Parkinson's Disease (PD) Detection from Audio Signal data

Prepared By,

Manthiramoorthy Cheranthian

Introduction

Aim is to discriminate healthy people from those with PD based on biomedical voice measurements

Dataset:

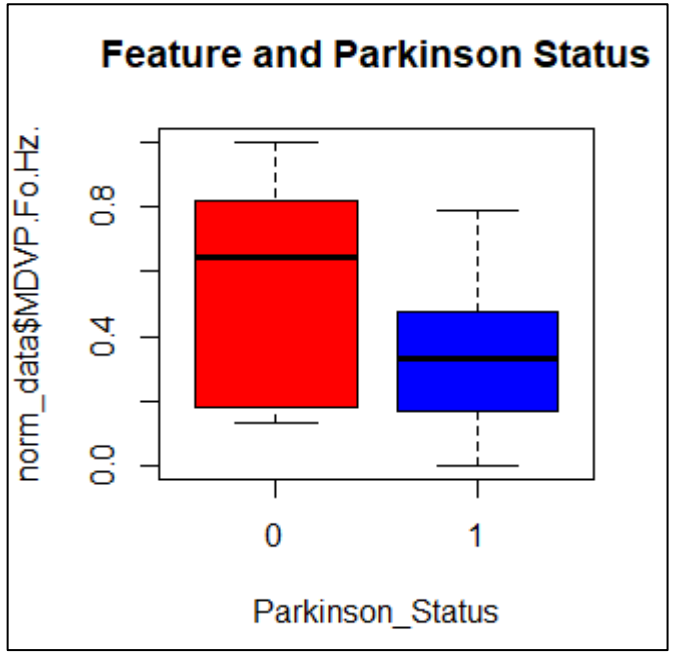
- Dataset has information of 32 individuals (6 recording per patient)
- Data consists of biomedical voice measurements of patients.
- Target variable- Status – Binary 0/1 indicating presence of PD or not.

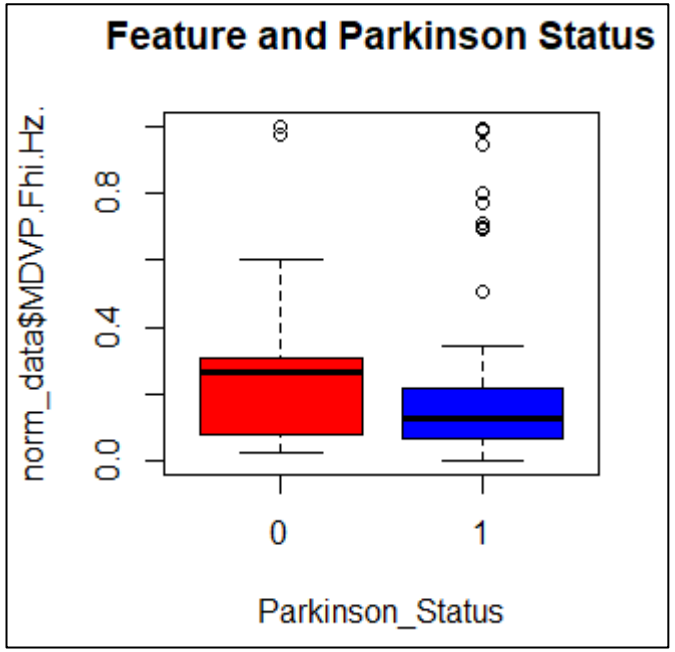
```r
##Creating a function for Normalization
##WE use Min Max Normalization

normalize=function(x){
  return ((x-min(x,na.rm = TRUE))/(max(x,na.rm = TRUE)-min(x,na.rm = TRUE)))

}

##Applying the function to the entire dataset
norm_data=as.data.frame(apply(x, 2, normalize))
norm_data$status=y
str(norm_data)
```

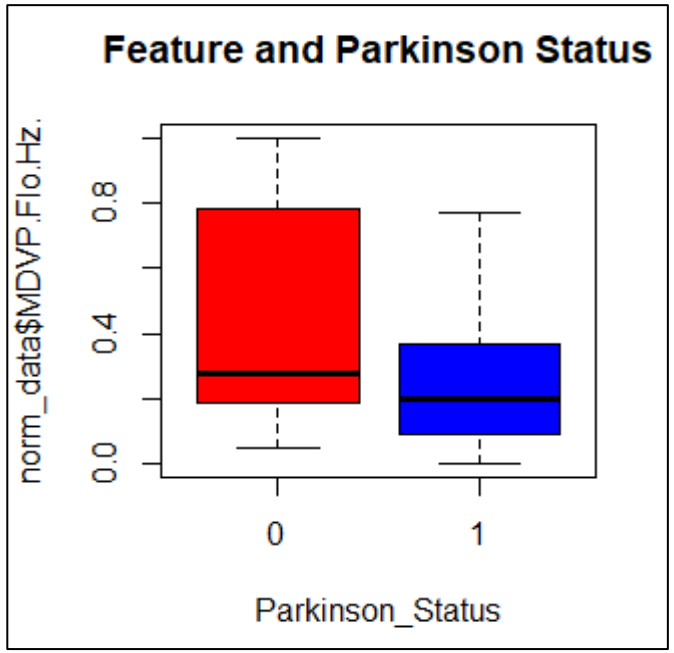
Data Preparation:

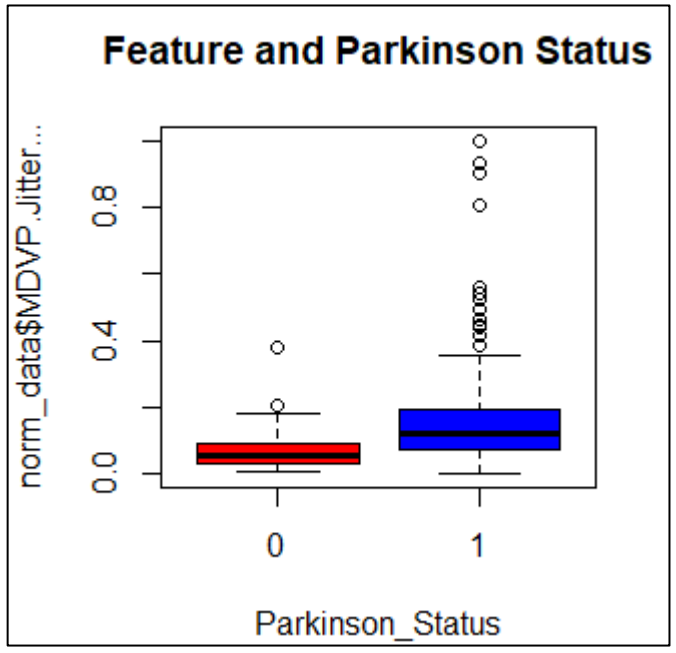
- Normalizing the data since each variable are in different scale ranges.
- Normalizing the data using Min Max scaler function.

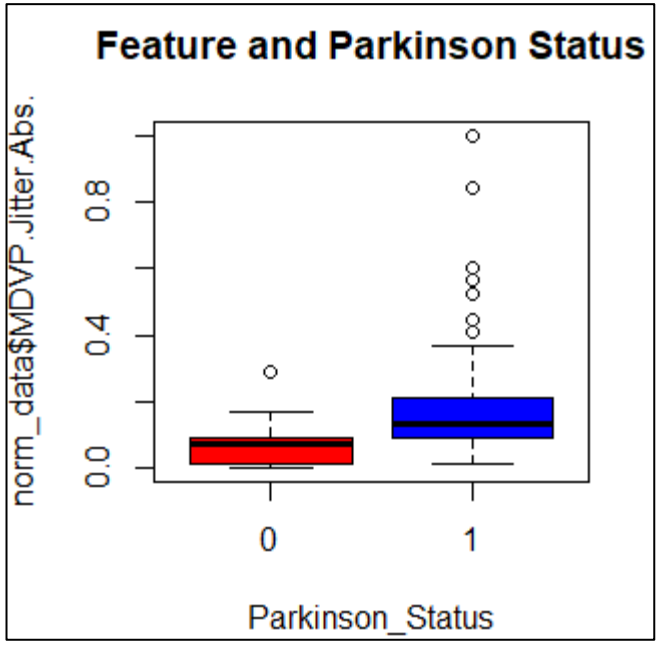
Exploratory Data Analysis

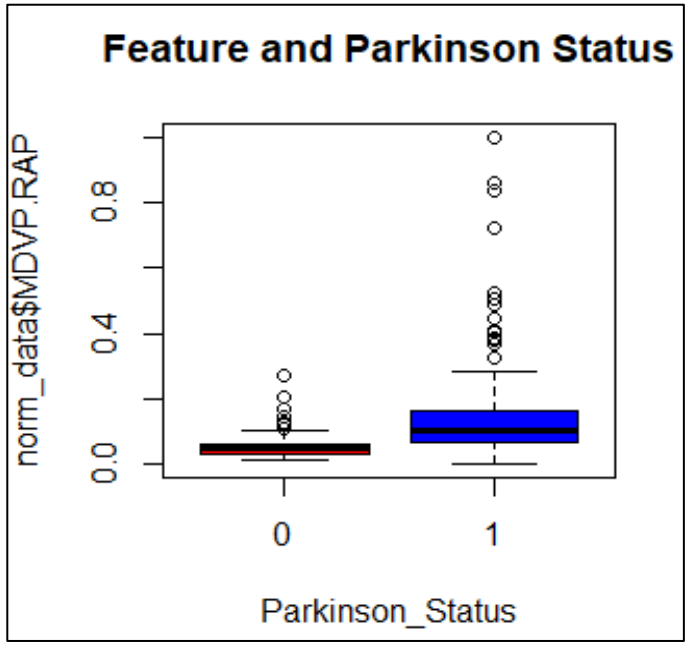

Feature and Parkinson Status
norm_data$MDVP.Fo.Hz.
0.0
0.4
0.8
0
1
Parkinson_Status


Feature and Parkinson Status
norm_data$MDVP.Fhi.Hz.
0.0
0.4
0.8
0
1
Parkinson_Status


Feature and Parkinson Status
norm_data$MDVP.Flo.Hz.
0.0
0.4
0.8
0
1
Parkinson_Status


Feature and Parkinson Status
norm_data$MDVP.Jitter...
0.0
0.4
0.8
0
1
Parkinson_Status


Feature and Parkinson Status
norm_data$MDVP.Jitter.Abs.
0.0
0.4
0.8
0
1
Parkinson_Status


Feature and Parkinson Status
norm_data$MDVP.RAP
0.0
0.4
0.8
0
1
Parkinson_Status

To confirm the significance using t.tests:

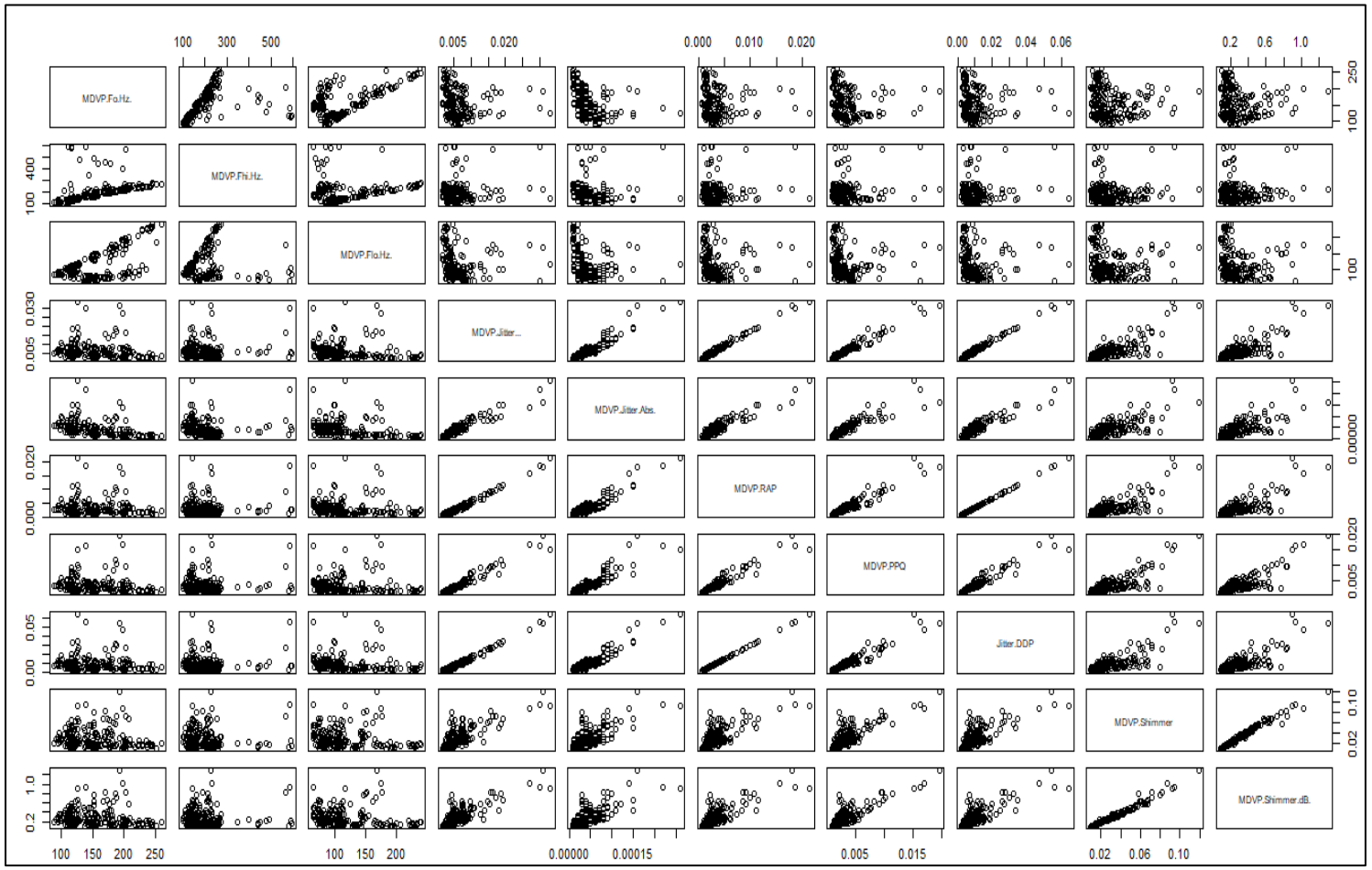
t.tests also proves that all the variables are significant in affecting target PD variable.

```
> t.test(norm_data$MDVP.Fo.Hz.      ~norm_data$status)

        Welch Two Sample t-test

data:  norm_data$MDVP.Fo.Hz. by norm_data$status
t = 4.5575, df = 58.974, p-value = 2.658e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1200342 0.3079402
sample estimates:
mean in group 0 mean in group 1
      0.5449361       0.3309489
```

```
> t.test(norm_data$MDVP.Fhi.Hz.     ~norm_data$status)

        Welch Two Sample t-test

data:  norm_data$MDVP.Fhi.Hz. by norm_data$status
t = 2.2349, df = 74.313, p-value = 0.02843
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.007794628 0.135893329
sample estimates:
mean in group 0 mean in group 1
      0.2480006       0.1761566
```

```
> t.test(norm_data$MDVP.Flo.Hz.     ~norm_data$status)

        Welch Two Sample t-test

data:  norm_data$MDVP.Flo.Hz. by norm_data$status
t = 4.3103, df = 56.54, p-value = 6.585e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1180874 0.3230762
sample estimates:
mean in group 0 mean in group 1
      0.4590331       0.2384513
```

```
> t.test(norm_data$MDVP.Jitter...   ~norm_data$status)

        Welch Two Sample t-test

data:  norm_data$MDVP.Jitter... by norm_data$status
t = -5.9588, df = 187.04, p-value = 1.239e-08
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.13205807 -0.06636696
sample estimates:
mean in group 0 mean in group 1
     0.06944224      0.16865476
```



Testing Collinearity between Independent variables:

- It seems like there are some strong correlations and collinearity problem between independent variables
- It's better to remove the highly collinear variables using VIF else it might weaken our model

Collinearity Reduction:

```
##Selecting significant variables are very essential in developing a string model
##We have the following possible ways to select significant variables
##Regularization (TO penalize the coefficients towards zero for the insgnificant variables)
##Subset Selection
##Exhaustive Regression to look at all possible Linear models but it is inefficient compute wise
##hence we select Stepwise method which selects only restricted possible models which is computatinally efficient

##Claculating VIF
modl=glm(status~.,data=norm_data,family = binomial(link = 'logit'))
library(car)
vif(modl)

##Eliminating variables with VIF > 5 which has higher collinearity

##FInally the selected variables are
##MDVP.Fhi.Hz.
##MDVP.Flo.Hz.
##NHR
##RPDE
##DFA
##spread2
##D2
```

```
##Step Wise Regression to select the significant variables step by step from a model with less AIC

interceptmodl=glm(status~1,data=train,family='binomial')
mod=glm(status~MDVP.Fo.Hz.+MDVP.Flo.Hz.+NHR+RPDE+DFA+spread2+D2,data=train,family='binomial')

step(interceptmodl,direction = 'both',scope = formula(mod))
```

```
Step:  AIC=110.79
status ~ spread2 + MDVP.Flo.Hz. + D2 + DFA

               Df Deviance    AIC
<none>             100.794 110.79
+ MDVP.Fo.Hz.   1   99.462 111.46
+ RPDE          1  100.014 112.01
- spread2       1  104.249 112.25
+ NHR           1  100.783 112.78
- DFA           1  106.977 114.98
- MDVP.Flo.Hz.  1  112.621 120.62
- D2            1  116.595 124.59

Call:  glm(formula = status ~ spread2 + MDVP.Flo.Hz. + D2 + DFA, family = "binomial",
    data = train)

Coefficients:
 (Intercept)       spread2  MDVP.Flo.Hz.            D2           DFA
      -4.086         3.243        -3.412         8.570         3.318

Degrees of Freedom: 151 Total (i.e. Null);  147 Residual
Null Deviance:      168.7
Residual Deviance: 100.8        AIC: 110.8
```

```
##THe final model obtained after step wise regression is
##-4.086+(3.243)*spread2+(-3.412)*MDVP.Flo.Hz.+(8.57)*D2+(3.318)*DFA

##Sigificant variables selected using Step wise regression
##spread2
##MDVP.Flo.Hz.
##D2
##DFA
```

Selecting Significant Variables using Step wise Regression:

- Final Model with reduced AIC (Akaike information criterion is an estimator of out of sample prediction error) is selected.

```
Step:  AIC=99.07
status1 ~ spread2 + MDVP.Flo.Hz. + D2 + DFA + MDVP.Flo.Hz.:D2 +
    spread2:DFA

                       Df Deviance     AIC
<none>                      85.072  99.072
+ DFA:D2                1   83.915  99.915
+ spread2:D2            1   84.016 100.016
+ MDVP.Flo.Hz.:DFA      1   85.005 101.005
+ MDVP.Flo.Hz.:spread2  1   85.062 101.062
- MDVP.Flo.Hz.:D2       1   91.646 103.646
- spread2:DFA           1   95.767 107.767

Call:  glm(formula = status1 ~ spread2 + MDVP.Flo.Hz. + D2 + DFA + MDVP.Flo.Hz.:D2 +
    spread2:DFA, family = "binomial", data = interactiondata)

Coefficients:
    (Intercept)          spread2     MDVP.Flo.Hz.              D2             DFA  MDVP.Flo.Hz.:D2
         -9.216           24.894          -17.566          -1.793          16.485           39.340
    spread2:DFA
        -31.297

Degrees of Freedom: 151 Total (i.e. Null);  145 Residual
Null Deviance:      168.7
Residual Deviance: 85.07        AIC: 99.07
```

Handling Interaction Effects:

- Some variables add value to the model only on combination with other variables.
- **Step 1**: Creating all combinations of 2-way interactions.
- **Step 2**: Using all individual and combined variables in a step wise regression model.
- **Step 3:** Getting significant interaction variables based on p values.
- After these steps, following variables have the significant interactions
- ***MDVP.Flo.Hz* and *D2***
- ***Spread2* and *DFA***

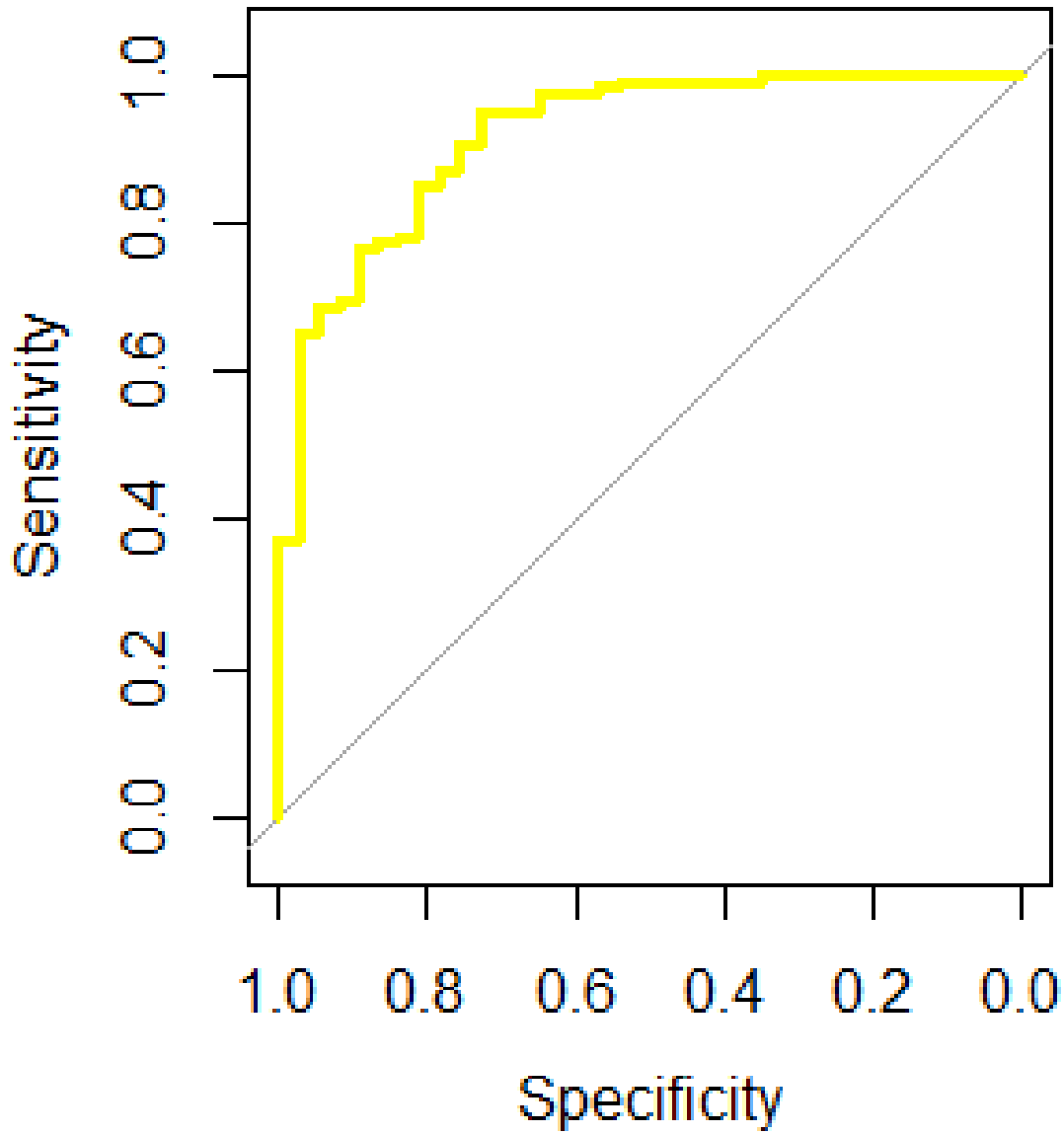
Building a Logistic Regression model:

```
##Building our final model with all significant variables and Significant variables using the train data

glmodel=glm(status ~ spread2 + MDVP.Flo.Hz. + D2 + DFA + MDVP.Flo.Hz.:D2 +
      spread2:DFA, family = "binomial", data = train)
pr=plogis(predict(glmodel,test))

pred=ifelse(pr>0.5,1,0)
library(caret)
confusionMatrix(as.factor(pred),test$status,positive = '1')
```

```
> confusionMatrix(as.factor(pred),test$status,positive = '1')
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0  2  0
         1  9 32

               Accuracy : 0.7907
                 95% CI : (0.6396, 0.8996)
    No Information Rate : 0.7442
    P-Value [Acc > NIR] : 0.307745

                  Kappa : 0.2485

 Mcnemar's Test P-Value : 0.007661

            Sensitivity : 1.0000
            Specificity : 0.1818
         Pos Pred Value : 0.7805
         Neg Pred Value : 1.0000
             Prevalence : 0.7442
         Detection Rate : 0.7442
   Detection Prevalence : 0.9535
      Balanced Accuracy : 0.5909

       'Positive' Class : 1
```

- ***Our model has high sensitivity which is what we highly desire but still specificity is very low.***
- ***We can address this problem by doing sampling technique to balance the data since Negative values of PD is way too low compared to Positive values.***
- ***We use over sampling to address this issue.***

ROC Curve for Logistic Regression:



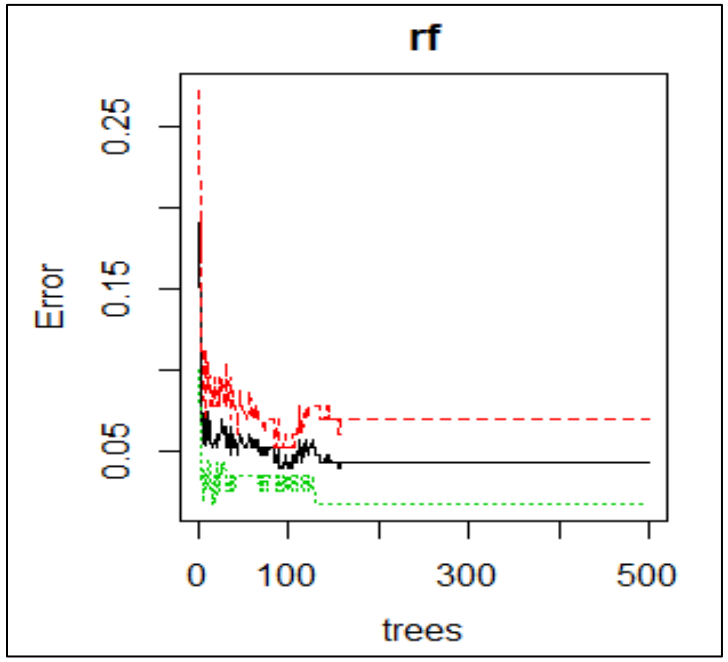
Oversampling:

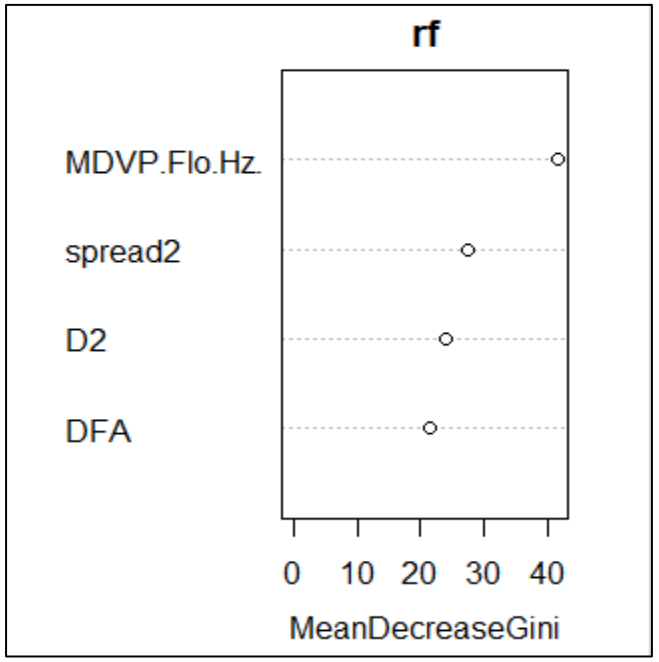
```
##To solve the low specificity value we do balancing techniques on the data since Negative values are very lesser compared to Positive values
##We do over sampling to scale the negative values to match with the Positive values
table(train$status)
library(ROSE)
over=ovun.sample(status~.,data=train,method='over',N=230)$data
table(over$status)
##Both 1 and 0 values are now equal to 115 values
```

Random Forest Model and plots over sampled data:

Random forests gave a descent accuracy along with Sensitivity and Specificity values

```
##Building a Random Forest Model
library(randomForest)

rf=randomForest(status ~ spread2 + MDVP.Flo.Hz. + D2 + DFA + MDVP.Flo.Hz.:D2 +
                spread2:DFA, data = over)
plot(rf)
varImpPlot(rf)
```






```
> pr=predict(rf,test)
> confusionMatrix(pr,test$status,positive = '1')
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0  7  1
         1  4 31

               Accuracy : 0.8837
                 95% CI : (0.7492, 0.9611)
    No Information Rate : 0.7442
    P-Value [Acc > NIR] : 0.02115

                  Kappa : 0.6646

 Mcnemar's Test P-Value : 0.37109

            Sensitivity : 0.9688
            Specificity : 0.6364
         Pos Pred Value : 0.8857
         Neg Pred Value : 0.8750
             Prevalence : 0.7442
         Detection Rate : 0.7209
   Detection Prevalence : 0.8140
      Balanced Accuracy : 0.8026

       'Positive' Class : 1
```

- From rf plot we could see that error reduction rate stabilizes after 200 hence we select 200 as best number of trees
- Variable importance plot signifies which variables has higher significance sorted in order.

Building an SVM Model:

Finally SVM radial kernel has the highest accuracy, sensitivity and specificity values.

```
> svmmodel=svm(status ~ spread2 + MDVP.Flo.Hz. + D2 + DFA + MDVP.Flo.Hz.:D2+spread2:DFA, data = over,probability=TRUE)
> summary(svmmodel)

Call:
svm(formula = status ~ spread2 + MDVP.Flo.Hz. + D2 + DFA + MDVP.Flo.Hz.:D2 + spread2:DFA, data = over,
    probability = TRUE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  118

 ( 60 58 )


Number of Classes:  2

Levels:
 1 0
```

```
> confusionMatrix(pr,test$status,positive = '1')
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0  8  1
         1  3 31

               Accuracy : 0.907
                 95% CI : (0.7786, 0.9741)
    No Information Rate : 0.7442
    P-Value [Acc > NIR] : 0.007125

                  Kappa : 0.7402

 Mcnemar's Test P-Value : 0.617075

            Sensitivity : 0.9688
            Specificity : 0.7273
         Pos Pred Value : 0.9118
         Neg Pred Value : 0.8889
             Prevalence : 0.7442
         Detection Rate : 0.7209
   Detection Prevalence : 0.7907
      Balanced Accuracy : 0.8480
```

Conclusion:

	Accuracy	Sensitivity	Specificity	F1-Score (2*Sensitivity*Specificity)/(Specificity+Specificity)
Logistic Regression	79.07	100	18.18	30.76
Random Forests	88.37	96.88	63.64	76.81
SVM (radial)	**90.7**	**96.88**	**72.73**	**83.08**

- We select SVM with radial kernel as a best model for our analysis which has higher accuracy and sensitivity when compared to other models
- F1-score acts as an important parameter in selecting models which takes into account both Sensitivity and Specificity.
- Hence selecting SVM radial as a best model for our analysis
- Future developments may be done to create a neural networks model which can give even better prediction values compared to SVM.