Artificial Neural Network (ANN) on Diabetes data

Remi Oni

February 24, 2022

Contents

1 INTRODUCTION

This project deals with the prediction of the Diabetes status of patient using machine learning procedure.I tend to examine the relationship of some factors that can influence the status of a Diabetic patient.In other to achieve that, the data need to be subjected to various data inspection such as Data cleaning, Exploration and Variable Selection for us to be able to construct an efficient Machine learning model to predict or forecast the Diabetes status of a patient. The goal of this project is to use a machine learning classification model to predict Diabetes status (“0” as Positive and “1” as Negative) based on predictor variables in (columns 1-16) of the Diabetes data set.

2 DATA INPUT

In other to develop a machine learning model for the Diabetes data set . The data must be read into R first and the dimension of the data set must be known, the sample size of the data and the number of variable involved must be examined.

```
dat <- read.csv(
file="http://archive.ics.uci.edu/ml/machine-learning-databases/00529/diabetes_data_uploa
dim(dat)
```

```
## [1] 520  17
```

The dimension shows I have 520 data size (in Rows) and 17 variables (in Columns) i.e 1 target variable and 16 predictors variable. The first 6 rows of the data set was displayed above. The target variable “class” is characterized as having value “0” as “Positive” and value “1” as “negative”.Columns 1-16 contain the predictors variables.

3 DATA PREPARATION AND EXPLORATORY DATA ANALYSIS

Inspecting the variable types, missing values and possibly wrong records, and other issues.

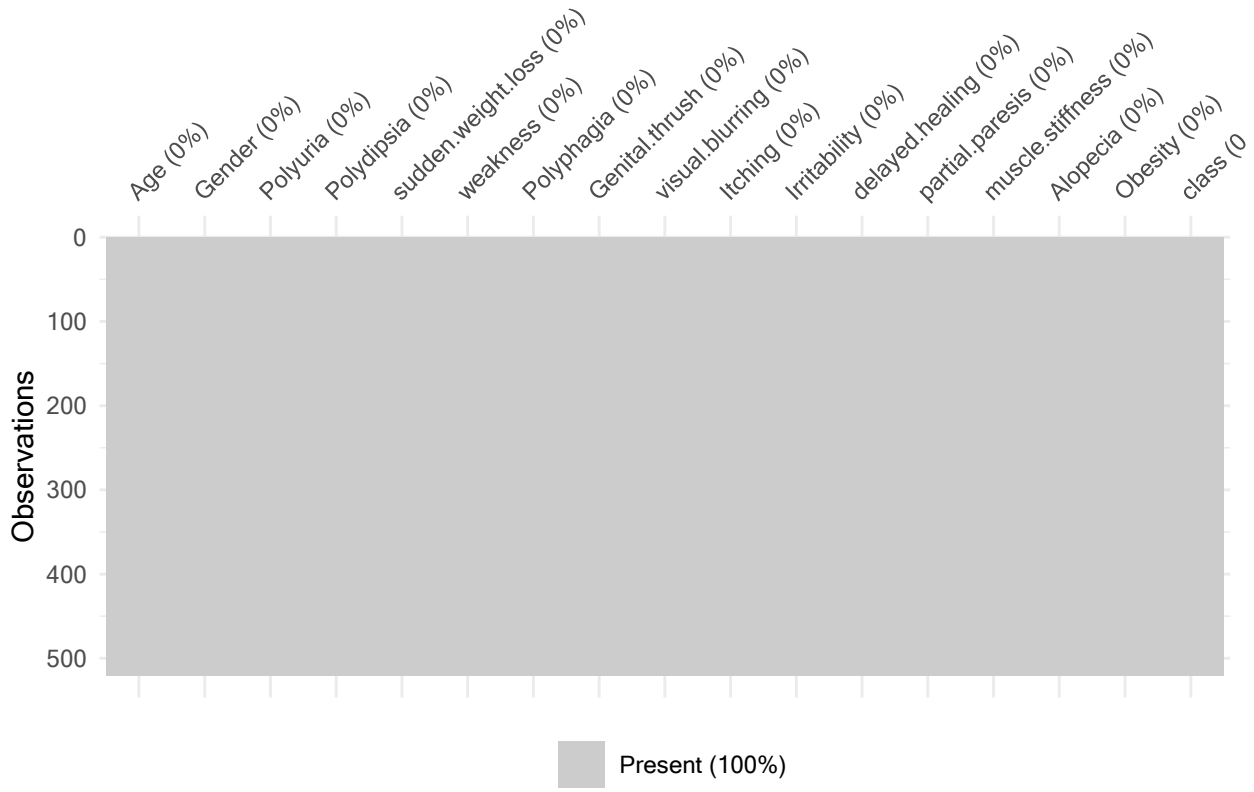
```
dat$Gender <-ifelse(dat$Gender =="Male", 0,1)
dat$Polyuria <-ifelse(dat$Polyuria=="Yes", 0,1)
dat$Polydipsia<-ifelse(dat$Polydipsia=="Yes", 0,1)
dat$sudden.weight.loss <-ifelse(dat$sudden.weight.loss=="Yes", 0,1)
dat$weakness <-ifelse(dat$weakness=="Yes", 0,1)
dat$Polyphagia <-ifelse(dat$Polyphagia=="Yes", 0,1)
dat$Genital.thrush <-ifelse(dat$Genital.thrush=="Yes", 0,1)
dat$visual.blurring <-ifelse(dat$visual.blurring=="Yes", 0,1)
dat$Itching <-ifelse(dat$Itching=="Yes", 0,1)
dat$Irritability<-ifelse(dat$Irritability=="Yes", 0,1)
dat$delayed.healing <-ifelse(dat$delayed.healing=="Yes", 0,1)
dat$partial.paresis <-ifelse(dat$partial.paresis=="Yes", 0,1)
dat$muscle.stiffness <-ifelse(dat$muscle.stiffness=="Yes", 0,1)
dat$Alopecia <-ifelse(dat$Alopecia=="Yes", 0,1)
dat$Obesity <-ifelse(dat$Obesity=="Yes", 0,1)
dat$class <-ifelse(dat$class=="Positive", 0,1)
head(dat)
```

```
##   Age Gender Polyuria Polydipsia sudden.weight.loss weakness Polyphagia
## 1  40      0        1          0                  1        0          1
## 2  58      0        1          1                  1        0          1
## 3  41      0        0          1                  1        0          0
## 4  45      0        1          1                  0        0          0
## 5  60      0        0          0                  0        0          0
## 6  55      0        0          0                  1        0          0
##   Genital.thrush visual.blurring Itching Irritability delayed.healing
## 1              1               1       0            1               0
## 2              1               0       1            1               1
## 3              1               1       0            1               0
## 4              0               1       0            1               0
## 5              1               0       0            0               0
## 6              1               0       0            1               0
##   partial.paresis muscle.stiffness Alopecia Obesity class
## 1               1                0        0       0     0
## 2               0                1        0       1     0
## 3               1                0        0       1     0
## 4               1                1        1       1     0
## 5               0                0        0       0     0
## 6               1                0        0       0     0
```

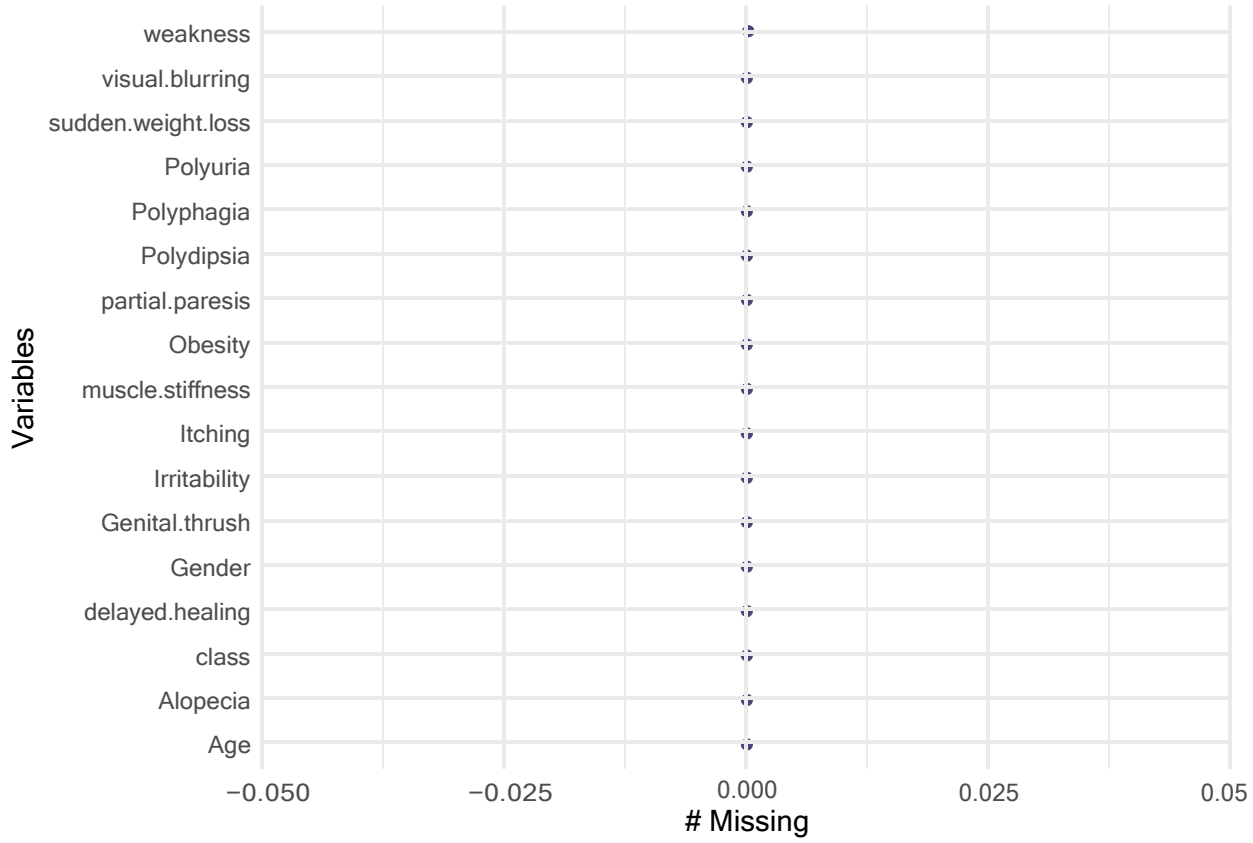
```
anyNA(dat)
```

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

```
library(naniar)
vis_miss(dat)
```




```
gg_miss_var(dat)
```



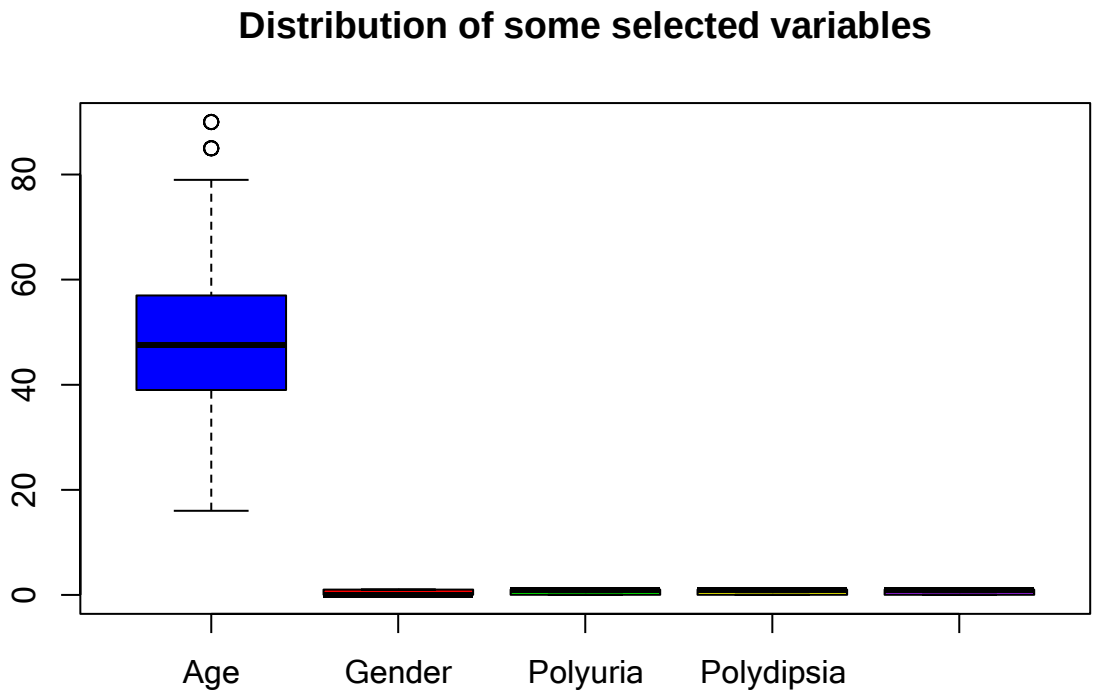

```
data = dat
vnames <- colnames(data)
n <- nrow(data)
out <- NULL
for (j in 1:ncol(data)){
  vname <- colnames(data)[j]
  x <- as.vector(data[,j])
  n1 <- sum(is.na(x), na.rm=TRUE)  # NA
  n2 <- sum(x=="NA", na.rm=TRUE) # "NA"
  n3 <- sum(x==" ", na.rm=TRUE)  # missing
  nmiss <- n1 + n2 + n3
  nmiss <- sum(is.na(x))
  ncomplete <- n-nmiss
  out <- rbind(out, c(col.num=j, v.name=vname, mode=mode(x),
                      n.level=length(unique(x)),
                      ncom=ncomplete, nmiss= nmiss, miss.prop=nmiss/n))
}
out <- as.data.frame(out)
row.names(out) <- NULL
out
```

```
##    col.num             v.name    mode n.level ncom nmiss miss.prop
## 1        1                Age numeric      51  520     0         0
## 2        2             Gender numeric       2  520     0         0
## 3        3           Polyuria numeric       2  520     0         0
## 4        4         Polydipsia numeric       2  520     0         0
## 5        5 sudden.weight.loss numeric       2  520     0         0
## 6        6           weakness numeric       2  520     0         0
## 7        7         Polyphagia numeric       2  520     0         0
## 8        8     Genital.thrush numeric       2  520     0         0
## 9        9    visual.blurring numeric       2  520     0         0
## 10      10            Itching numeric       2  520     0         0
## 11      11       Irritability numeric       2  520     0         0
## 12      12    delayed.healing numeric       2  520     0         0
## 13      13    partial.paresis numeric       2  520     0         0
## 14      14   muscle.stiffness numeric       2  520     0         0
## 15      15           Alopecia numeric       2  520     0         0
## 16      16            Obesity numeric       2  520     0         0
## 17      17              class numeric       2  520     0         0
```

The categorical variables in the data set were all encoded to take a numerical values.Predictor variable Gender was encoded as having value “0” to represent Male and “1” to represent Female, the remaining predictor variables were encoded as having value “0” to represent Yes and “1” to represent No.

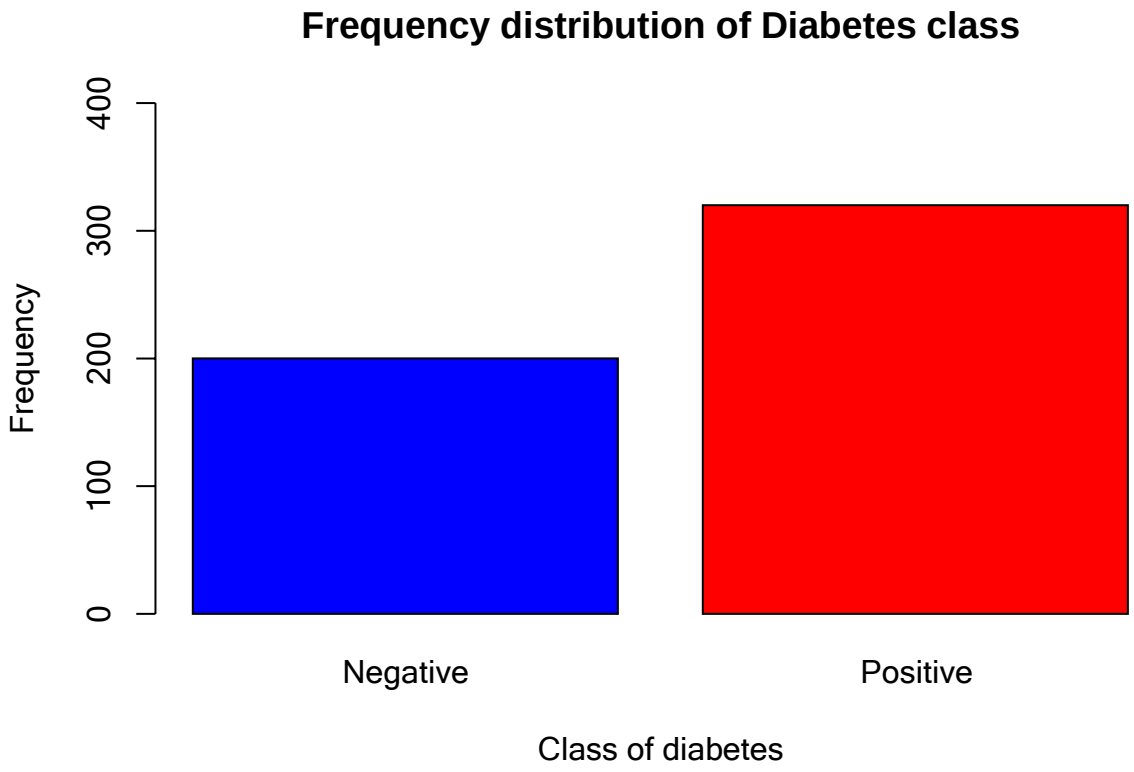
16 out of the 17 variables are binary(Categorical) except the predictor variable Age which is continuous (originally discreet but behaves in a continuous way) . The plot and the analysis above also shows that there is no presence of missing values in the data set and no presence of any wrong records after full inspection of the Data set.

```
boxplot(dat[,1:5],col=c("blue","red","green","yellow","purple"),
        main="Distribution of some selected variables")
```



The categorical Predictor variables in the data set cannot contain an outlier since they have been categorized into two classes. The Boxplot above was plotted for the continuous predictor variable Age to inspect if there is any occurrence of potentials outliers. The plot shows there exist two potential outliers in the data set.
I extracted the potential outliers 85,90,85,90 from rows 102,103,186,187 of the data set. Since the potential outliers are not from the rest of the Age observation,then they do not pose any threat in the data set and therefore be retained in the dataset.
In other to explore the frequency distribution of the output variable in the data set. I obtain the Bar plot of the target variable outcome. The associationof the Target variable on some selected predicted variables will also be observed.

```
diabetes_class= dat$class
diabetes_class<- ifelse(diabetes_class==0, "Positive", "Negative")
tab=table(diabetes_class,useNA="no")
barplot(tab,names.arg=row.names(tab),col=c("blue","red"),ylab="Frequency",
        xlab="Class of diabetes",
        main="Frequency distribution of Diabetes class",ylim=c(0,400))
```



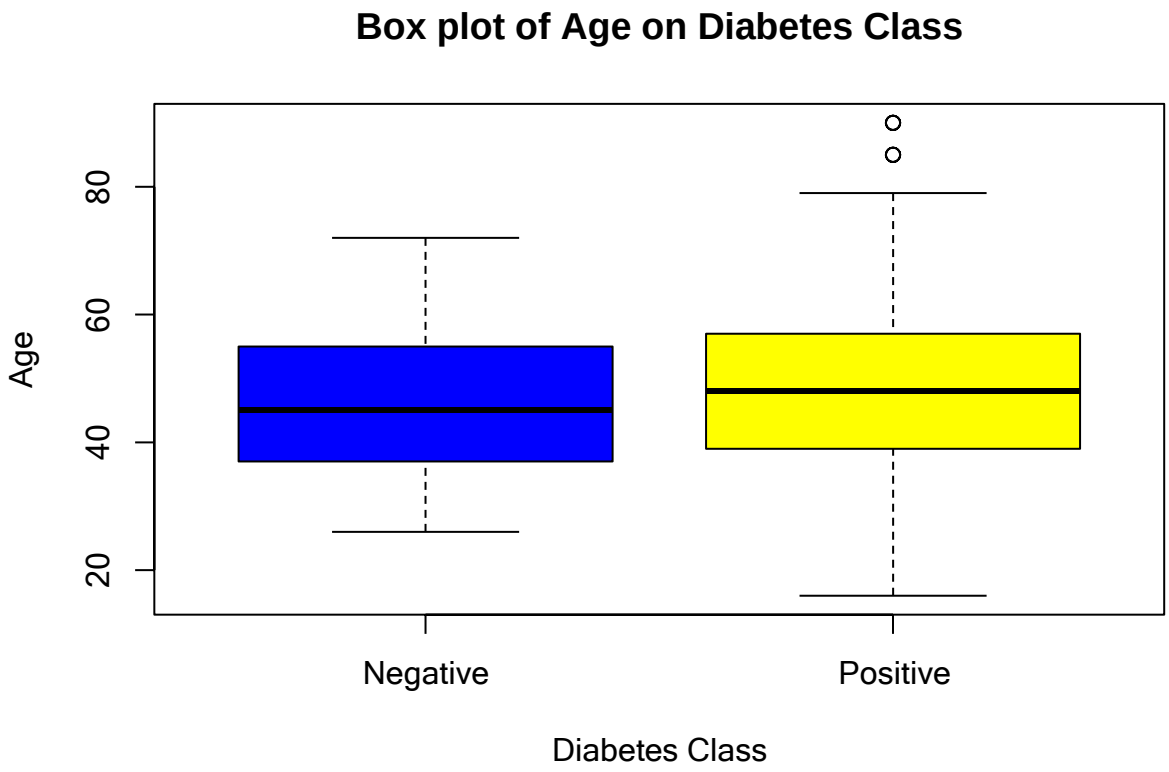
The plot above shows that the frequency distribution of the target variable "class". I have 320 (62%) frequency of class "Positive" and 200 (38%)frequency of class "200". This shows I have an unbalanced classification problem.

4 VARIABLE SCREENING

I am going to explore the marginal (bi-variate) associations between class and each attribute/predictor to identify the important predictor variables and the unimportant ones in the data set.

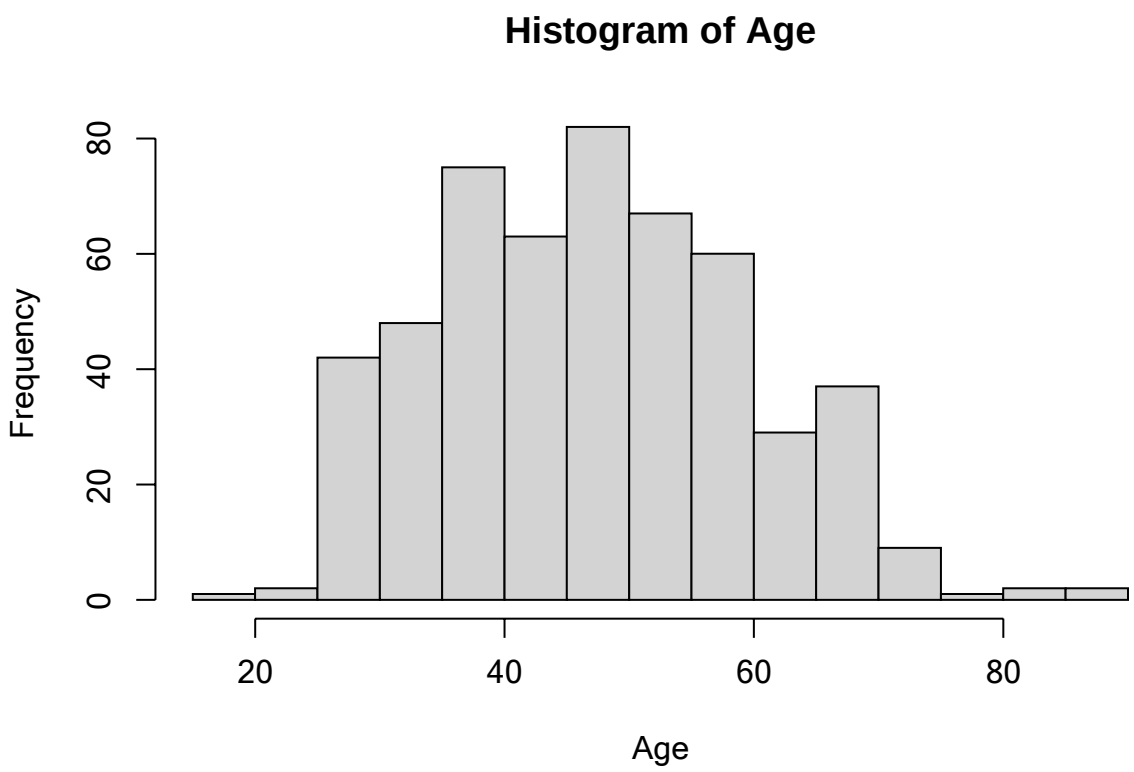
Diabetes class and Age

```
boxplot<-boxplot(dat$Age~diabetes_class,xlab="Diabetes Class", ylab="Age"
 ,main="Box plot of Age on Diabetes Class",col=c("blue","yellow"))
```



The Boxplot above shows the distribution of the predictor variable Age on Diabetes Status, which indicate the presence of association between predictor variable Age on target variable “class”.

```
hist(dat$Age,main="Histogram of Age",xlab="Age")
```



Since the predictor variable Age is continuous and the target variable is categorical ,therefore a parametric two-sample t-test or the non parametric Wilcoxon rank-sum test will be employed to test the association between the two variables.

The plot shows the distribution of the Age is a little bit skewed to the right.

But it is generally close to normality.

Further evidence will be presented using the two normality test below.

```
library(nortest)
ad.test(dat$Age)
```

```
##
##  Anderson-Darling normality test
##
## data:  dat$Age
## A = 1.9758, p-value = 4.871e-05
```

```
shapiro.test(dat$Age)
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  dat$Age
## W = 0.98313, p-value = 9.992e-06
```

The p-value of both test are all less than alpha level of significance 0.05, which indicate that the distribution is not normal. Therefore the assumption of t-test is violated, then I use the non parametric alternative approach called wilcoxonrank sum test to examine the association between Age and Diabetes Class.

```
age_test=wilcox.test(dat$Age~diabetes_class, data=dat, alternative = "two.sided"
            ,na.action = na.omit)
age_test
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  dat$Age by diabetes_class
## W = 27834, p-value = 0.0124
## alternative hypothesis: true location shift is not equal to 0
```

The p-value of the test above which is less than alpha level of significance 0.05 indicate there exist a significance relationship between predictor variable Age and target variable “class”.

Chi-square test or Fisher Exact test will be use to test the association between the categorical predictor variable in the data set and the target variable Diabetes class.

```
chisq1=chisq.test(dat$class,dat$Gender)
chisq2=chisq.test(dat$class,dat$Polyuria)
chisq3=chisq.test(dat$class,dat$Polydipsia)
chisq4=chisq.test(dat$class,dat$sudden.weight.loss)
chisq5=chisq.test(dat$class,dat$weakness)
chisq6=chisq.test(dat$class,dat$Polyphagia)
chisq7=chisq.test(dat$class,dat$Genital.thrush)
chisq8=chisq.test(dat$class,dat$visual.blurring)
chisq9=chisq.test(dat$class,dat$Itching)
chisq10=chisq.test(dat$class,dat$Irritability)
chisq11=chisq.test(dat$class,dat$delayed.healing)
chisq12=chisq.test(dat$class,dat$partial.paresis)
chisq13=chisq.test(dat$class,dat$muscle.stiffness)
```

```
chisq14=chisq.test(dat$class,dat$Alopecia)
chisq15=chisq.test(dat$class,dat$Obesity)


tab <- matrix(c(
  age_test$statistic,round(age_test$p.value,digits = 5),"important",
    chisq1$statistic,round(chisq1$p.value,digits=5),"important",
  chisq2$statistic,round(chisq2$p.value,digits=5),"important",
    chisq3$statistic,round(chisq3$p.value,digits =5),"important",
    chisq4$statistic,round(chisq4$p.value,digits =5),"important",
    chisq5$statistic,round(chisq5$p.value,digits  =5),"important",
  chisq6$statistic,round(chisq6$p.value,digits =5),"important",
    chisq7$statistic,round(chisq7$p.value,digits  =5),"important",
  chisq8$statistic,round(chisq8$p.value,digits =5),"important",
    chisq9$statistic,round(chisq9$p.value,digits  =5),"unimportant",
  chisq10$statistic,round(chisq10$p.value,digits =5),"important",
  chisq11$statistic,round(chisq11$p.value,digits  =5),"unimportant",
    chisq12$statistic,round(chisq12$p.value,digits  =5),"important",
    chisq13$statistic,round(chisq13$p.value,digits  =5),"important",
    chisq14$statistic,round(chisq14$p.value,digits  =5),"important",
    chisq15$statistic,round(chisq15$p.value,digits =5),"important"
      ), ncol=3, byrow=TRUE)
colnames(tab) <- c("Test-Statistic","P-value","Decision")
rownames(tab) <- c("Age","Gender","Polyuria","Polydipsia","sudden.weight.loss",
                   "weakness",
                   "Polyphagia","Genital.thrush","visual.blurring","Itching"
                   ,"Irritability","delayed.healing","partial.paresis",
                   "muscle.stiffness",
                   "Alopecia","Obesity"
                   )
tab <- as.table(tab)
tab
```

```
##                   Test-Statistic     P-value Decision
## Age               27834              0.0124  important
## Gender            103.036859279726   0       important
## Polyuria          227.865838954968   0       important
## Polydipsia        216.171632695787   0       important
## sudden.weight.loss 97.2963034782741  0       important
## weakness          29.7679184140297   0       important
## Polyphagia        59.5952535372963   0       important
## Genital.thrush    5.79214855752817   0.0161  important
## visual.blurring   31.8084558328722   0       important
## Itching           0.0462354369291347 0.82975 unimportant
```

```
## Irritability        45.2083484408992   0        important
## delayed.healing     0.962093688113284  0.32666  unimportant
## partial.paresis     95.3876274432915   0        important
## muscle.stiffness    7.28866666666667   0.00694  important
## Alopecia            36.0641434165042   0        important
## Obesity             2.32747395833333   0.12711  important
```

The above table is the result from the test of association between each predictor variables and Target variable Diabetes Class. Wilcoxon test was used for variable Age and I have chisquare for the rest of the predictor variables.

The p-values were compared against liberal threshold significance level of 0.25 decision were made on whether a particular predictor variable is important or not for the Model buliding.

Predictor variable Itching and Delayed healing were considers unimportant based on their respective p-values. Therefore, I will remove them from the dataset.

I also considered the correlation of the predictor variables against the target variablein the correlation plot below.

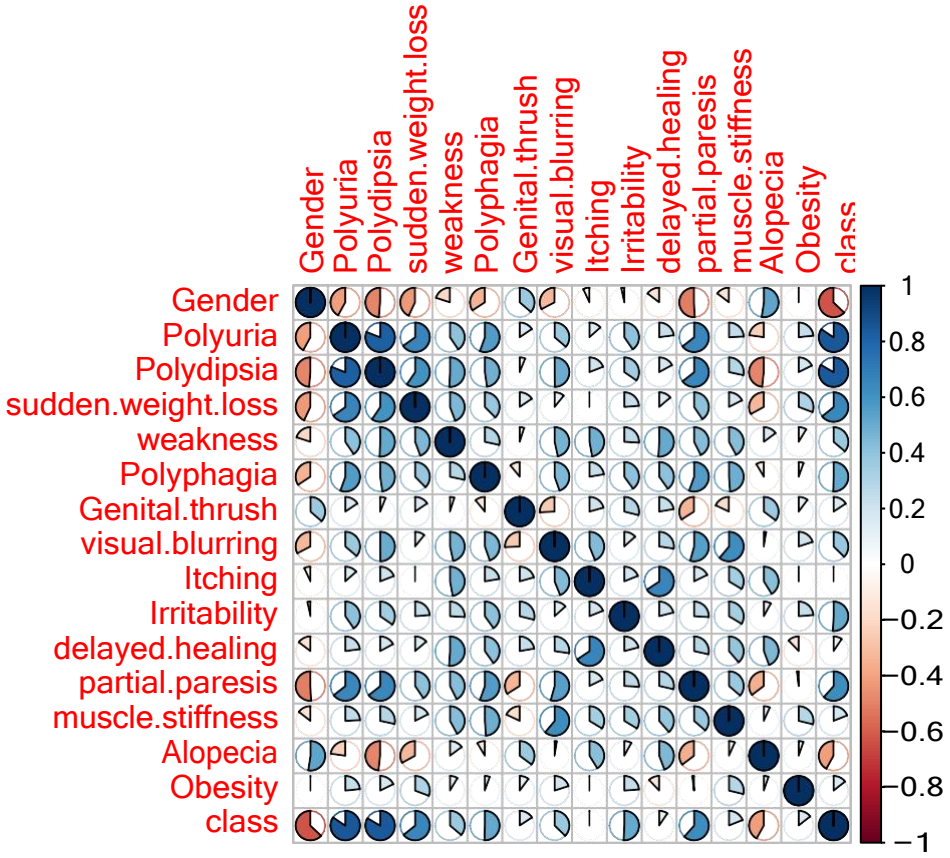
Tetrachoric correlation was used to determine the strength of relationship between the cathegorical predictor variables and the target variable.

Also, a Biserial Correlation was used to determine the strength of relationship between the continuous predictor variable Age and the target variable Diabetes class.

```
library(psych)
d=tetrachoric(dat[,2:17])
```

```
## Warning in cor.smooth(mat): Matrix was not positive definite, smoothing was done
```

```
library(corrplot)
corrplot(d$rho,  method="pie")
```



It can be deduced from the plot that predictor variable Itching and Delayed healing has the least correlation with the target variable Diabetes class.

```
library(ltm)
```

```
## Warning: package 'ltm' was built under R version 4.1.2
```

```
## Warning: package 'polycor' was built under R version 4.1.2
```

```
biserial.cor(dat$Age,dat$class, use = c("all.obs"), level = 2)
```

```
## [1] -0.108679
```

The biseral correlation of value -0.109 also shows the strength of the relationship between Age and Diabetes class in our dataset.

Since the Predictor variable Itching and Delayed healing were considers unimportant based on their respective p-values. Therefore, I will remove them from the dataset. which leave us with 15 variables in total. 14 predictor and 1 target.

```
dat= dat[, -c(10,12)]
dim(dat)
```

```
## [1] 520  15
```

5 DATA PARTITIONING

Partitioning our data set into train and test with a ratio of approximately 2:1 in sample size. The train data takes 67% of the whole data set and the test takes 33% of the whole data set.

```
set.seed(123)
n <- nrow(dat)
split_data <- sample(x=1:2, size = n, replace=TRUE, prob=c(0.67, 0.33))
D1<- dat[split_data == 1, ]
D2 <- dat[split_data == 2, ]
yobs <- D2$class
D2<- D2[ , -15]
```

I have the training data set with “D1” as training input. Finally I have the testingdata set with “D2” as testing input and “yobs” as testing output.

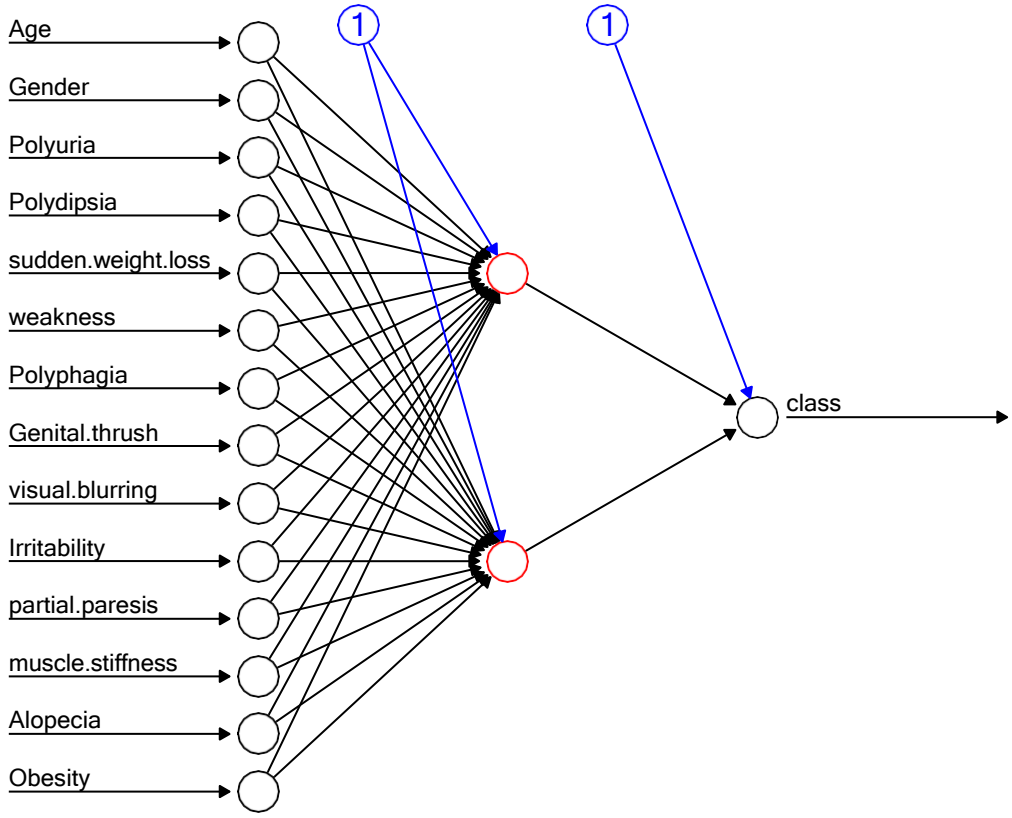
6 MODEL BUILDING

Since our Target variable is binary (Categorical), I am resorting to a Classifier machine model. In this project I will be using a Artificial Neural Network to predict Diabetes status of a patient.I am going to explore ANN under different numbers of layers and different number of neurons for the classification of the Diabetes class.

6.1 ANN with 1 hidden layer and 2 neuron

```
set.seed(123)
library(neuralnet)
library(caret)
options(digits=3)
```

```
net1 <- neuralnet(class ~ .,
         data=D1,hidden=2,
    act.fct="logistic", err.fct="sse", linear.output=FALSE, likelihood=TRUE)
plot(net1, rep="best", show.weights=FALSE, dimension=6.5, radius=.15,
     col.hidden="red", col.hidden.synapse="black", lwd=1, fontsize=9)
```



The plot shows the weights of the developed ANN model in neurons and also the hidden layers produced by the neural network.

With predicted Values:

```
ypred <- compute(net1, covariate=D2)$net.result
MSE.c <- mean((yobs-ypred)^2)
MSE.c
```

```
## [1] 0.084
```

```
ypred <- ifelse(ypred>0.5,1,0)
as.vector(ypred)
```

```
##   [1] 1 1 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
##  [38] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 1 0 1 0 1 0 1 1 1 1 1 0 0 1
## [112] 1 0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1
## [149] 1 1 1 0 0 0 1 1 0 1 0 1 1 1 0 0 1
```

An MSE of 8% shows the strength of the trained model in predicting the original simulation Diabetes classes.

6.2 Model Evaluation

It is the phase that is decided whether the model performs better. Therefore, it is critical to consider the model outcomes according to every possible evaluation method. Applying different methods can provide different perspectives.

```
library(cvAUC)
rf_AUC <- ci.cvAUC(predictions = ypred, labels =yobs, folds=1:NROW(D2), confidence = 0.9
rf_AUC
```

```
## $cvAUC
## [1] 0.913
##
## $se
## [1] 0.0436
##
## $ci
## [1] 0.828 0.999
##
## $confidence
## [1] 0.95
```

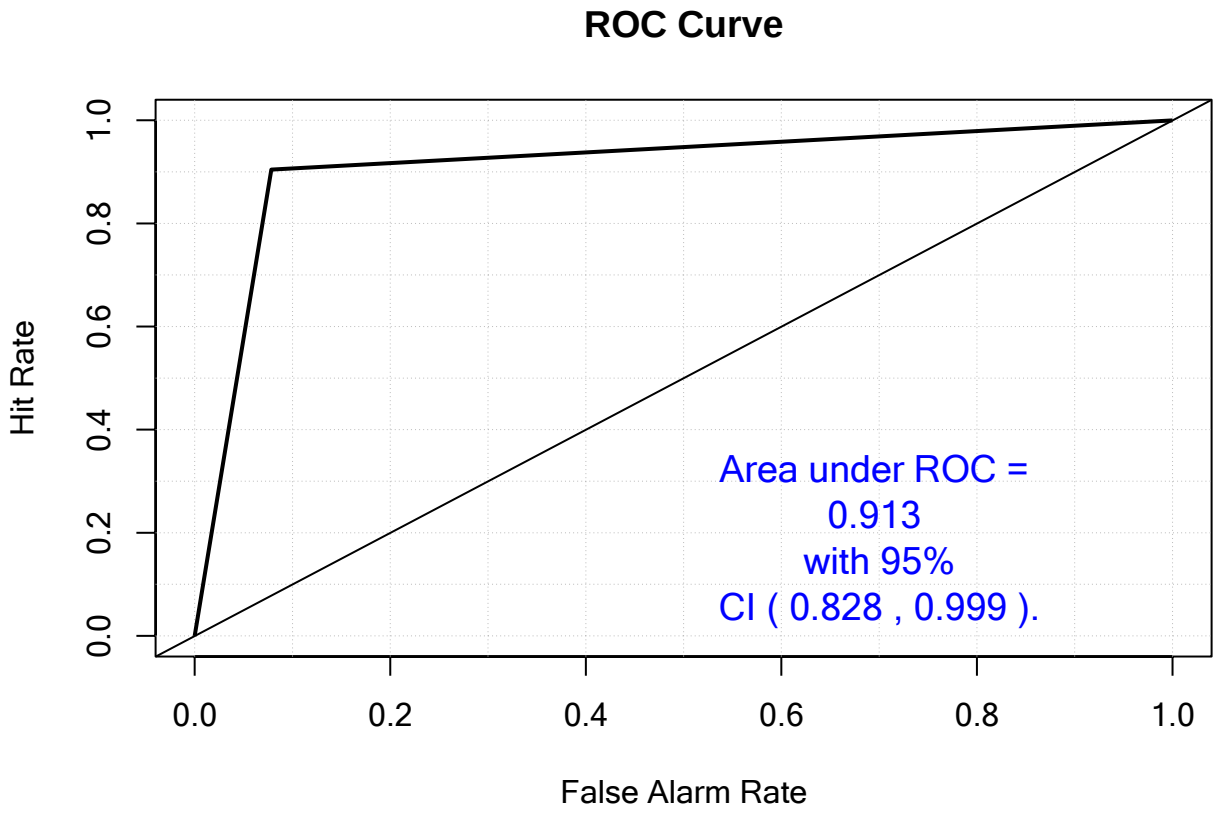
```
rf_auc.ci <- round(rf_AUC$ci, digits = 3)
```

AUC of 0.913 indicate that our proposed ANN model has 91% ability to classify correctly either the Diabetes class is either “Positive” or “Negative”.
The confidence interval also indicate the true AUC falls within the interval (0.828 ,0.999). Therefore I am 95% confident that our AUC is accurate.

```
library(verification)
mod.nn <- verify(obs = yobs, pred = ypred)
```

```
## If baseline is not included, baseline values  will be calculated from the  sample obs
```

```
roc.plot(mod.nn,  plot.thres=NULL)
text(x=0.7,  y=0.2,  paste("Area under ROC = \n",
                           round(rf_AUC$cvAUC, digits = 3),
                           "\n with 95% \nCI (",
 rf_auc.ci[1],  ",",  rf_auc.ci[2],  ").",  sep = " "),  col="blue",  cex =1.2)
```



The ROC curve above shows the trade-off between sensitivity (or TPR) and False Positive Rate(1 – Specificity). It further indicates that the model performs better against the bench mark (50%) with total area of 0.913(91%) and a misclassification rate of 0.0848 (8%) below.

```
(miss.rate <- mean(yobs != ypred))
```

```
## [1] 0.0848
```

```
confusionMatrix(factor(yobs),  factor(ypred))
```

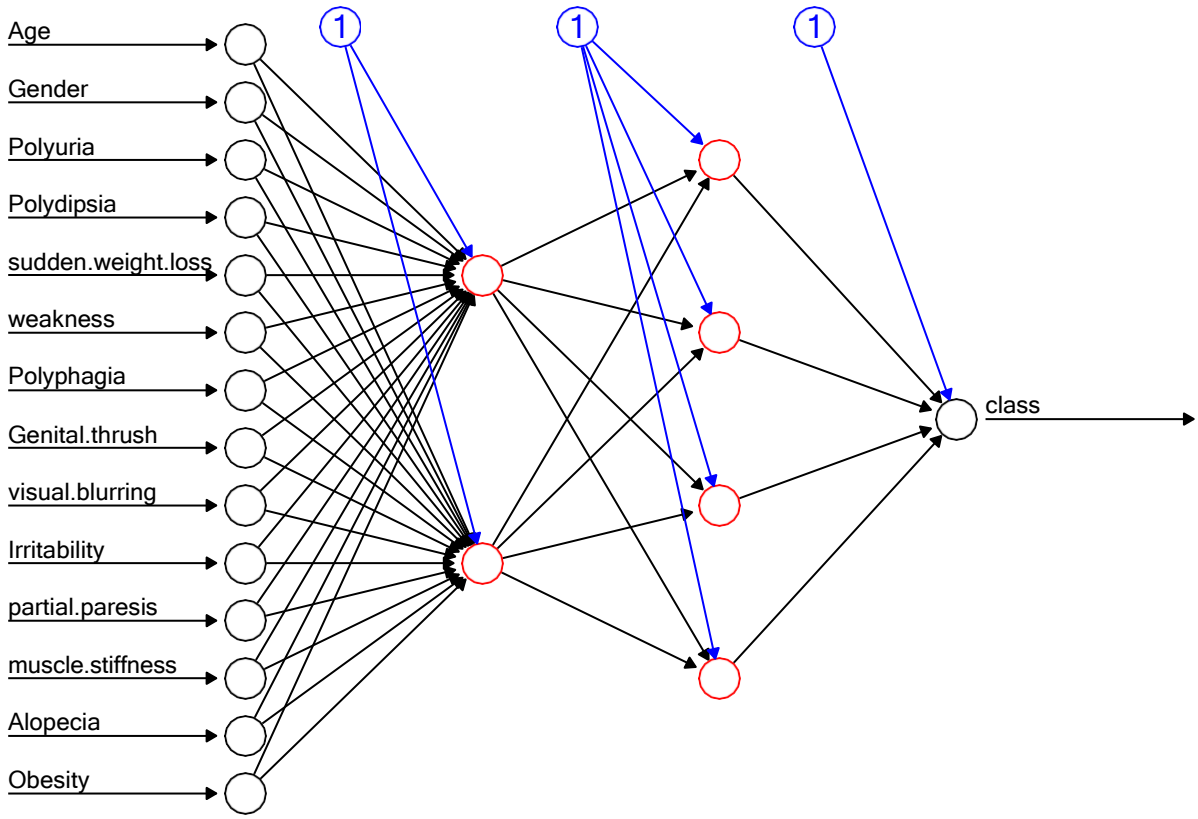
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 94  8
##          1  6 57
##
##                Accuracy : 0.915
##                  95% CI : (0.862, 0.953)
##     No Information Rate : 0.606
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.821
##
##  Mcnemar's Test P-Value : 0.789
##
##             Sensitivity : 0.940
##             Specificity : 0.877
##          Pos Pred Value : 0.922
##          Neg Pred Value : 0.905
##              Prevalence : 0.606
##          Detection Rate : 0.570
##    Detection Prevalence : 0.618
##       Balanced Accuracy : 0.908
##
##       'Positive' Class : 0
##
```

From the result of the Confusion matrix, the positive is represented as "Positive" with the value "0" and the negative is represented as "Negative" with value "1". Therefore, the Sensitivity (True Positive rate) also Known as "Recall" of 0.940 (94%) calculated as $\frac{94}{94+6}$ shows the model has a high percentage of detecting a Postive status of Diabetes of a Diabetic patient.
The Specificity(True Negative rate) of 0.877 (88%) calculated as $\frac{57}{57+8}$ shows the model has a high percentage of detecting a Negative status of Diabetes.
Our trained ANN Model has an accuracy of 91.5% in terms of performance with a precision (Pos Pred Value) of 92% $\frac{94}{94+8}$ that indicate our Model has a low false positive rate (very low wrong classification of a Positive Diabetes status). Finally with a Negative Predicted Value of 90.5% $\frac{57}{57+6}$ that indicate

our Model has a very low false negative rate (a very low wrong lassification of a Negative Diabetes status).

6.3 ANN with 2 hidden layers and 2,4 neurons respectively

```
set.seed(123)
library(neuralnet)
options(digits=3)
net1 <- neuralnet(class ~ .,
                  data=D1,
                  hidden=c(2,4), #1 hidden layer, 2 neurons
  act.fct="logistic", err.fct="sse", linear.output=FALSE, likelihood=TRUE)
plot(net1, rep="best", show.weights=FALSE, dimension=6.5, radius=.15,
     col.hidden="red", col.hidden.synapse="black", lwd=1,
     fontsize=9,arrow.length=0.19)
```



The plot shows the weights of the developed ANN model in neurons and also the hidden layers produced by the neural network.

With predicted Values:

```
ypred <- compute(net1, covariate=D2)$net.result
MSE.c <- mean((yobs-ypred)^2)
MSE.c
```

```
## [1] 0.0722
```

```
ypred <- ifelse(ypred>0.5,1,0)
as.vector(ypred)
```

```
##   [1] 1 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1
##  [75] 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 1 0 0 1
## [112] 1 0 0 0 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1
## [149] 1 1 1 0 0 0 1 1 1 1 0 1 1 1 0 0 1
```

An MSE of 7% shows the strength of the trained model in predicting the original simulation Diabetes classes.

6.4 Model Evaluation

```
library(cvAUC)
rf_AUC <- ci.cvAUC(predictions = ypred, labels =yobs, folds=1:NROW(D2), confidence = 0.9
rf_AUC
```

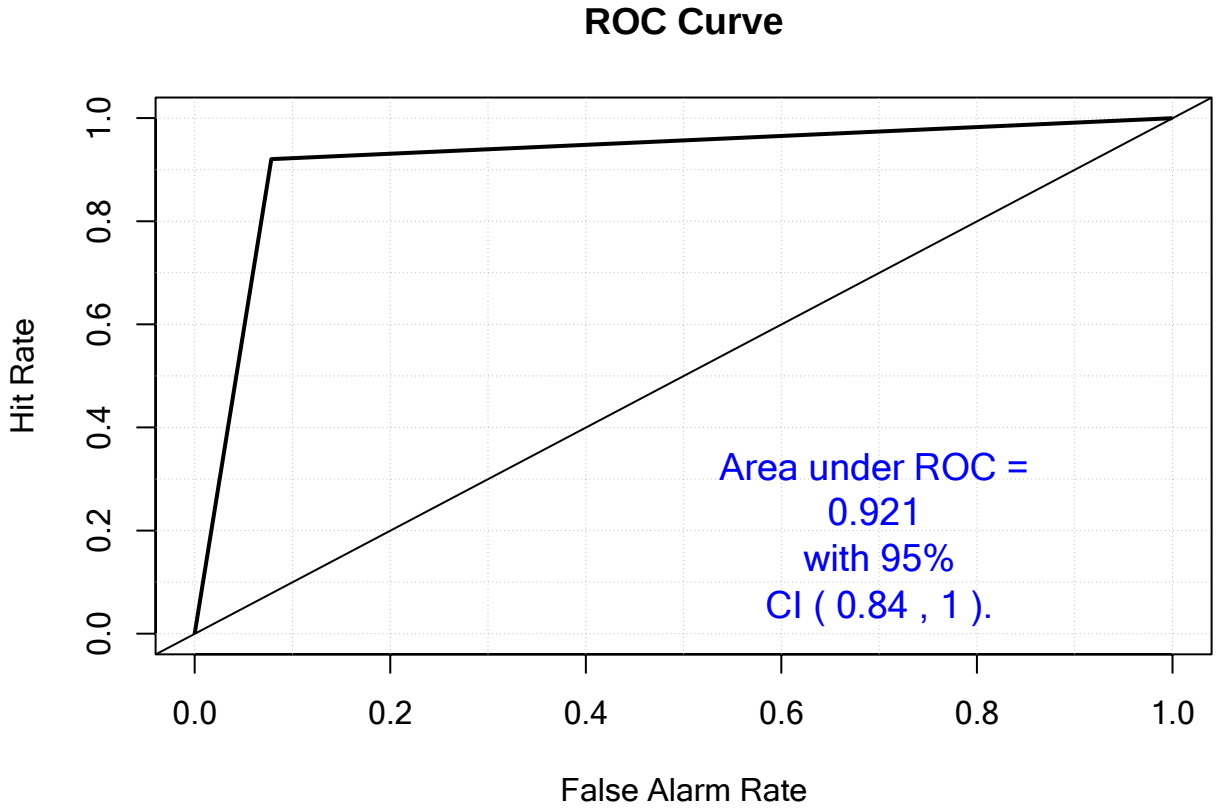
```
## $cvAUC
## [1] 0.921
##
## $se
## [1] 0.0415
##
## $ci
## [1] 0.84 1.00
##
## $confidence
## [1] 0.95
```

AUC of 0.921 indicate that our proposed ANN model has 92% ability to classify correctly either the Diabetes class is either “Positive” or “Negative”.
The confidence interval also indicate the true AUC falls within the interval (0.84 ,1.00). Therefore I am 95% confident that our AUC is accurate.

```
rf_auc.ci <- round(rf_AUC$ci, digits = 3)
library(verification)
mod.nn <- verify(obs = yobs, pred = ypred)
```

```
## If baseline is not included, baseline values  will be calculated from the  sample obs
```

```
roc.plot(mod.nn,  plot.thres=NULL)
text(x=0.7, y=0.2, paste("Area under ROC = \n",
                         round(rf_AUC$cvAUC, digits = 3),
                         "\n with 95% \nCI (",
 rf_auc.ci[1], ",", rf_auc.ci[2], ").", sep = " "), col="blue", cex =1.2)
```



The ROC curve above shows the trade-off between sensitivity (or TPR) and False Positive Rate(1 – Specificity). It further indicates that the model performs better against the bench mark (50%) with total area of 0.921(92%) with a misclassification rate of 0.0788 (7.9%) below.

```
(miss.rate <- mean(yobs != ypred))
```

```
## [1] 0.0788
```

```
confusionMatrix(factor(yobs), factor(ypred))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 94  8
##          1  5 58
##
##                Accuracy : 0.921
##                  95% CI : (0.869, 0.957)
##     No Information Rate : 0.6
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.835
##
##  Mcnemar's Test P-Value : 0.579
##
##             Sensitivity : 0.949
##             Specificity : 0.879
##          Pos Pred Value : 0.922
##          Neg Pred Value : 0.921
##              Prevalence : 0.600
##          Detection Rate : 0.570
##    Detection Prevalence : 0.618
##       Balanced Accuracy : 0.914
##
##       'Positive' Class : 0
##
```

From the result of the Confusion matrix, the positive is represented as "Positive" with the value "0" and the negative is represented as "Negative" with value "1". Therefore, the Sensitivity (True Positive rate) also Known as "Recall" of 0.949 (95%) calculated as $\frac{94}{94+5}$ shows the model has a high percentage of detecting a Postive status of Diabetes of a Diabetic patient.

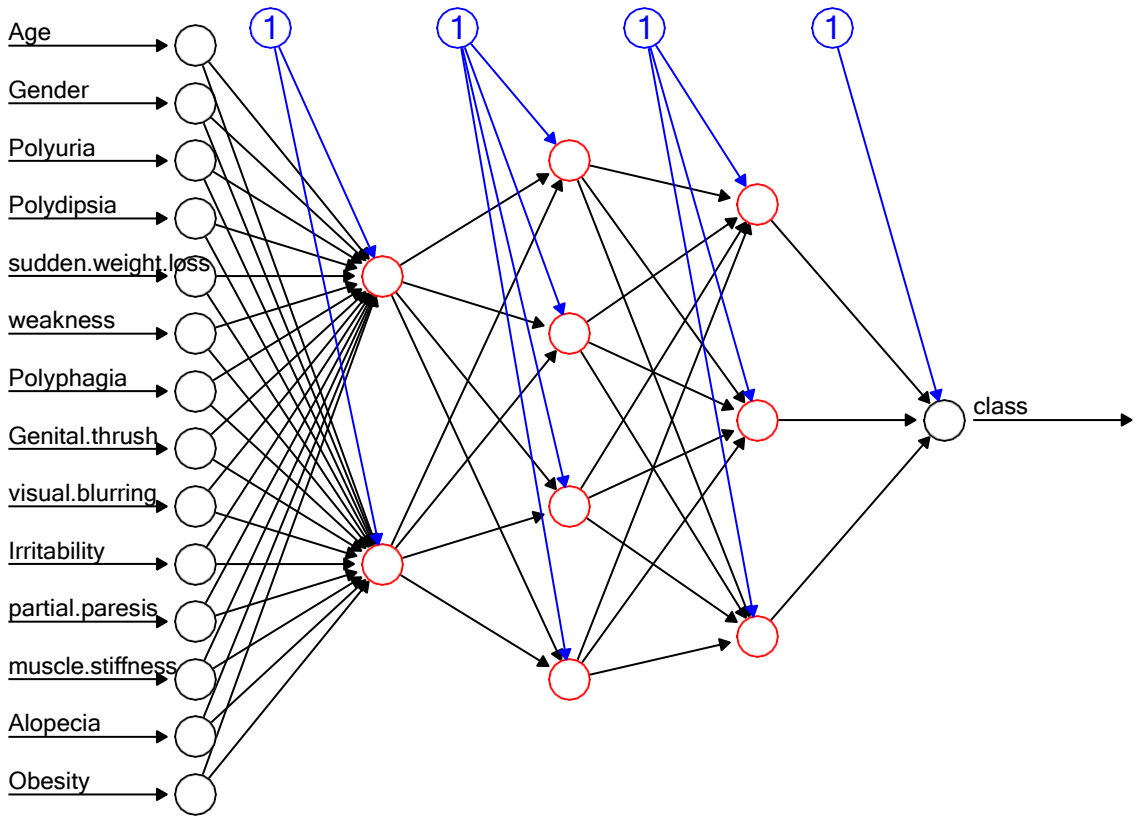
The Specificity(True Negative rate) of 0.879 (88%) calculated as $\frac{58}{58+8}$ shows the model has a high percentage of detecting a Negative status of Diabetes.

Our trained ANN Model has an accuracy of 92.1% in terms of performance with a precision (Pos Pred Value) of 92% $\frac{94}{94+8}$ that indicate our Model has

a low false positive rate (very low wrong classification of a Positive Diabetes status). Finally with a Negative Predicted Value of 92.1% $\frac{58}{58+5}$ that indicate our Model has a very low false negative rate (a very low wrong lassification of a Negative Diabetes status).

6.5 ANN with 3 hidden layers and 2,4 ,3 neurons respectively

```
set.seed(123)
library(neuralnet)
options(digits=3)
net1 <- neuralnet(class ~ .,
                  data=D1,
      hidden=c(2,4,3), #1 hidden layer, 2 neurons
act.fct="logistic", err.fct="sse", linear.output=FALSE, likelihood=TRUE)
plot(net1, rep="best", show.weights=FALSE, dimension=6.5, radius=.15,
     col.hidden="red", col.hidden.synapse="black", lwd=1,
     fontsize=9,arrow.length=0.15)
```



The plot shows the weights of the developed ANN model in neurons and also the

hidden layers produced by the neural network.

With predicted Values:

```
ypred <- compute(net1, covariate=D2)$net.result
MSE.c <- mean((yobs-ypred)^2)
MSE.c
```

```
## [1] 0.0704
```

```
ypred <- ifelse(ypred>0.5,1,0)
as.vector(ypred)
```

```
##   [1] 1 1 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1
##  [75] 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 1 0 1 0 1 1 1 1 1 1 0 0 1
## [112] 1 0 0 0 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 1 1
## [149] 1 1 1 0 0 0 1 1 1 1 0 1 1 1 0 0 1
```

An MSE of 7% shows the strength of the trained model in predicting the original simulation Diabetes classes.

6.6 Model Evaluation

```
library(cvAUC)
rf_AUC <- ci.cvAUC(predictions = ypred, labels =yobs,
                   folds=1:NROW(D2), confidence = 0.95)
rf_AUC
```

```
## $cvAUC
## [1] 0.927
##
## $se
## [1] 0.0386
##
## $ci
## [1] 0.851 1.000
##
## $confidence
## [1] 0.95
```

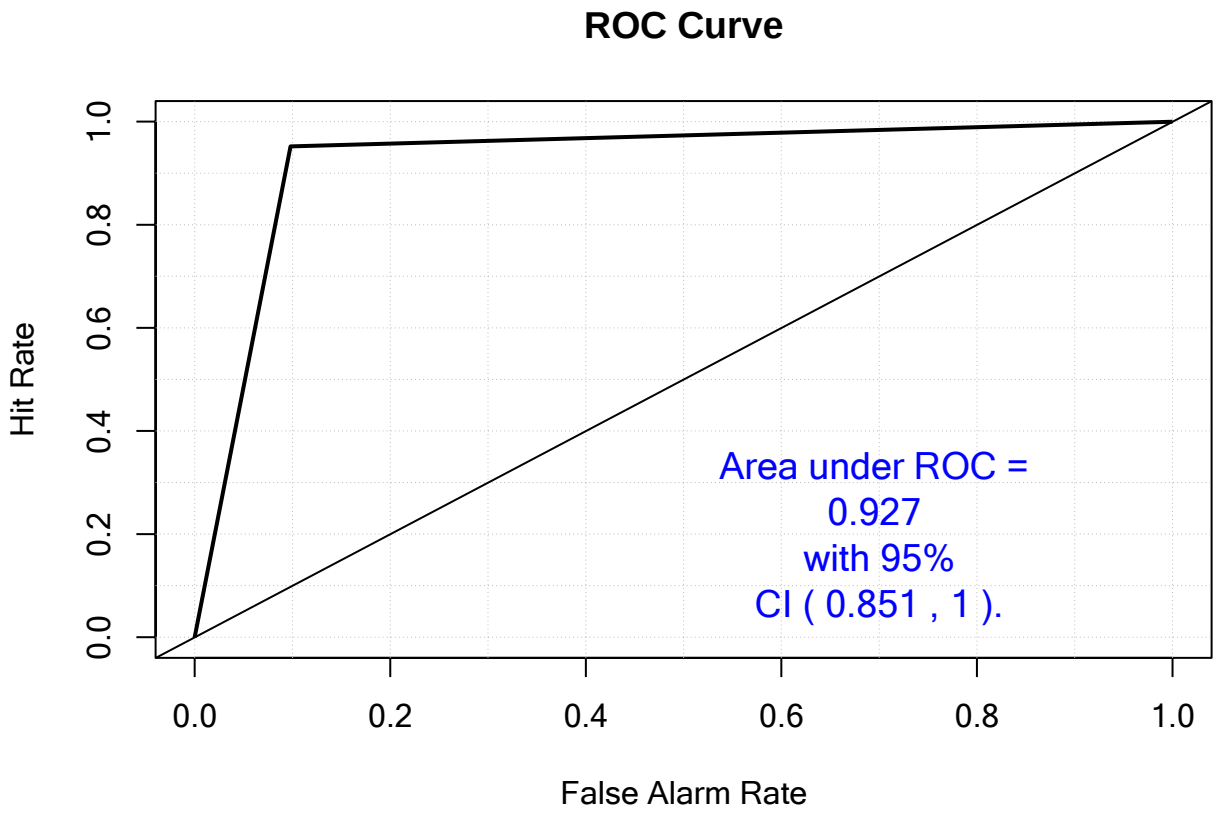
AUC of 0.927 indicate that our proposed ANN model has 93% ability to classify correctly either the Diabetes class is either “Positive” or “Negative”.

The confidence interval also indicate the true AUC falls within the interval (0.851 ,1.00). Therefore I am 95% confident that our AUC is accurate.

```
rf_auc.ci <- round(rf_AUC$ci, digits = 3)
library(verification)
mod.nn <- verify(obs = yobs, pred = ypred)
```

```
## If baseline is not included, baseline values  will be calculated from the  sample obs
```

```
roc.plot(mod.nn,  plot.thres=NULL)
text(x=0.7, y=0.2, paste("Area under ROC = \n",
                         round(rf_AUC$cvAUC, digits = 3),
                         "\n with 95% \nCI (",
 rf_auc.ci[1], ",", rf_auc.ci[2], ").", sep = " "), col="blue", cex =1.2)
```



The ROC curve above shows the trade-off between sensitivity (or TPR) and False Positive Rate(1 – Specificity). It further indicates that the model performs

better against the bench mark (50%) with total area of 0.921(92%) with a misclassification rate of 0.0788 (7.8%) below.

```
(miss.rate <- mean(yobs != ypred))
```

```
## [1] 0.0788
```

```
confusionMatrix(factor(yobs), factor(ypred))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 92 10
##          1  3 60
##
##                Accuracy : 0.921
##                  95% CI : (0.869, 0.957)
##     No Information Rate : 0.576
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.837
##
##  Mcnemar's Test P-Value : 0.0961
##
##             Sensitivity : 0.968
##             Specificity : 0.857
##          Pos Pred Value : 0.902
##          Neg Pred Value : 0.952
##              Prevalence : 0.576
##          Detection Rate : 0.558
##    Detection Prevalence : 0.618
##       Balanced Accuracy : 0.913
##
##       'Positive' Class : 0
##
```

From the result of the Confusion matrix, the positive is represented as "Positive" with the value "0" and the negative is represented as "Negative" with value "1". Therefore, the Sensitivity (True Positive rate) also Known as "Recall" of 0.968 (97%) calculated as $\frac{92}{92+3}$ shows the model has a high percentage of

detecting a Postive status of Diabetes of a Diabetic patient.
The Specificity(True Negative rate) of 0.857 (86%) calculated as $\frac{60}{60+10}$ shows the model has a high percentage of detecting a Negative status of Diabetes.
Our trained ANN Model has an accuracy of 92.1% in terms of performance with a precision (Pos Pred Value) of 90% $\frac{92}{92+10}$ that indicate our Model has a low false positive rate (very low wrong classification of a Positive Diabetes status).
Finally with a Negative Predicted Value of 95.2% $\frac{60}{60+3}$ that indicate our Model has a very low false negative rate (a very low wrong lassification of a Negative Diabetes status).

7 MODEL COMPARISON

Different levels of hidden layers and number of nuerons considered to build the ANN machine learning model.

1. In the first model, I considered 1 hidden layer and 2 neurons
2. Second Model, I considered 2 hidden layers and 2,4 neurons respectively
3. Third Model, I considered 3 hidden layers and 2,4,3 neurons respectively

I have the analysis of the result below:

Evaluation metrics	Model1	Model2	Model3
Accuracy	0.9150	0.9210	0.9210
Sensitivity	0.9400	0.9490	0.9680
Specificity	0.8770	0.8790	0.8570
Precision	0.9220	0.9220	0.9020
Neg. Pred.Val	0.9050	0.9210	0.9520
MSE	0.0840	0.0722	0.0704
Miss.rate	0.0848	0.0788	0.0788
AUC	0.9130	0.9210	0.9270

Three models all performs very Ill in all areas of the evaluation metrics.
Model3 had the highest AUC and Sensitivity with a tie Accuracy value with Model2.

8 SUMMARY AND CONCLUSION

The accuracy of the ANN model was high in all different levels of hidden layers and number of nuerons considered. Though it appears that increase in number of layers is likely to improve the model accuracy but care as to be taken to avoid over fitting the data set and having a bad estimate on the test set.