Predicting patients diabetes

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

The datasets consists of several medical predictor variables and one target variable, Outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

Can you build a machine learning model to accurately predict whether or not the patients in the dataset have diabetes or not?

```r
getwd()

## [1] "C:/Users/badal/Desktop/R use cases"

#install.packages("ggplot2")
#install.packages("corrplot")
#install.packages("ROCR")

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.6.3

library(corrplot)

## Warning: package 'corrplot' was built under R version 3.6.1

## corrplot 0.84 loaded

library(ROCR)

## Warning: package 'ROCR' was built under R version 3.6.1

## Loading required package: gplots

## Warning: package 'gplots' was built under R version 3.6.1

## 
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
## 
##     lowess
```

Load dataset

display first 6 rows of data

```
dbt <- read.csv("file:///C:/Users/badal/Desktop/datset_/diabetes.csv")
head(dbt)
```

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

```
any(is.na(dbt))
```

```
## [1] FALSE
```

```
summary(dbt)
```

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

Understand the structure of the dataset

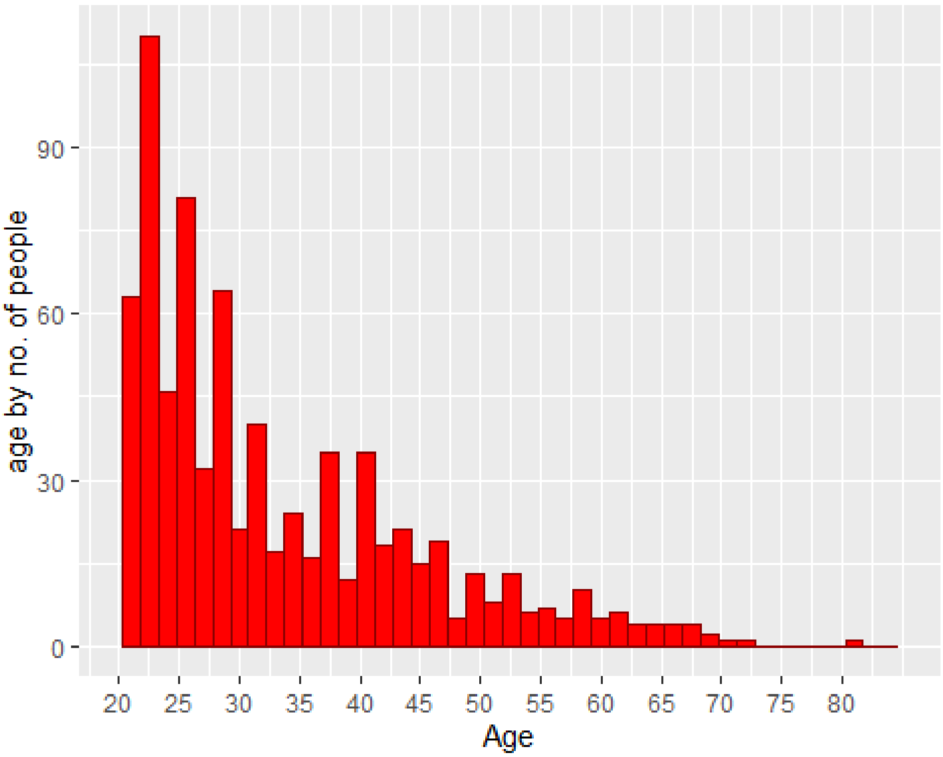
```
str(dbt)

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

Create Age by Category column

```
Age_ <- ifelse(dbt$Age < 21, "<21",
                   ifelse((dbt$Age>=21) & (dbt$Age<=30), "21-30",
                   ifelse((dbt$Age>21) & (dbt$Age<=40), "31-40",
                   ifelse((dbt$Age>41) & (dbt$Age<=50), "41-50",
                   ifelse((dbt$Age>51) & (dbt$Age<=90), "51-60",
                          ">61")))))
table(Age_)

## Age_
##   >61 21-30 31-40 41-50 51-60
##    30   417   157    91    73

Age_1<- factor(Age_, levels = c('<21','21-30','31-40','41-50','51-60','>61'))
table(Age_1)

## Age_1
##   <21 21-30 31-40 41-50 51-60   >61
##     0   417   157    91    73    30
```

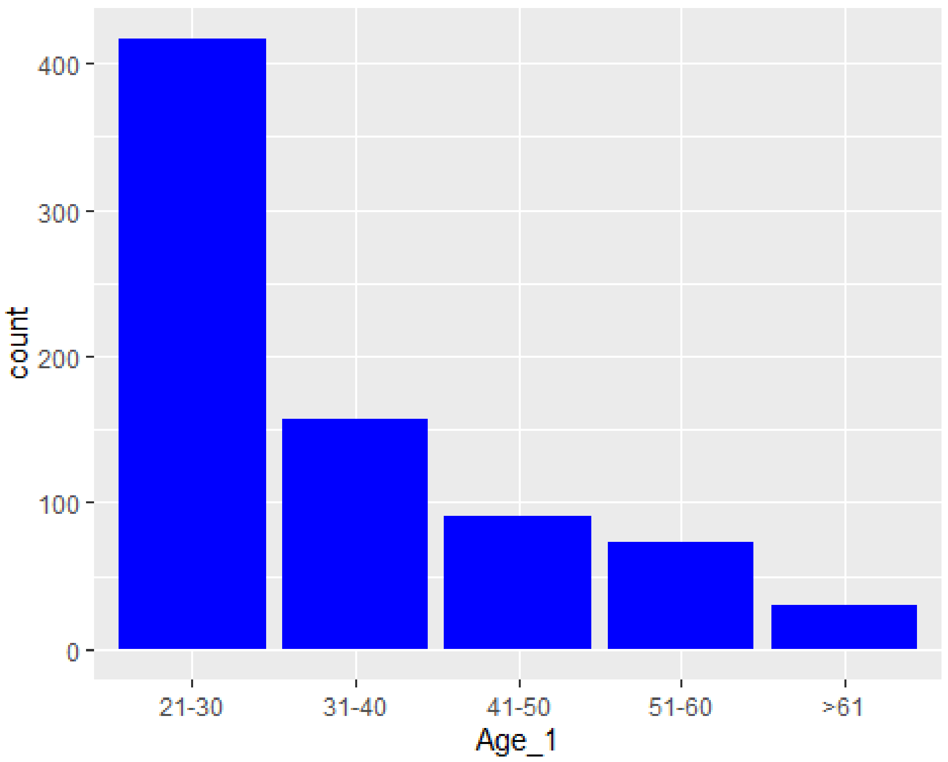
Histogram of Age

```
library(ggplot2)

ggplot(aes(x = Age), data=dbt) +
        geom_histogram(binwidth=1.5, color=' darkred', fill = "Red") +
        scale_x_continuous(limits=c(20,85), breaks=seq(20,80,5)) +
        xlab("Age") +
        ylab("age by no. of people")

## Warning: Removed 2 rows containing missing values (geom_bar).
```



# Barplot by Age by category, Most of the people are in between the ages 21 - 30

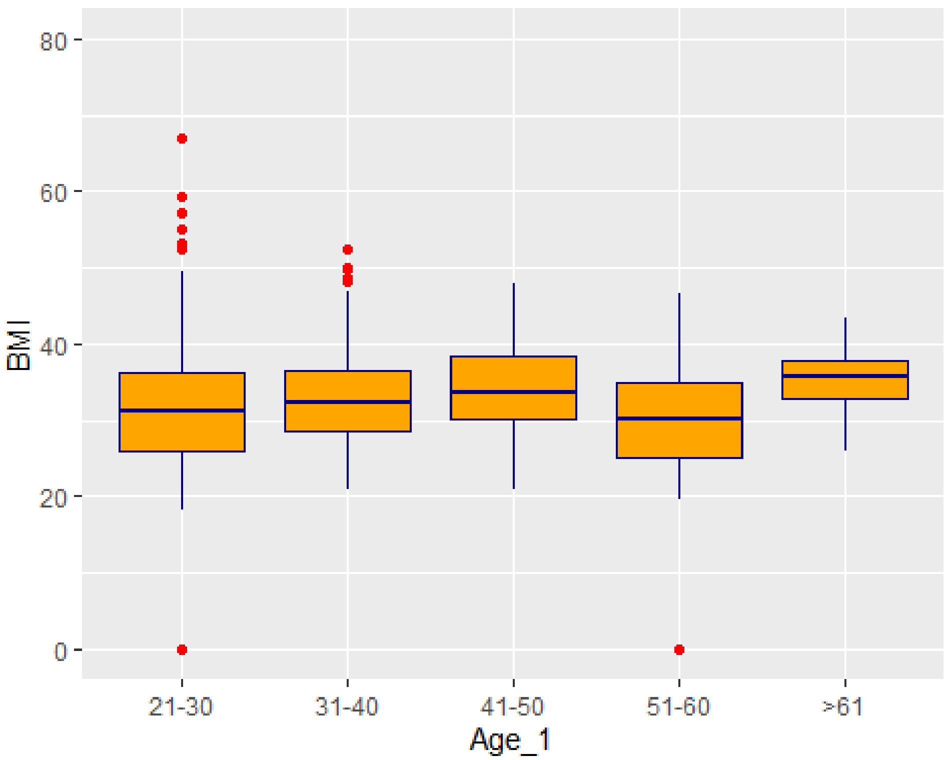
```
library(ggplot2)
ggplot(aes(x = Age_1), data = dbt) +
            geom_bar(fill='blue')
```



# box plot of Age

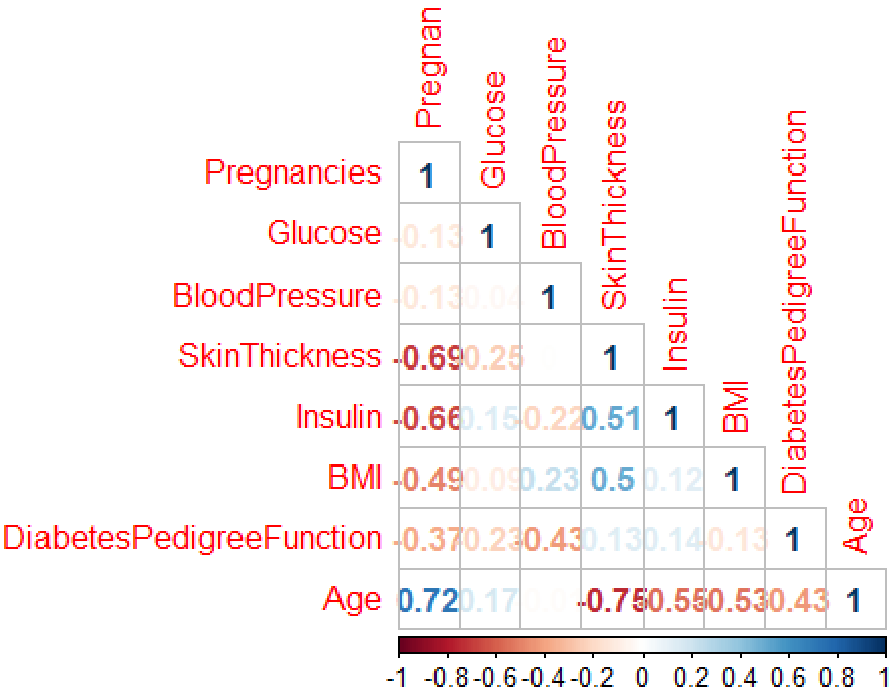
Catetogry vs BMI

```
ggplot(aes(x=Age_1, y = BMI ), data = dbt) +
        geom_boxplot(color='darkblue', fill = "orange", outlier.colour =
"red") +
        coord_cartesian(ylim = c(0,80))
```



#correlation matrix

```
dbt_cor <- round(cor(dbt[1:8]),1)
dbt_cor

##                          Pregnancies Glucose BloodPressure SkinThickness
## Pregnancies                      1.0     0.1           0.1          -0.1
## Glucose                          0.1     1.0           0.2           0.1
## BloodPressure                    0.1     0.2           1.0           0.2
## SkinThickness                   -0.1     0.1           0.2           1.0
## Insulin                         -0.1     0.3           0.1           0.4
## BMI                              0.0     0.2           0.3           0.4
## DiabetesPedigreeFunction         0.0     0.1           0.0           0.2
## Age                              0.5     0.3           0.2          -0.1
##                          Insulin BMI DiabetesPedigreeFunction  Age
## Pregnancies                 -0.1 0.0                      0.0  0.5
## Glucose                      0.3 0.2                      0.1  0.3
## BloodPressure                0.1 0.3                      0.0  0.2
## SkinThickness                0.4 0.4                      0.2 -0.1
## Insulin                      1.0 0.2                      0.2  0.0
## BMI                          0.2 1.0                      0.1  0.0
## DiabetesPedigreeFunction     0.2 0.1                      1.0  0.0
## Age                          0.0 0.0                      0.0  1.0

library(corrplot)
corrplot(cor(dbt_cor), method = "number",
         type = "lower")
```



there are No strong correlation observed between variables.so we can do further analysis withoiut droppiong any columns.

```
require(caTools)

## Loading required package: caTools

## Warning: package 'caTools' was built under R version 3.6.1

set.seed(123)
sample = sample.split(dbt$Outcome, SplitRatio=0.80)
train = subset(dbt, sample==TRUE)
test = subset(dbt, sample==FALSE)

nrow(dbt)

## [1] 768

nrow(train)

## [1] 614

nrow(test)

## [1] 154
```

Fit model - using all independent variables

```
model_1 <- glm(Outcome ~ . , data = train, family = binomial(link= "logit"))

summary(model_1)

##
## Call:
## glm(formula = Outcome ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4548  -0.7104  -0.4188   0.7042   2.9252
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -8.202293   0.786099 -10.434  < 2e-16 ***
## Pregnancies               0.133846   0.036810   3.636 0.000277 ***
## Glucose                   0.036551   0.004209   8.684  < 2e-16 ***
## BloodPressure            -0.016071   0.005827  -2.758 0.005816 **
## SkinThickness             0.007252   0.007892   0.919 0.358146
## Insulin                  -0.002382   0.001101  -2.164 0.030434 *
## BMI                       0.086303   0.016810   5.134 2.84e-07 ***
## DiabetesPedigreeFunction  0.681589   0.340474   2.002 0.045297 *
## Age                       0.014815   0.010522   1.408 0.159132
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 793.94  on 613  degrees of freedom
## Residual deviance: 575.40  on 605  degrees of freedom
## AIC: 593.4
##
## Number of Fisher Scoring iterations: 5
```

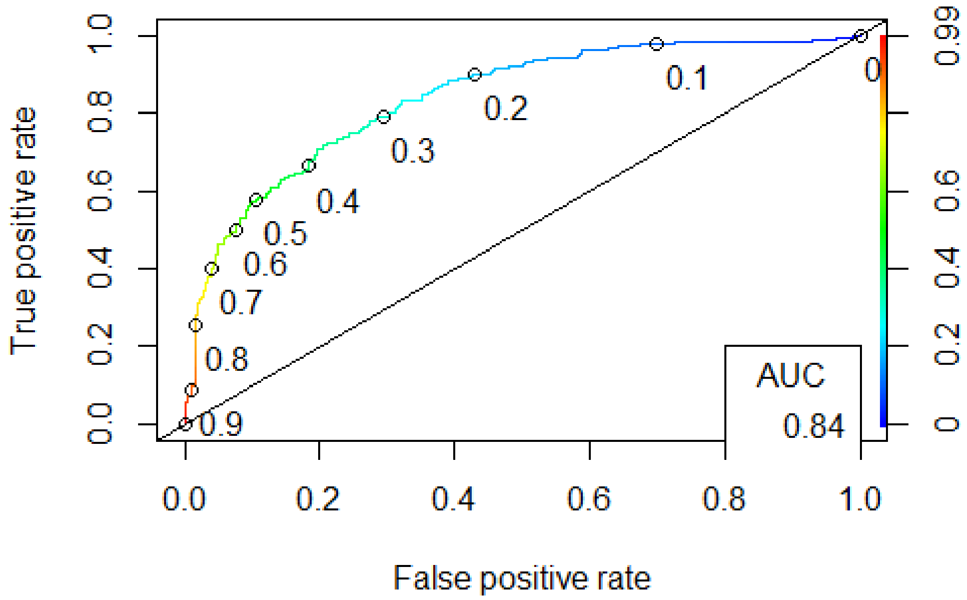
#predict Outcome on Training dataset

```
Predict <- predict(model_1, type = "response")
summary(Predict)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001904 0.117246 0.260366 0.348534 0.537707 0.990924
```

#the average prediction for each of the two outcomes

```
tapply(Predict, train$Outcome, mean)

##         0         1
## 0.2333061 0.5639139
```

```r
# Generate ROC Curves

library(ROCR)

ROC_pred = prediction(Predict, train$Outcome)
ROC_perf = performance(ROC_pred, "tpr", "fpr")

# Adding threshold labels
plot(ROC_perf, colorize=TRUE, print.cutoffs.at = seq(0,1,0.1), text.adj = c(-
0.2, 1.7))
abline(a=0, b=1)

auc_train <- round(as.numeric(performance(ROC_pred, "auc")@y.values),2)
legend(.8, .2, auc_train, title = "AUC", cex=1)
```



```r
# Making predictions on test set

Pred_Test <- predict(model_1, type = "response", newdata = test)

# Convert probabilities to values using the below

## Based on ROC curve above, selected a threshold of 0.5
test_tab <- table(test$Outcome, Pred_Test > 0.5)
test_tab
```

```
##
##     FALSE TRUE
##   0    84   16
##   1    24   30

accuracy_test <- round(sum(diag(test_tab))/sum(test_tab),2)
sprintf("Accuracy on test set is %s", accuracy_test)

## [1] "Accuracy on test set is 0.74"

ROCRPredTest = prediction(Pred_Test, test$Outcome)
auc = round(as.numeric(performance(ROCRPredTest, "auc")@y.values),2)
auc

## [1] 0.82
```

From the above graph it is inferred that we get an accuracy rate of 82% on our Test data. Hence, the model is 82% accurate to predict whether the person is Diabetic or not.