tDiab[1:8], 2, skewness)
head(skewValues)

#checking for outliers after the transformation:
par(mfrow=c(2,4))
for(xx in 1:8){
  boxplot(tDiab[xx],main = colnames(Diab[xx]))
}

# Training Data -----------------------------------------------------------

#Data Splitting
set.seed(1)
trainingRows <- createDataPartition(YY, p = .80, list= FALSE)
trainPredictors <- data.frame(XX[trainingRows,])
trainClasses <- YY[trainingRows]

testPredictors <- data.frame(XX[-trainingRows,])
testClasses <- YY[-trainingRows]

ctrl <- trainControl(method = "LGOCV",
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE,
                     savePredictions = TRUE)
```

```
#Logistic Model
Diab_logistic = train( trainPredictors, trainClasses, method="glm",
                metric="ROC", trControl=ctrl )
Diab_logistic
par(mfrow=c(1,1))
FullRoc_logistic <- roc(response = Diab_logistic$pred$obs,
          predictor = Diab_logistic$pred$Diabetic,
          levels = rev(levels(Diab_logistic$pred$obs)))
plot(FullRoc_logistic, legacy.axes = TRUE, main = "Logistic Model")
auc(FullRoc_logistic)

#Linear Discriminant Analysis Model
Diab_lda <- train(trainPredictors, trainClasses, method = "lda",
            trControl = ctrl, metric = "ROC",
            preProc=c("center","scale"))
Diab_lda
par(mfrow=c(1,1))
FullRoc_LDA <- roc(response = Diab_lda$pred$obs,
                 predictor = Diab_lda$pred$Diabetic,
                 levels = rev(levels(Diab_lda$pred$obs)))
plot(FullRoc_LDA, legacy.axes = TRUE, main = "Linear Discriminant Analysis Model")
auc(FullRoc_LDA)

#Partial Least Squares Discriminant Analysis Model
Diab_plsda = train( trainPredictors, trainClasses, method="pls",
              tuneGrid=expand.grid(.ncomp=1:8),
              preProc=c("center","scale"),
              metric="ROC", trControl=ctrl )
Diab_plsda
plot(Diab_plsda, main = "PLSDA Model")
par(mfrow=c(1,1))
FullRoc_plsda <- roc(response = Diab_plsda$pred$obs,
             predictor = Diab_plsda$pred$Diabetic,
             levels = rev(levels(Diab_plsda$pred$obs)))
plot(FullRoc_plsda, legacy.axes = TRUE, main = "PLSDA Model")
auc(FullRoc_plsda)

#Penalized Model
glmnGrid <- expand.grid(.alpha = c(0, .1, .2, .4, .6, .8, 1),
                 .lambda = seq(0,12, length = 10))
```

```
Diab_pglm = train( trainPredictors, trainClasses,
               method="glmnet", tuneGrid=glmnGrid,
               preProc=c("center","scale"), metric="ROC",
               trControl=ctrl )
Diab_pglm
plot(Diab_pglm, plotType = "level", main = "Penalized GLM")
par(mfrow=c(1,1))
FullRoc_pglm <- roc(response = Diab_pglm$pred$obs,
                predictor = Diab_pglm$pred$Diabetic,
                levels = rev(levels(Diab_pglm$pred$obs)))
plot(FullRoc_pglm, legacy.axes = TRUE, main = "P-GLM Model")
auc(FullRoc_pglm)

#Nonlinear Mixture Discriminant Analysis Model
set.seed(476)
Diab_mda <- train(trainPredictors, trainClasses,
            method = "mda",
            metric = "ROC",
            tuneGrid = expand.grid(.subclasses = 1:3),
            trControl = ctrl)
Diab_mda
plot(Diab_mda, main = "MDA Model")
par(mfrow=c(1,1))
FullRoc_mda <- roc(response = Diab_mda$pred$obs,
               predictor = Diab_mda$pred$Diabetic,
               levels = rev(levels(Diab_mda$pred$obs)))
plot(FullRoc_mda, legacy.axes = TRUE, main = "MDA Model")
auc(FullRoc_mda)

#Neural Network Model
nnetGrid <- expand.grid(.size = 1:20, .decay = c(0, .1, .3, .5, 1))
maxSize <- max(nnetGrid$.size)
numWts <- (maxSize * (8 + 1) + (maxSize+1)*2) ## 8 is the number of predictors; 2 is the
number of classes
Diab_nnet <- train(trainPredictors, trainClasses,
             method = "nnet",
             metric = "ROC",
             preProc = c("center", "scale", "spatialSign"),
             tuneGrid = nnetGrid,
             trace = FALSE,
                   maxit = 2000,
                   MaxNWts = numWts,
                   trControl = ctrl)
Diab_nnet
plot(Diab_nnet, main = "Neural Network Model")
par(mfrow=c(1,1))
FullRoc_nnet <- roc(response = Diab_nnet$pred$obs,
                    predictor = Diab_nnet$pred$Diabetic,
                    levels = rev(levels(Diab_nnet$pred$obs)))
plot(FullRoc_nnet, legacy.axes = TRUE, main = "Neural Network Model")
auc(FullRoc_nnet)

#Flexible Discriminant Analysis Model
marsGrid <- expand.grid(.degree = 1:2, .nprune = 2:10)
Diab_fda <- train(trainPredictors, trainClasses,
                  method = "fda",
                  metric = "ROC",
                  tuneGrid = marsGrid,
                  trControl = ctrl)
Diab_fda
plot(Diab_fda, main = "FDA Model")
par(mfrow=c(1,1))
FullRoc_fda <- roc(response = Diab_fda$pred$obs,
                   predictor = Diab_fda$pred$Diabetic,
                   levels = rev(levels(Diab_fda$pred$obs)))
plot(FullRoc_fda, legacy.axes = TRUE, main = "FDA Model")
auc(FullRoc_fda)

#Support Vector Machines Model
sigmaRangeReduced <- sigest(as.matrix(trainPredictors))
svmRGridReduced <- expand.grid(.sigma = sigmaRangeReduced[1],
                               .C = 2^(seq(-4, 6)))
set.seed(476)
Diab_svmR <- train(trainPredictors, trainClasses,
                   method = "svmRadial",
                   metric = "ROC",
                   preProc = c("center", "scale"),
                   tuneGrid = svmRGridReduced,
                   fit = FALSE,
                   trControl = ctrl)
```

```
Diab_svmR
plot(Diab_svmR, main = "SVM Model")
par(mfrow=c(1,1))
FullRoc_svm <- roc(response = Diab_svmR$pred$obs,
             predictor = Diab_svmR$pred$Diabetic,
             levels = rev(levels(Diab_svmR$pred$obs)))
plot(FullRoc_svm, legacy.axes = TRUE, main = "SVM Model")
auc(FullRoc_svm)

#K-Nearest Neighbors Model
set.seed(476)
Diab_knn <- train(trainPredictors, trainClasses,
           method = "knn",
           metric = "ROC",
           preProc = c("center", "scale"),
           tuneGrid = data.frame(.k = 1:70),
           trControl = ctrl)
Diab_knn
plot(Diab_knn, main = "K Nearest Neighbours Model")
par(mfrow=c(1,1))
FullRoc_knn <- roc(response = Diab_knn$pred$obs,
             predictor = Diab_knn$pred$Diabetic,
             levels = rev(levels(Diab_knn$pred$obs)))
plot(FullRoc_knn, legacy.axes = TRUE, main = "Knn Model")
auc(FullRoc_knn)

#Naive Bayes
set.seed(476)
Diab_nb <- train( trainPredictors, trainClasses,
           method = "nb",
           metric = "ROC",
           preProc = c("center", "scale"),
           tuneGrid = data.frame(.fL = 2,.usekernel = TRUE,.adjust = TRUE),
           trControl = ctrl)

Diab_nb
par(mfrow=c(1,1))
FullRoc_nb <- roc(response = Diab_nb$pred$obs,
             predictor = Diab_nb$pred$Diabetic,
             levels = rev(levels(Diab_nb$pred$obs)))
```

```
plot(FullRoc_nb, legacy.axes = TRUE, main = "NB Model")
auc(FullRoc_nb)

# Testing Data ----------------------------------------------------------

#Logistic Model
Diab_logistic_pred_prob = predict( Diab_logistic, testPredictors, type = "prob")
Diab_logistic_pred = predict( Diab_logistic, testPredictors)
confusionMatrix(Diab_logistic_pred, testClasses, positive = "Diabetic")
par(mfrow=c(1,1))
TestRoc_logistic <- roc(response = testClasses,
                    predictor = Diab_logistic_pred_prob$Diabetic, levels=c("Diabetic",
"NonDiab"))
plot(TestRoc_logistic, legacy.axes = TRUE, main = "Logistic Model")
auc(TestRoc_logistic)

#Linear Discriminant Analysis Model
Diab_lda_pred_prob <- predict(Diab_lda, testPredictors, type = "prob")
Diab_lda_pred <- predict(Diab_lda, testPredictors)
confusionMatrix(Diab_lda_pred, testClasses, positive = "Diabetic")
par(mfrow=c(1,1))
TestRoc_lda <- roc(response = testClasses,
                    predictor = Diab_lda_pred_prob$Diabetic, levels=c("Diabetic", "NonDiab"))
plot(TestRoc_lda, legacy.axes = TRUE, main = "LDA Model")
auc(TestRoc_lda)

#Important Predictor Variables
plot(varImp(Diab_lda, scale = FALSE), top = 5, scales = list(y = list(cex = .95)), main = "
Important Variables: LDA Model")
```