MIS 6356.005 Business Analytics with R – S19

Final Report

Pima Indians Diabetes Prediction

Group 7

Adityan Rajendran

Lixuan Chen

Ming Wei

Shalini Singh

Siddharth Govindarajan

Table of Contents

Pima Indians Diabetes Prediction

Executive Summary

A research conducted by National Institute of Diabetes and Digestive and Kidney diseases (NIH), one of the prominent medical research centers, shows that diabetes affects 30.30 million people closely in United States. Also, NIH claims that diabetes is the seventh leading cause of death. U.S. Department of Health & Human Services, a government organization reports 15 million women in the United States have diabetes. Therefore, this project aims to awake our awareness of diabetes and help detect whether a patient has diabetes or not.

The dataset that we have chosen is originally from the 'National Institute of Diabetes and Digestive and Kidney Diseases', Maryland, US and is named as 'Pima Indians Diabetes Database'. The dataset consists of eight medical predictor variables and one target variable, '*Outcome*'. The predictor variables include the number of pregnancies the patient has had, their BMI (Body Mass Index) level, insulin levels, age, blood pressure, glucose, skin thickness, and diabetes pedigree function.

We used R, a statistical computing and graphics tool for building our BI model. At first, we explored the dataset to understand our dataset and develop a general idea for further analysis. We find the correlations among variables, for example, there is a negative relationship between pregnancies and skin thickness; also, the distribution of BMI is symmetric.

Then, we use this dataset to build classification models, including decision tree model and logistic regression model, to predict whether a given female patient, with certain diagnostic measurements, has diabetes or not. For the decision tree model, we addressed usual zeroes, sampled 60% of records as training data sets and 40% of records as validation data sets, plotted the decision tree, and evaluated the decision tree model using confusion matrix and ROC. For logistic regression model, we completed data imputation, sampled training data sets and validation data sets, built the logistic regression model, computed odds ratios, and evaluated the logistic regression model using confusion matrix and ROC.

After running and evaluating both the models, we concluded that decision tree model is better than logistic regression model in our project. Not only because decision tree model has a higher accuracy rate in confusion matrix and a higher area in ROC than logistic regression model has, but also because logistic regression model relies largely on the independent variables, if we include wrong independent variables, there is no predictive value using this model. Also, decision tree is easy to implement and interpret.

Project Background

Diabetes affects an estimated 30.3 million people in the United States and is the seventh leading cause of death. Diabetes can affect many parts of the body and is associated with serious complications, such as heart disease and stroke, blindness, kidney failure, and lower limb amputation. In addition to increasing the risk for these complications, diabetes also doubles the risk for many forms of cancer, some forms of dementia, hearing loss, erectile dysfunction, urinary incontinence, and many other common diseases.

The Pima Indians, on whom the dataset is based, have the highest rate of type two diabetes in the world. While biomedical studies have identified a genetic variable associated with the high prevalence of diabetes among Pima Indians, genetics is only one factor that encompasses an individual's risk for developing a disease. Information on other medical factors relating to the development of type two diabetes amongst this population is necessary. Though Pima Indians have the highest incidence of Diabetes in the world, Pima women are chosen as the main focus of our study because incidence rate of Pima women is 40.8% as compared to 34.2% for Pima men.

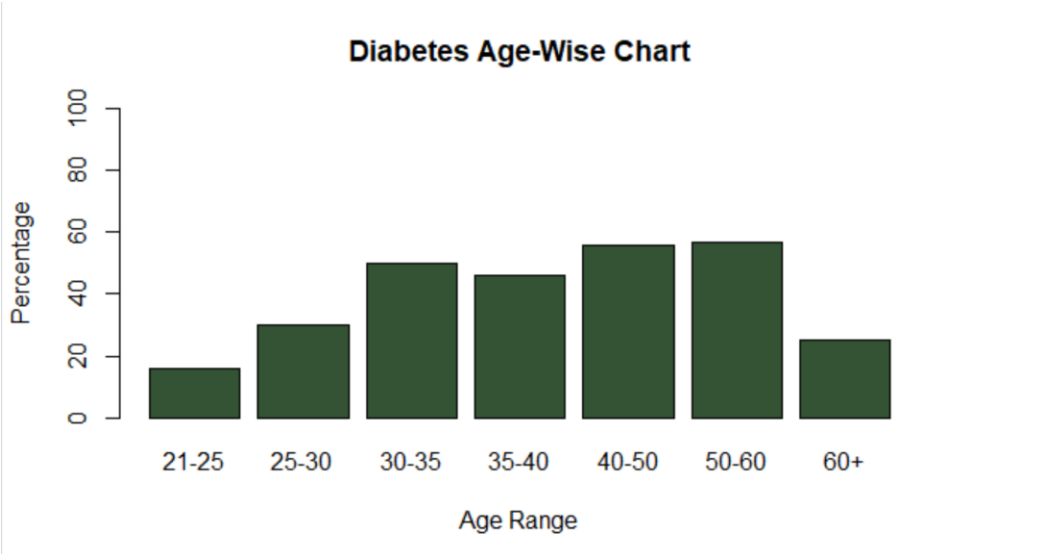
Hence, this project is proposed to be built to study the medical factors and diagnostics of Pima Indian heritage females above the age of 21 and develop a classification model which could predict the incidence of diabetes among them given a set of medical diagnostic measurements.

Data Selection

The dataset that we have chosen is originally from the 'National Institute of Diabetes and Digestive and Kidney Diseases', Maryland, US and is named as 'Pima Indians Diabetes Database'. The dataset was obtained from Kaggle (www.kaggle.com) and several constraints were placed on the selection of these instances from a larger database.

National Institute of Diabetes and Digestive and Kidney Diseases conduced a diabetes test for 768 women who were at least 21 years old of Pima Indian heritage. The test had 657 women who had been pregnant at least once in their life and 111 women who were never been pregnant. In the test taken, women ages 50-60 had the maximum percentage of diabetes. Women chosen were among this following age range (R codes for tables below attached in R scripts):

Age	Number of Women
21-25	267
25-30	150
30-35	81
35-40	76
40-50	113
50-60	54
>60	27



The dataset consists of certain diagnostic measurements of female patients from the 'Pima Indian Heritage' and if they currently have diabetes or not. The dataset consists of several medical predictor variables and one target variable, '*Outcome*'. The predictor

variables include the number of pregnancies the patient has had, their BMI (Body Mass Index) level, insulin levels, age, and so on.

Approaching the dataset in a technical perspective, the dataset is in a 'Comma Separated' file – 'diabetes.csv'. We have all female patients at least 21 years old of Pima Indian Heritage. The list of independent and target variables is listed below with a brief description.

Independent Variables

Independent Variables	Description
Pregnancies	Number of times the patient has had a pregnancy
Glucose	Plasma glucose concentration in a 2 hours oral glucose tolerance test
BloodPressure	Diastolic blood pressure (mm Hg)
SkinThickness	Triceps skin fold thickness (mm)
Insulin	2-Hour serum insulin (mu U/ml)
BMI	Body mass index (weight in kg) (height in meter^2)
DiabetesPedigreeFunction	Diabetes Pedigree Function
Age	Age of the patient (in years)

Target/Dependent Variable

Target/Dependent Variable	Description
Outcome	Class variable (0 or 1) (1 – Diabetes present, 0 – No Diabetes)

Pima Indians Diabetes Prediction

BI Model

Data Visualization and Exploration

Firstly, we present data in a form that makes sense to us so that we can have a general idea about the data and find directions for further analysis. We detect correlations among these variables and the distribution of variables.

```
diabetes.df <- read.csv("diabetes.csv")
summary(diabetes.df)
```

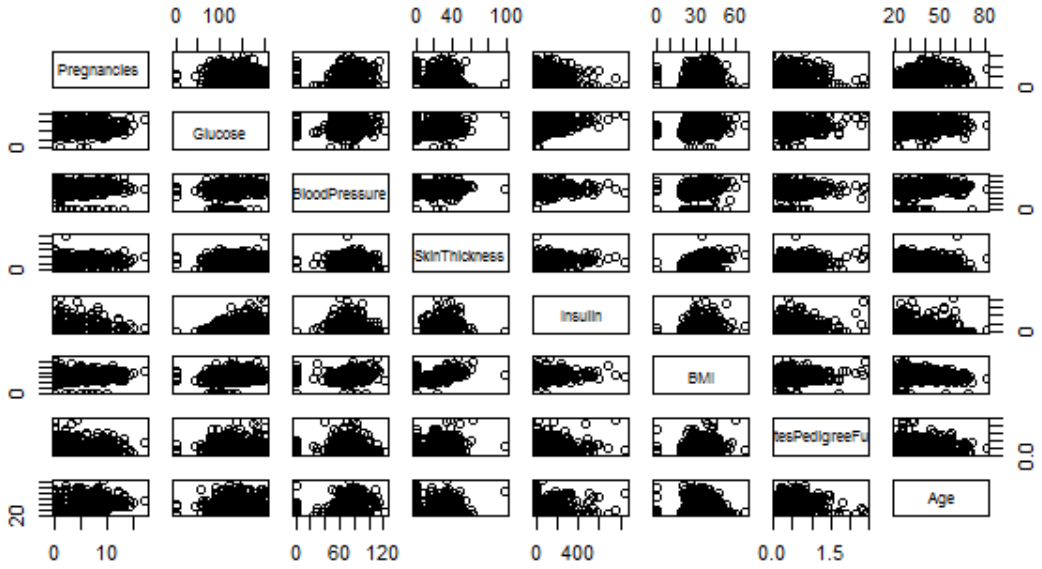
```
  Pregnancies        Glucose      BloodPressure    SkinThickness      Insulin
 Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00   Min.   :  0.0
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00   1st Qu.:  0.0
 Median : 3.000   Median :117.0   Median : 72.00   Median :23.00   Median : 30.5
 Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54   Mean   : 79.8
 3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00   3rd Qu.:127.2
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00   Max.   :846.0
      BMI        DiabetesPedigreeFunction      Age            Outcome
 Min.   : 0.00   Min.   :0.0780           Min.   :21.00   Min.   :0.000
 1st Qu.:27.30   1st Qu.:0.2437           1st Qu.:24.00   1st Qu.:0.000
 Median :32.00   Median :0.3725           Median :29.00   Median :0.000
 Mean   :31.99   Mean   :0.4719           Mean   :33.24   Mean   :0.349
 3rd Qu.:36.60   3rd Qu.:0.6262           3rd Qu.:41.00   3rd Qu.:1.000
 Max.   :67.10   Max.   :2.4200           Max.   :81.00   Max.   :1.000
```

```
cor(diabetes.df[,sapply(diabetes.df,is.numeric)],use = "complete.obs")
```

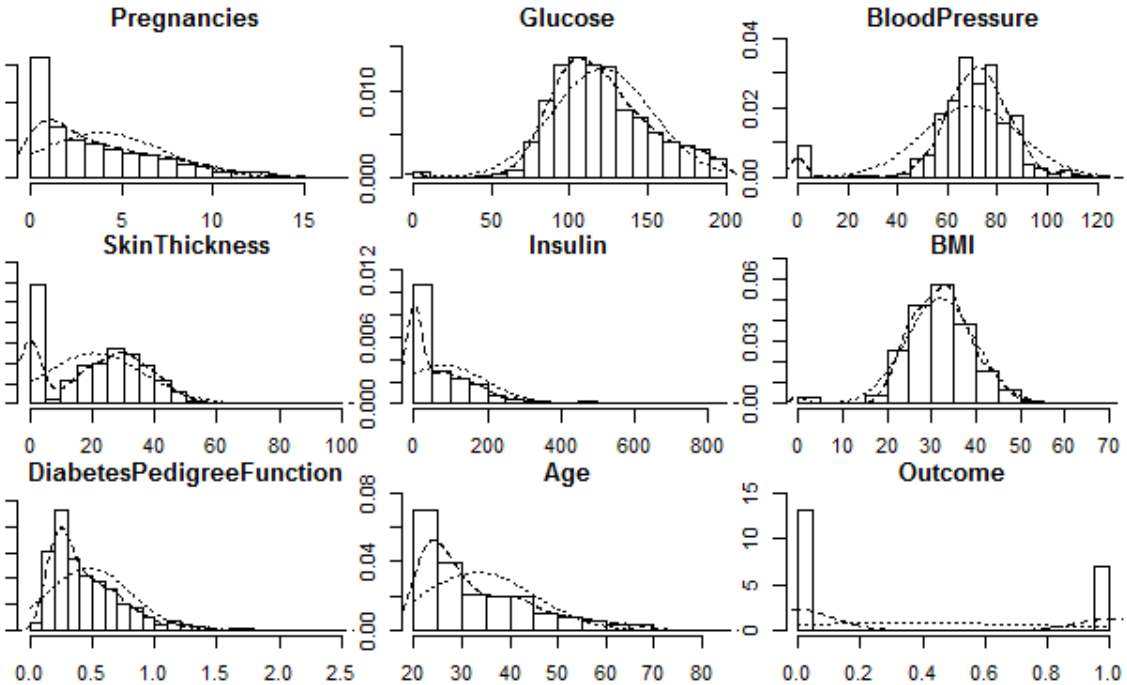
```
                          Pregnancies    Glucose BloodPressure SkinThickness
Pregnancies                1.00000000 0.12945867    0.14128198   -0.08167177
Glucose                    0.12945867 1.00000000    0.15258959    0.05732789
BloodPressure              0.14128198 0.15258959    1.00000000    0.20737054
SkinThickness             -0.08167177 0.05732789    0.20737054    1.00000000
Insulin                   -0.07353461 0.33135711    0.08893338    0.43678257
BMI                        0.01768309 0.22107107    0.28180529    0.39257320
DiabetesPedigreeFunction -0.03352267 0.13733730    0.04126495    0.18392757
Age                        0.54434123 0.26351432    0.23952795   -0.11397026
Outcome                    0.22189815 0.46658140    0.06506836    0.07475223
                             Insulin        BMI DiabetesPedigreeFunction         Age
Pregnancies               -0.07353461 0.01768309              -0.03352267  0.54434123
Glucose                    0.33135711 0.22107107               0.13733730  0.26351432
BloodPressure              0.08893338 0.28180529               0.04126495  0.23952795
SkinThickness              0.43678257 0.39257320               0.18392757 -0.11397026
Insulin                    1.00000000 0.19785906               0.18507093 -0.04216295
BMI                        0.19785906 1.00000000               0.14064695  0.03624187
DiabetesPedigreeFunction   0.18507093 0.14064695               1.00000000  0.03356131
Age                       -0.04216295 0.03624187               0.03356131  1.00000000
Outcome                    0.13054795 0.29269466               0.17384407  0.23835598
                            Outcome
Pregnancies              0.22189815
Glucose                  0.46658140
BloodPressure            0.06506836
SkinThickness            0.07475223
Insulin                  0.13054795
BMI                      0.29269466
DiabetesPedigreeFunction 0.17384407
Age                      0.23835598
Outcome                  1.00000000
```

Pima Indians Diabetes Prediction

```
pairs(~Pregnancies+Glucose+BloodPressure+SkinThickness+Insulin+BMI+DiabetesPedigr
eeFunction+Age, data=diabetes.df)
```




```
install.packages("psych")
library(psych)
multi.hist(diabetes.df)
```



From the above results, we can see that Glucose, BloodPressure, SkinThickness, Insulin and BMI have minimum values of zeroes. These are abnormal records and will influence our analysis so that we will fix them later. There are negative relationships between Pregnancies and SkinThickness, Pregnancies and Insulin, Pregnancies and DiabetesPedigreeFunction, SkinThickness and Age, Insulin and Age. Also, the distribution of BMI seems to be symmetric; distribution of SkinThickness, Age and Insulin seem to be skewed.

Classification – Decision Tree and Logistic Regression

The dataset has eight independent variables and one target variable. We use this dataset to build classification models to predict whether a given female patient, with certain diagnostic measurements, has diabetes or not.

1. Decision Tree

Our first classification model for prediction is the decision tree model which is one of the simplest yet powerful model. The decision tree model leads to classification of the target variable based on 'rules' formed using independent variables.

Implementing the decision tree model for our scenario, we firstly noted that the data set which we have contains an unusual number of zeroes. So, we replaced the zero values with 'NA' because decision tree model can handle missing values in RStudio when the independent variables have missing a value. As it is already mentioned above, we have replaced the unusual zero values in some of the predictors with 'NA' (null), so that we can have a more accurate model.

```
diabetes.df[,"SkinThickness"][diabetes.df[,"SkinThickness"] == 0] <- NA
diabetes.df[,"BMI"][diabetes.df[,"BMI"] == 0] <- NA
diabetes.df[,"BloodPressure"][diabetes.df[,"BloodPressure"] == 0] <- NA
diabetes.df[,"Glucose"][diabetes.df[,"Glucose"] == 0] <- NA
diabetes.df[,"Insulin"][diabetes.df[,"Insulin"] == 0] <- NA
```

Secondly, we set up the training dataset and validation dataset. 60% of records for training data sets and 40% of records for validation data sets. Validation dataset will help us prevent overfitting problems.

```
set.seed(1)
train.index <- sample(c(1:dim(diabetes.df)[1]),dim(diabetes.df)[1]*0.6)
train.df <- diabetes.df[train.index, ]
valid.df <- diabetes.df[-train.index, ]
```

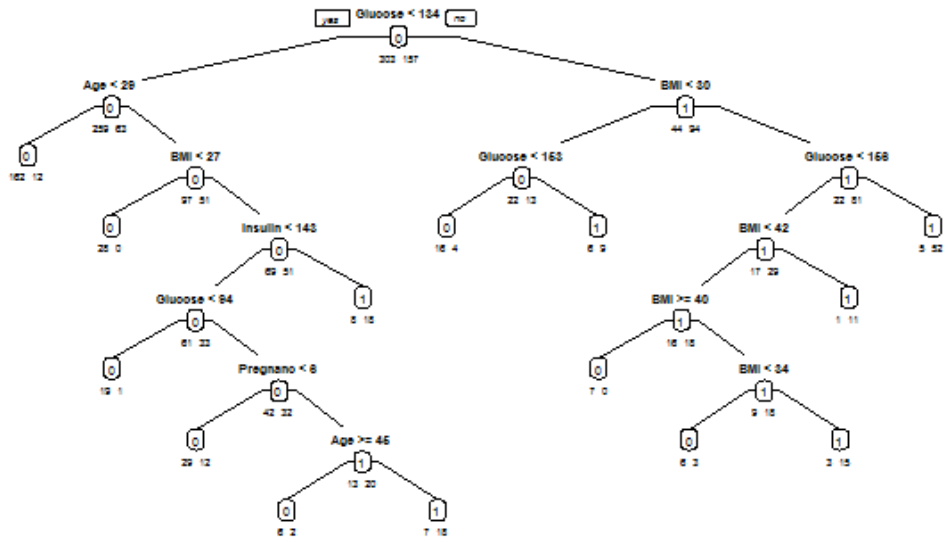
while implementing decision tree, we felt that the number of splits in the tree had to be reduced to a level to avoid overfitting. The tree originally has 8 levels of splitting. The

'maxdepth' option, which specifies the 'level of splits', was chosen after a trial-and-error run of various maxdepth options with a focus for training accuracy compared to validation accuracy. The table of computed values is shown below for reference:

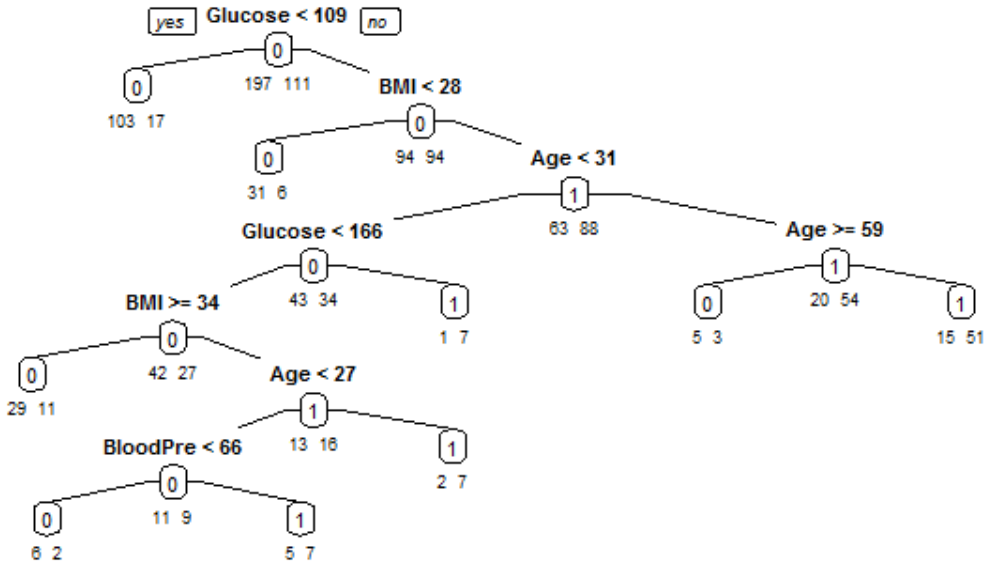
Maxdepth	Training Accuracy	Validation Accuracy
2	0.787	0.7208
3	0.7935	0.75
4	0.8152	0.776
5	0.8304	0.7857
6	0.8522	0.7922
7	0.8609	0.7987
8	0.8717	0.7987

Past the 'maxdepth' value of 7, the validation accuracy is saturated, and the model performs the same. So, we have chosen a value of 7 as the 'maxdepth' or the 'level of splits' for the decision tree.

```
library(rpart)
library(rpart.plot)
train.ct <- rpart(Outcome ~ ., data = train.df, method = "class", maxdepth = 7)
prp(train.ct, type = 1, extra = 1, under = TRUE)
```



```
valid.ct <- rpart(Outcome ~ ., data = valid.df, method = "class", maxdepth = 7)
prp(valid.ct, type = 1, extra = 1, under = TRUE)
```



For decision trees of training data sets and validation data sets, Glucose is the first split. Decision trees start at the root node if the Glucose meets the condition, and traverses down until a leaf node is reached, diabetes or not.

The Decision tree model is implemented, and the results are captured using the 'Accuracy' parameter in Confusion Matrix and 'ROC index' (AUC – Area Under Curve) of the ROC (Receiver Operating Characteristic) curve. A high accuracy rate and large ROC area mean more accurate the model is.

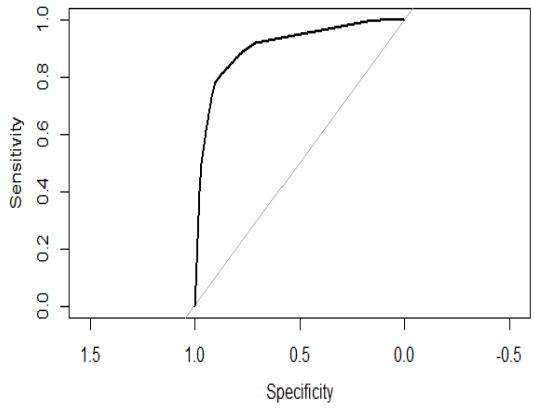
```
library(caret)
train.ct.pred <- predict(train.ct, train.df, type = "class")
confusionMatrix(train.ct.pred, as.factor(train.df$Outcome))

valid.ct.pred <- predict(valid.ct, valid.df, type = "class")
confusionMatrix(valid.ct.pred, as.factor(valid.df$Outcome))
```

Pima Indians Diabetes Prediction

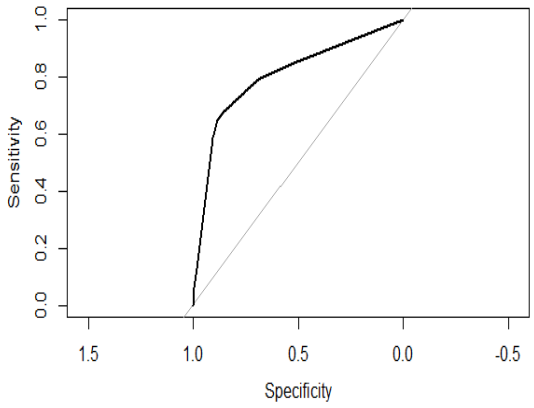
```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 273  34
         1  30 123

               Accuracy : 0.8609
                 95% CI : (0.8258, 0.8912)
    No Information Rate : 0.6587
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.6887
 Mcnemar's Test P-Value : 0.7077

            Sensitivity : 0.9010
            Specificity : 0.7834
         Pos Pred Value : 0.8893
         Neg Pred Value : 0.8039
             Prevalence : 0.6587
         Detection Rate : 0.5935
   Detection Prevalence : 0.6674
      Balanced Accuracy : 0.8422

       'Positive' Class : 0
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 174  39
         1  23  72

               Accuracy : 0.7987
                 95% CI : (0.7495, 0.842)
    No Information Rate : 0.6396
    P-Value [Acc > NIR] : 8.997e-10

                  Kappa : 0.5492
 Mcnemar's Test P-Value : 0.05678

            Sensitivity : 0.8832
            Specificity : 0.6486
         Pos Pred Value : 0.8169
         Neg Pred Value : 0.7579
             Prevalence : 0.6396
         Detection Rate : 0.5649
   Detection Prevalence : 0.6916
      Balanced Accuracy : 0.7659

       'Positive' Class : 0
```

```
library(pROC)
train.ct.pred.roc <- predict(train.ct, train.df, type = "prob")
predict(train.ct, train.df, type = "prob")
rt.train <- roc(train.df$Outcome, train.ct.pred.roc[,2])
plot.roc(rt.train)
auc(rt.train)

valid.ct.pred.roc <- predict(valid.ct, valid.df, type = "prob")
rt.valid <- roc(valid.df$Outcome, valid.ct.pred.roc[,2])
plot.roc(rt.valid)
auc(rt.valid)
```



Area under the curve: 0.9042



Area under the curve: 0.8031

From the above results, accuracy rate of training data 0.8609 is higher than accuracy rate of validation data 0.7987. ROC of training data has a larger area than ROC of validation data. These observations meet our expectations. The outputs are consolidated and recorded for further comparisons with the 'Logistic Regression' model.

2. Logistic Regression

Logistic regression extends the ideas of linear regression to the situation where the outcome variable, Y, is categorical. In our dataset we use logistic regression for classification. We deal only with a binary outcome variable having two possible classes, diabetes or not.

Firstly, the data is read and imputed if there are some unusual or missing values and then processed. As we replace all abnormal zeroes with NAs before, here, we do the data imputation and replace all NAs with their mean values.

```
diabetes.df$SkinThickness[is.na(diabetes.df$SkinThickness)] <-
mean(diabetes.df$SkinThickness,na.rm = TRUE)
diabetes.df$BMI[is.na(diabetes.df$BMI)] <- mean(diabetes.df$BMI,na.rm = TRUE)
diabetes.df$BloodPressure[is.na(diabetes.df$BloodPressure)] <-
mean(diabetes.df$BloodPressure,na.rm = TRUE)
diabetes.df$Glucose[is.na(diabetes.df$Glucose)] <- mean(diabetes.df$Glucose,na.rm =
TRUE)
diabetes.df$Insulin[is.na(diabetes.df$Insulin)] <- mean(diabetes.df$Insulin,na.rm = TRUE)
```

We randomly selected 60% of the records as training data sets, and the remaining records as validation data sets to verify this model. This is followed by what we did in building decision tree model.

```
logit.train.df <- diabetes.df[train.index, ]
logit.valid.df <- diabetes.df[-train.index, ]
```

And then we perform the logistic regression on training and validation data and observe the summaries of logistic regression model for training data and validation data. Hence, we can have the coefficients of variables and know the significance of each variable, for example, variable Glucose is significant here. When we have one person's record, we can output whether he/she has diabetes or not with this model.

```
logit.train.reg <- glm(Outcome ~.,data = logit.train.df, family = "binomial")
summary(logit.train.reg)
```

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6769  -0.6558  -0.3264   0.5577   2.1733

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)              -9.851736   1.137068  -8.664  < 2e-16 ***
Pregnancies               0.160604   0.045287   3.546 0.000391 ***
Glucose                   0.041281   0.005408   7.634 2.28e-14 ***
BloodPressure            -0.020099   0.012662  -1.587 0.112428
SkinThickness            -0.003838   0.018442  -0.208 0.835122
Insulin                   0.001642   0.001911   0.859 0.390302
BMI                       0.109673   0.024445   4.486 7.24e-06 ***
DiabetesPedigreeFunction  0.971549   0.406607   2.389 0.016876 *
Age                       0.016446   0.013835   1.189 0.234552
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 590.55  on 459  degrees of freedom
Residual deviance: 383.41  on 451  degrees of freedom
AIC: 401.41

Number of Fisher Scoring iterations: 5
```

```
logit.valid.reg <- glm(Outcome ~.,data = logit.valid.df, family = "binomial")
summary(logit.valid.reg)
```

```
Call:
glm(formula = Outcome ~ ., family = "binomial", data = logit.valid.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0407  -0.8218  -0.4861   0.8482   2.2607

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)              -8.526536   1.238314  -6.886 5.75e-12 ***
Pregnancies               0.076993   0.049407   1.558  0.11915
Glucose                   0.033835   0.005848   5.786 7.22e-09 ***
BloodPressure             0.006171   0.012175   0.507  0.61221
SkinThickness             0.013653   0.019779   0.690  0.49002
Insulin                  -0.003328   0.001722  -1.933  0.05320 .
BMI                       0.077151   0.028370   2.719  0.00654 **
DiabetesPedigreeFunction  0.875442   0.444145   1.971  0.04872 *
Age                       0.004636   0.013800   0.336  0.73693
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 402.64  on 307  degrees of freedom
Residual deviance: 317.27  on 299  degrees of freedom
AIC: 335.27

Number of Fisher Scoring iterations: 4
```

The odds are defined as the ratio of the probability of success and the probability of failure. Therefore, odds ratio will help us understand the model and realize the influence of each variable on the outcome. For example, odds ratio of pregnancies for validation data is 1.08003, which means one more pregnancy will result in 8.003% increase in the probability of diabetes.

round(data.frame(summary(logit.train.reg)$coefficient,odds.ratio=exp(coef(logit.train.reg))),5)

```
                          Estimate Std..Error  z.value Pr...z.. odds.ratio
(Intercept)               -9.85174    1.13707 -8.66415  0.00000    0.00005
Pregnancies                0.16060    0.04529  3.54638  0.00039    1.17422
Glucose                    0.04128    0.00541  7.63354  0.00000    1.04215
BloodPressure             -0.02010    0.01266 -1.58737  0.11243    0.98010
SkinThickness             -0.00384    0.01844 -0.20814  0.83512    0.99617
Insulin                    0.00164    0.00191  0.85907  0.39030    1.00164
BMI                        0.10967    0.02445  4.48648  0.00001    1.11591
DiabetesPedigreeFunction   0.97155    0.40661  2.38941  0.01688    2.64203
Age                        0.01645    0.01384  1.18872  0.23455    1.01658
```

round(data.frame(summary(logit.valid.reg)$coefficient,odds.ratio=exp(coef(logit.valid.reg))),5)

Pima Indians Diabetes Prediction

	Estimate	Std..Error	z.value	Pr...z..	odds.ratio
(Intercept)	-8.52654	1.23831	-6.88560	0.00000	0.00020
Pregnancies	0.07699	0.04941	1.55835	0.11915	1.08003
Glucose	0.03383	0.00585	5.78567	0.00000	1.03441
BloodPressure	0.00617	0.01217	0.50692	0.61221	1.00619
SkinThickness	0.01365	0.01978	0.69027	0.49002	1.01375
Insulin	-0.00333	0.00172	-1.93327	0.05320	0.99668
BMI	0.07715	0.02837	2.71947	0.00654	1.08021
DiabetesPedigreeFunction	0.87544	0.44415	1.97107	0.04872	2.39994
Age	0.00464	0.01380	0.33592	0.73693	1.00465

Confusion matrix is a measurement that used to represent the performance of a classification model by recording the sources of errors: false positives and false negatives. We use confusion matrix to depict the accuracy of the training data. Accuracy rate for training data set is 80.43% whereas accuracy rate for validation data set is 73.05%. Validation data set has lower accuracy as compared to training data. This meets our expectation.

```
logit.reg.pred.train <- predict(logit.train.reg, logit.train.df)
confusionMatrix(as.factor(ifelse(logit.reg.pred.train > 0.5, 1, 0)),
as.factor(logit.train.df$Outcome))
```

```
logit.reg.pred.valid <- predict(logit.valid.reg, logit.valid.df)
confusionMatrix(as.factor(ifelse(logit.reg.pred.valid > 0.5, 1, 0)),
as.factor(logit.valid.df$Outcome))
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 284  71
         1  19  86

               Accuracy : 0.8043
                 95% CI : (0.7651, 0.8396)
    No Information Rate : 0.6587
    P-Value [Acc > NIR] : 3.879e-12

                  Kappa : 0.5271
 Mcnemar's Test P-Value : 7.621e-08

            Sensitivity : 0.9373
            Specificity : 0.5478
         Pos Pred Value : 0.8000
         Neg Pred Value : 0.8190
             Prevalence : 0.6587
         Detection Rate : 0.6174
   Detection Prevalence : 0.7717
      Balanced Accuracy : 0.7425

       'Positive' Class : 0
```

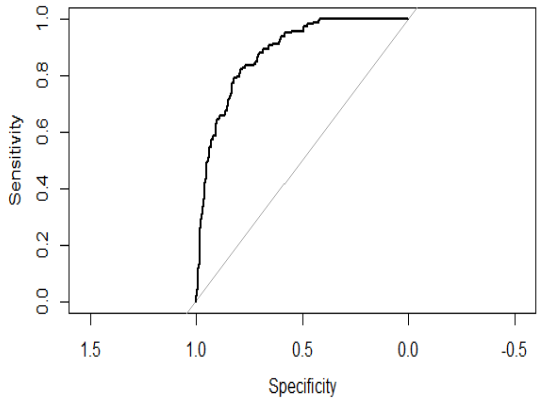
```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 182  68
         1  15  43

               Accuracy : 0.7305
                 95% CI : (0.6773, 0.7793)
    No Information Rate : 0.6396
    P-Value [Acc > NIR] : 0.000435

                  Kappa : 0.3475
 Mcnemar's Test P-Value : 1.145e-08

            Sensitivity : 0.9239
            Specificity : 0.3874
         Pos Pred Value : 0.7280
         Neg Pred Value : 0.7414
             Prevalence : 0.6396
         Detection Rate : 0.5909
   Detection Prevalence : 0.8117
      Balanced Accuracy : 0.6556

       'Positive' Class : 0
```

Confusion matrix for Training data Confusion matrix for validation data

After we acquired accuracy of training data by confusion matrix, we use ROC index to figure out the accuracy of training data and validation data. We can see that training data has a larger ROC area than validation data. This meets our expectation.

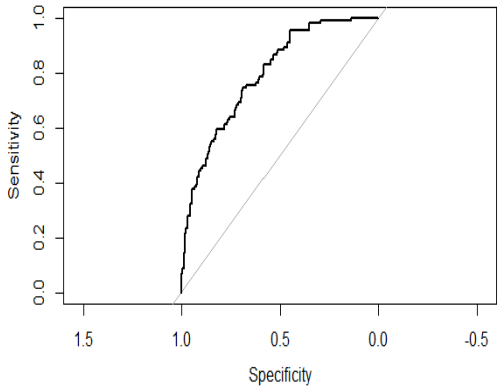
```
rl.train <- roc(train.df$Outcome,logit.reg.pred.train)
plot.roc(rl.train)
auc(rl.train)
```

```
rl.valid <- roc(valid.df$Outcome,logit.reg.pred.valid)
plot.roc(rl.valid)
auc(rl.valid)
```



ROC Curve for Training Data

```
Area under the curve: 0.8799
```



ROC Curve Validation Data

```
Area under the curve: 0.7952
```

Conclusion

We have built the logistic regression model and decision tree model and we have captured the results and performance metrics of the same. For the confusion metrics, decision tree model for validation data sets has the accuracy rate of 0.7987 and logistic regression model for validation data sets has the accuracy rate of 0.7305. Observing the ROC index of the validation data sets of both the models, it is found that the Area under the curve of the decision tree model is higher than that of logistic regression model. Using these parameters as the base, we can say that the decision tree model performs better with the given data in this dataset. However, it's hard for us to conclude as a whole picture because no measure is categorically better than the others, and other measures can be also considered, for example, speed and scalability, robustness, and interpretability. Under this case, logistic regression model attempts to predict outcomes based on a set of independent variables, if we include wrong independent variables, this model will have little to predict the outcome. Therefore, decision tree model is better for our project analysis because decision trees are easy to implement and interpret, and they display higher accuracy than logistic regression model here.

The built decision tree model can be used as the basis for prediction of the incidence of diabetes among Pima Indian women. Using this model, one can employ it to deduce if a particular Pima Indian woman, with a certain medical profile, is likely to have diabetes or not. This model can be used to serve bigger populations and provide predictions that are highly accurate enough with minimum error. The model's predictions can be looked upon as the basis for improvement of measures to study various medical and hereditary factors that could help alleviate diabetes among Pima Indian women. Similar models can be built to study other diabetes prone populations and help in serving the society better with analytics and various classification techniques.

Reference

Clayton Booth, Maziar M. Nourian, Shannon Weaver, Bethany Gull. (2017). *Policy and Social Factors Influencing Diabetes among Pima Indians in Arizona, USA*. Retrieved April 15, 2019 from https://www.researchgate.net/publication/315772355_Policy_and_Social_Factors_Influencing_Diabetes_among_Pima_Indians_in_Arizona_USA

National Institute of Diabetes and Digestive and Kidney Diseases. (2019). *Diabetes.* Retrieved April 10, 2019 from https://www.niddk.nih.gov/about-niddk/research-areas/diabetes

MedlinePlus. (2019). *Diabetes* Retrieved April 11, 2019 from https://medlineplus.gov/diabetes.html

Mayo Clinic. (2018). *Diabetes*. Retrieved April 11, 2019 from https://www.mayoclinic.org/diseases-conditions/diabetes/symptoms-causes/syc-20371444

Office on Women's Health. (2017). *Diabetes.* Retrieved April 15, 2019 from https://www.womenshealth.gov/a-z-topics/diabetes