IME 780AN Final Project

Pima Indians Diabetes Dataset Analysis

Deepansh Arora

May 03, 2021

Table of Contents

Table of Figures

1. Introduction

The Pima Indians Diabetes dataset was chosen from the UCI Repository of Machine Learning Databases. The original owners of the dataset are National Institute of Diabetes and Digestive and Kidney Diseases [1]. Diabetes is a chronic disease which potentially damages the essential organs of the victim including heart, kidney, blood vessels. According to the World Health Organization (WHO) report, about 422 million people have diabetes worldwide [2]. It is perhaps one of the leading causes of the deaths across the globe.

There are 768 records in the dataset. The data was collected from the female patients who were at least 21 years old at the time of data collection. All the patients were of Pima Indian heritage. The dataset chosen have several predictor variables and a target variable. **Table 1** describes the variables in the dataset.

Table 1: Variables description

Number	Variable	Description	Data Type
1	Pregnant	Number of times pregnant	Numeric
2	Glucose	Plasma glucose concentration (glucose tolerance test)	Numeric
3	Pressure	Diastolic blood pressure (mm Hg)	Numeric
4	Triceps	Triceps skin fold thickness (mm)	Numeric
5	Insulin	2-Hour serum insulin (mu U/ml)	Numeric
6	Mass	Body mass index (weight in kg/(height in m)\^2)	Numeric
7	Pedigree	Diabetes pedigree function	Numeric
8	Age	Age (years)	Numeric
9	Diabetes	Class variable (test for diabetes)	Categorical

Classification machine learning algorithms were used to predict if the patient would be diabetic or not. Throughout the analysis, the first 8 variables from **Table 1** were considered to be predictor variables and the last variables was considered as target variable. Unsupervised machine learning technique was also used to group the patients based on certain characteristics.

2. Exploratory Data Analysis

To better understand the relationship between different variables, a scatter matrix was plotted as shown in Figure 1. The upper diagonal represents the Pearson correlation coefficient. The asterisk denotes the level of significance. 3 asterisk means p value less than 0.001 and no asterisk means p value less than 1. We do not see any strong linear correlation between any 2 variables.

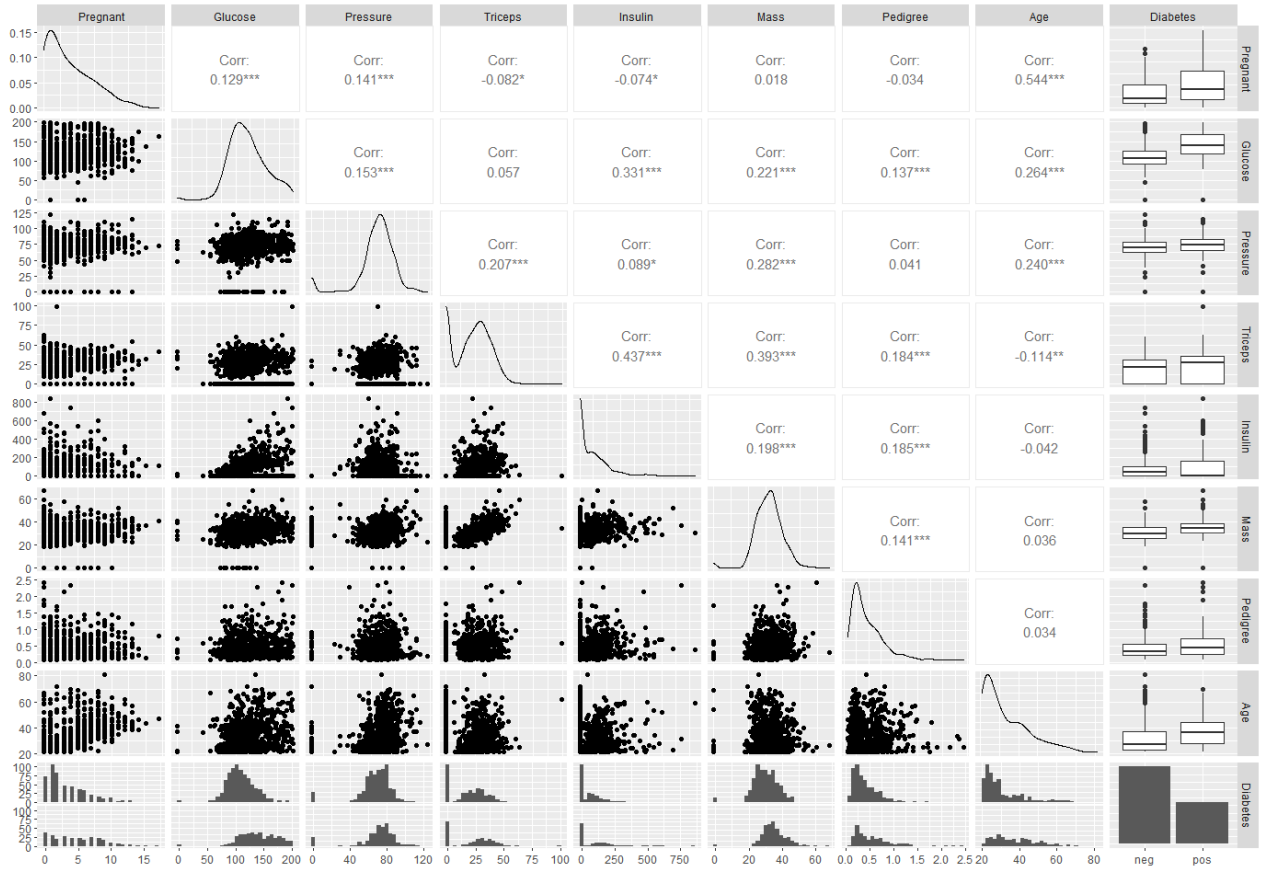

Figure 1: Scatter Matrix Plot

Pregnant	Glucose	Pressure	Triceps	Insulin	Mass	Pedigree	Age
0.8981549	0.1730754	-1.8364126	0.1089456	2.2633826	-0.4273073	1.9124179	1.1251880

Figure 2: Variables' skewness

The figure 2 denotes the skewness of the variables in the dataset. The further the distribution of the skew value from zero, the larger the skew to the left (negative skew value) or right (positive skew value) [3]. Most variables in the dataset are positively skewed. Figure 3 illustrates the class distribution of the dataset. It is clearly evident that the dataset is imbalance in nature.

	freq	percentage
neg	500	65.10417
pos	268	34.89583

Figure 3: Class distribution

3. Machine Learning Models

3.1. Unsupervised Learning

K-means clustering algorithm was used to group the patients according to the certain characteristics. The categorical attribute was removed from the dataset to apply the k-means algorithm. Euclidean distance metric was used. The data was normalized prior to applying K-means. This was done to ensure that all the columns have a common scale. **Figure 4** illustrates the elbow plot for choosing the number of clusters. After careful consideration, the number of clusters chosen were 3 as the total within sum of square error is almost constant after 3 clusters.

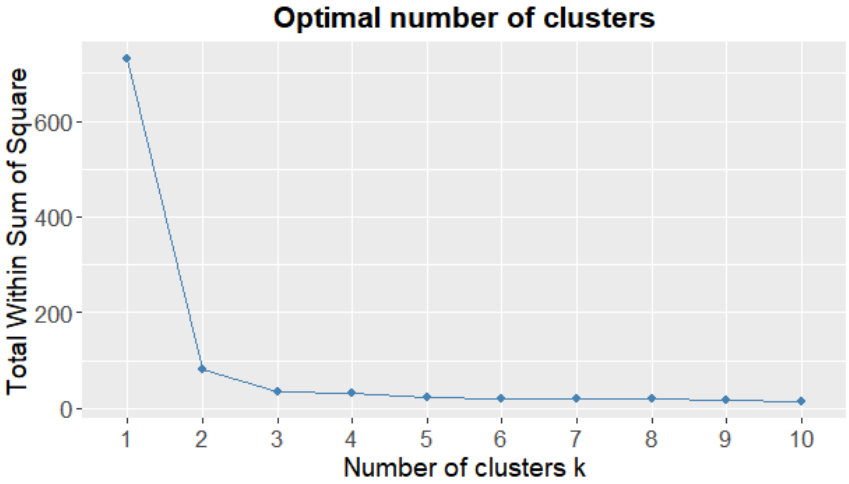

Figure 4: Optimal number of clusters

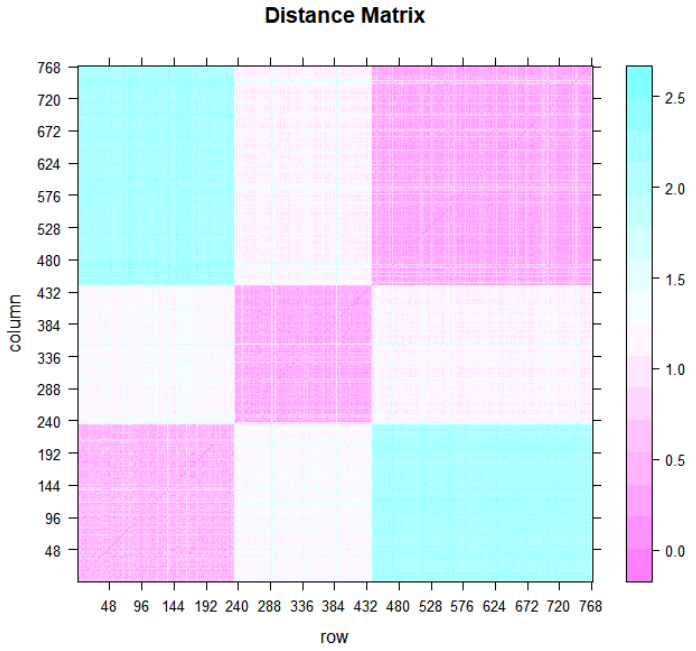

Figure 5: Distance Matrix

Figure 5 illustrates the distance matrix. Distance matrix helps in understanding the quality of the clusters. It can be clearly seen that the clusters do not blend well with the white background. It looks like intra-cluster distance in minimum and inter-cluster distance is maximum. This means that K=3 looks promising.

Figure 6: K-means clustering visualization

8 variables were used to form the K-means model. Principal Component Analysis (PCA) was used to visualize the clusters (**Figure 6**) in 2 dimensions. It can be seen in figure 6 that clusters 2 and 1 overlap each other a little bit. A cluster with good quality is supposed to have less intra-cluster distance and high inter cluster distance. In this case we see that the 3 clusters appear to be very close to each other and two of them slightly overlap each other.

3.2. Supervised Learning

Classification models were developed to predict if the patient would be diabetic or not based on certain characteristics. Firstly, the data was divided randomly into training and testing dataset. The split ratio was chosen to be 75%. This implies that 75% of the dataset was used for training and remaining was used for testing the classification model. In order to compare the 2 classification models, the same predictor variables were used in the 2 models. The predictor variables used were Pregnant, Glucose, Pedigree, and Mass. Based on the statistical analysis, only useful variables were chosen as predictor variables. The target variable is Diabetes, whose output could be either positive or negative. The 2 classification models developed are discussed below:

A. Logistic Regression

```
glm(formula = Diabetes ~ Pregnant + Glucose + Pedigree + Mass,
    family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7574  -0.7076  -0.3675   0.6713   2.4909

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.516192   0.886670 -10.733  < 2e-16 ***
Pregnant     0.188030   0.034722   5.415 6.12e-08 ***
Glucose      0.037514   0.004434   8.461  < 2e-16 ***
Pedigree     1.147900   0.384413   2.986  0.00283 **
Mass         0.088841   0.017316   5.130 2.89e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 663.60  on 512  degrees of freedom
Residual deviance: 466.43  on 508  degrees of freedom
AIC: 476.43

Number of Fisher Scoring iterations: 5
```

Figure 7: Logistic Regression Output

Figure 7 illustrates the summary of Logistic Regression output. It can be seen that all the predictor variables are significant. The equation of the model will be:

$$\log(odds) = -9.516 + Pregnant * 0.188 + Glucose * 0.0375 + Pedigree * 1.147 + Mass * 0.088$$

Figure 8 demonstrates the area under the curve (AUC) for the logistic regression classification model. It can be seen that the AUC is about 0.8. From the ROC Curve, the threshold value chosen was 0.3. It implies that if the probability is greater than 0.3, it will be classified as positive else negative. Since this dataset is related to healthcare, it is better to classify more samples as positive than negative.

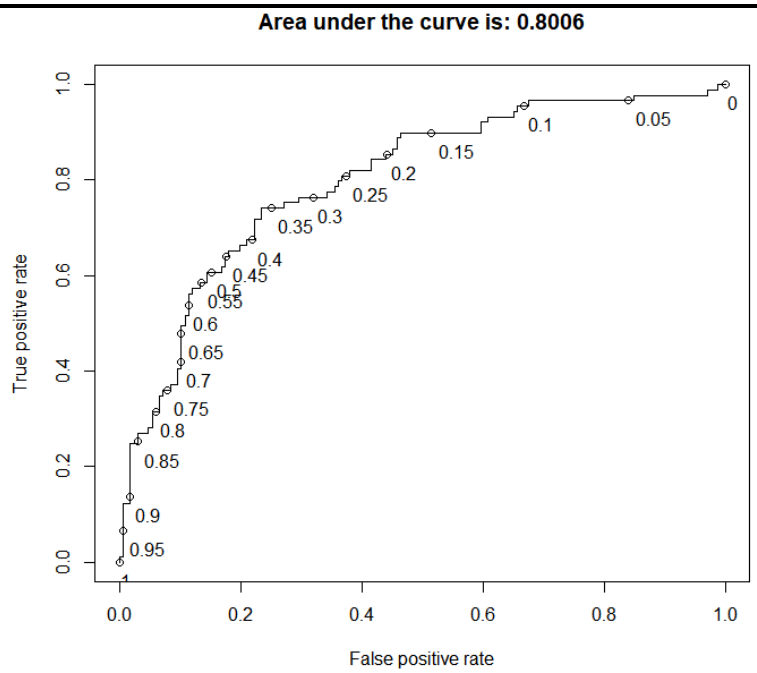

Figure 8: ROC Curve for Logistic Regression

```
Confusion Matrix and Statistics


df_class neg pos
     neg 113  21
     pos  53  68

               Accuracy : 0.7098
                 95% CI : (0.6499, 0.7647)
    No Information Rate : 0.651
    P-Value [Acc > NIR] : 0.0271240

                  Kappa : 0.4105

 Mcnemar's Test P-Value : 0.0003137

            Sensitivity : 0.6807
            Specificity : 0.7640
         Pos Pred Value : 0.8433
         Neg Pred Value : 0.5620
             Prevalence : 0.6510
         Detection Rate : 0.4431
   Detection Prevalence : 0.5255
      Balanced Accuracy : 0.7224

       'Positive' Class : neg
```

Figure 9: Confusion Matrix Output for Logistic Regression

From confusion matrix, it can be seen that the accuracy of the model is 70.98%. The positive class in the confusion matrix is “negative”. The model has a higher specificity value indicating that there is 76.4% probability that model will correctly classify the positive cases.

B. Decision Tree

Figure 10 illustrates the plot for complexity parameter (Cp) vs the relative error. To construct the decision tree, 0.02 was chosen as the value for Cp.

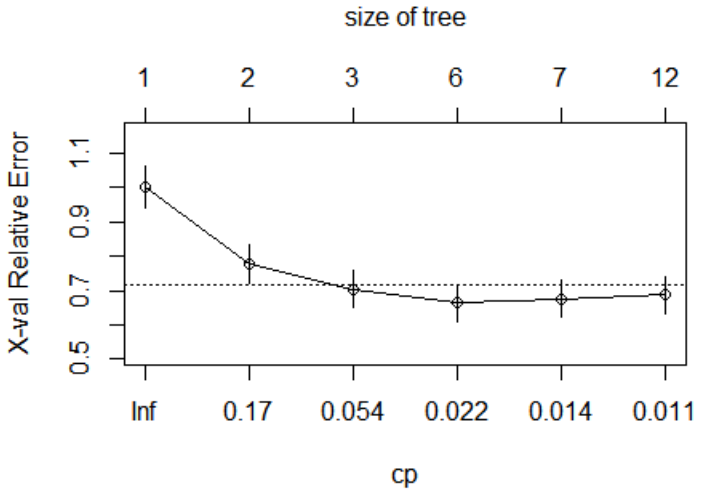

Figure 10: Cp Plot

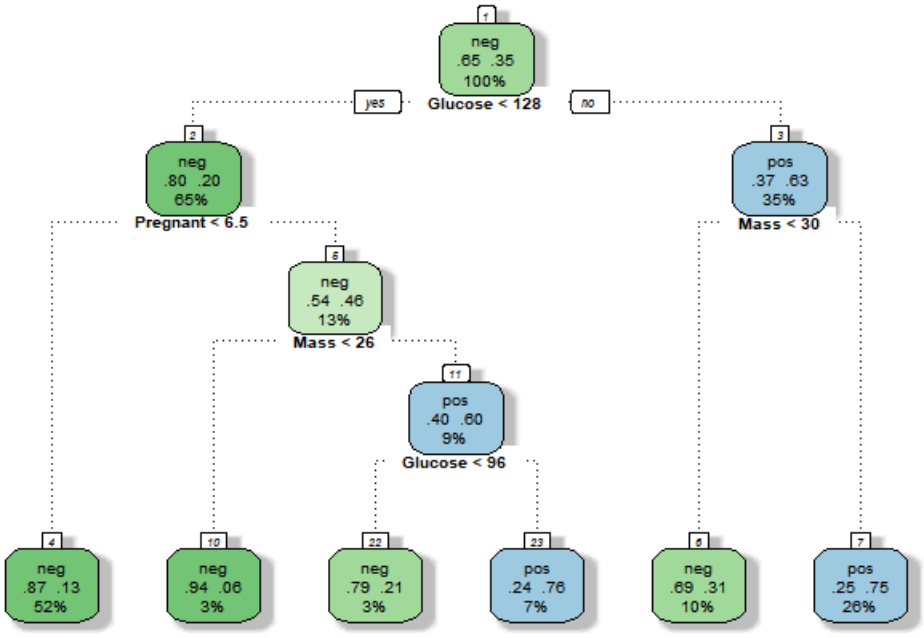

Figure 11: Decision Tree

Figure 11 illustrates the decision tree. In node 1, we can see that dataset contains 65% negative samples and 35% positive samples. This is in agreement with the **Figure 3**. If the answer to the question in node 1 is yes then we will go to the node 2 else node 3. If the answer to the question asked in node 2 is true then we go to node 4 else node 5 and so on. If a female has glucose level less than128 and less than 6.5 (<=6) children, then there is 87% probability that she will not be diabetic. Similarly, if a female has glucose

level greater than 128 and mass (BMI) greater than 30, than there is 25% chance that she will be diabetic.

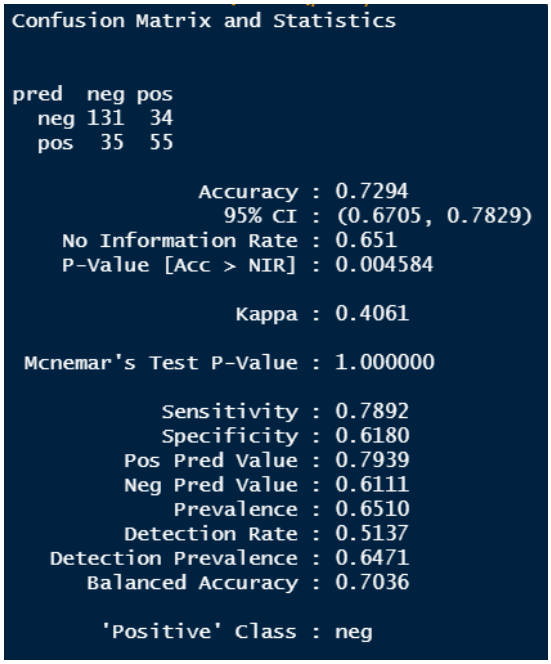

Figure 12: Confusion Matrix Output for decision tree

Figure 12 demonstrates the confusion matrix output for the decision tree model. As we can see that the accuracy is 72.94%. This model has low value of specificity than the previous one. However, this model has a very good sensitivity rate.

4. Conclusion

For this project, Pima Indians Diabetes Dataset was chosen for analysis. Statistical analysis was done to get an insight of the data. It was found that the dataset has an imbalance class distribution and most of the variables are positively skewed. Machine learning classification models were developed to predict if a patient is diabetic or not provided we know information about their Pregnancy, Glucose level, Pedigree, and Mass (BMI). Logistic regression and decision tree classification models were formulated. **Table 2** compares the two models developed above. Since the class distribution is imbalance, it may be worth considering kappa values as well for both models.

Although, decision trees seems to be having slightly higher accuracy than the logistic regression model, but latter seems to be having higher values for kappa and specificity. Kappa indicates the level of agreement between the true values and classification [4]. Since this dataset belongs to the medical field and is directly related to the well being of individuals, it is preferred that we predict the number of positive cases accurately than the number of negative cases. Therefore, Logistic regression maybe a better model for this dataset. The model can definitely be improved by tuning the parameters, collecting more data and/or making the class distribution balanced.

Table 2: Models Summary

Parameters	Logistic Regression	Decision Tree
Accuracy	70.98%	72.94%
Kappa	0.4105	0.4061
Specificity	68.07%	78.92%
Sensitivity	76.40%	61.80%

5. References

[1] U.S. Department of Health and Human Services. (2020, January 3). National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK). National Institutes of Health. https://www.nih.gov/about-nih/what-we-do/nih-almanac/national-institute-diabetes-digestive-kidney-diseases-niddk.

[2] Definition, Diagnosis and Classification of Diabetes Mellitus and its Complications. Part 1: Diagnosis and Classification of Diabetes Mellitus (WHO/NCD/NCS/99.2). Geneva: World Health Organization; 2019.

[3] J. Brownlee, Master Machine Learning Algorithms: Discover How They Work and Implement Them From Scratch, 2017.

[4] Lantz, Brett. Machine Learning with R: Discover How to Build Machine Learning Algorithms, Prepare Data, and Dig Deep into Data Prediction Techniques with R. Birmingham: Packt Publishing, 2015.

6. Appendix: R Code

```
## Loading required libraries
library(mlbench)
library(ggplot2)
library(dplyr)
library(GGally)
library(knitr)
library(e1071)
library(caTools)
library(ROCR)
library(caret)
library(lattice)
library(rpart)
library(rpart.plot)
library(tree)
library(heatmaply)
library(factoextra)
library(rattle)


## Loading the dataset
data(PimaIndiansDiabetes)
df = PimaIndiansDiabetes

## Changing the column names
colnames(df) = c("Pregnant","Glucose", "Pressure", "Triceps","Insulin",
"Mass","Pedigree","Age","Diabetes")

## Scatter matrix plot
#ggpairs(df)

## Class distribution and skewness
y <- df$Diabetes
kable(cbind(freq=table(y), percentage=prop.table(table(y))*100))
df_skew <- apply(df[,1:8], 2, skewness)
sum(is.na(df))
## K means
df_kmeans = df
df_normalize<-normalize(df[-c(9)]) ## Normalizing the data
df_clusters<-kmeans(df_normalize,centers=3,iter.max = 15, nstart = 10)
df_normalize$Cluster_no = df_clusters$cluster
df_sort = df_normalize %>% arrange(Cluster_no)

fviz_nbclust(df_normalize, kmeans, method = "wss") +
theme_gray()+theme(axis.title = element_text(size = 14),
    axis.text = element_text(size = 13),
    plot.title = element_text(size = 16,
        face = "bold", hjust = 0.5))

DistMatrixSort =  as.matrix(dist(df_sort, method="euclidean"))
x.scale <- list(at=seq(0,768,48))
y.scale <- list(at=seq(0,768,48))
levelplot(DistMatrixSort, scales=list(x=x.scale, y=y.scale), main="Distance
Matrix")
```

```
fviz_cluster(df_clusters, data = df_normalize,
             palette = c("#2E9FDF", "#00AFBB", "#E7B800"),
             geom = "point",
             ellipse.type = "convex",
             ggtheme = theme_bw()
) + theme(axis.title = element_text(size = 13,
    face = "bold"), axis.text = element_text(size = 15),
    plot.title = element_text(size = 16,
        face = "bold", hjust = 0.5), panel.background = element_rect(fill =
NA))

#Splitting the data in train & test
set.seed(123)
split = sample.split(df, SplitRatio = 0.75)
train = subset(df, split=="TRUE")
test = subset(df, split=="FALSE")

#Creating the logistic regression model
logregmodel<-glm(Diabetes ~Pregnant+Glucose+Pedigree+Mass , data = train,
family="binomial")
summary(logregmodel)
df_test<-predict(logregmodel, newdata = test, type = "response")
range(df_test)

## Plotting the regression model
ROCpred<- prediction(df_test, test$Diabetes) #putting the test data into the
classification model
ROCRperf<-performance(ROCpred, "tpr", "fpr") #using tpr and fpr
plot(ROCRperf) #Plotting the ROC Curve
AUC<-as.numeric(performance(ROCpred, 'auc')@y.values) #Calculating the area
under curve
plot(ROCRperf, print.cutoffs.at = seq(0,1,0.05), text.adj = c(-0.2, 1.7),
    main = paste("Area under the curve is:", round(AUC,5))) #plotting the
points on the curve

## Setting the threshold & predicting the accuracy of the model
pos_neg = ifelse(df_test>0.3, "pos", "neg")
df_class <- factor(pos_neg, levels = levels(test$Diabetes))
confusionMatrix(table(df_class, test$Diabetes))


## Decision Tree
set.seed(123)
k = trainControl(method="cv", number=10)
cpvalues = expand.grid(.cp = seq(0.01, 0.2, 0.01)) ## Performing 10 fold
cross validation
train(Diabetes ~Pregnant+Glucose+Pedigree+Mass, data =train,
      method = "rpart", trControl = k, tuneGrid = cpvalues)
dec_tree_df<-rpart(Diabetes ~Pregnant+Glucose+Pedigree+Mass+Pedigree,
                   data = train, method = "class", cp=0.02)

pred<-predict(dec_tree_df, newdata = test, type="class", na.action = na.pass)
head(pred)
confusionMatrix(table(pred, test$Diabetes)) ## Analyzing the model
```

```
rpart.plot(dec_tree_df)
plotcp(dec_tree_df)
rattle()
## Plotting the decision tree
fancyRpartPlot(dec_tree_df, cex = 0.63)
```