Build and deploy a parkinson prediction model using R

Ronak Fathi

2024-09-09

About Data Analysis Report

According to Oxford, Parkinson's Disease is a progressive disease of the central nervous system, and is marked by tremor, muscular rigidity, and slow, imprecise movement, chiefly affecting the middle-aged and elderly people.

It can last for years or even be lifelong. The complications of a person dealing with Parkinson's Disease include: thinking difficulties, emotional changes and depression, swallowing problems, chewing and eating problems, sleep disorders, bladder problems, constipation and may also prove fatal.

This RMarkdown file contains the report of the data analysis done for the project on building and deploying a parkinson prediction model in R. It contains analysis such as data exploration, summary statistics and building the prediction model. The final report was completed on Mon Sep 9 19:46:43 2024.

Data Description:

This data science project in R aims to predict the severity of Parkinson's disease based on the UCI Parkinsons dataset using machine learning algorithms. The dataset includes various features related to Parkinson's symptoms, and We have used Principal Component Analysis (PCA) for dimensionality reduction and other tools for attribute-correlation and Variable importance to aid in the efficient construction of the classification-based prediction system. Lastly, we have used random forest model with COREModel functionality to train and test our data.

Since RMSE Metric is not applicable for classification-based systems, therefore different metrics like **accuracy, precision etc.** to evaluate my prediction model in this case.

Features:
name - ASCII subject name and recording number
MDVP:Fo(Hz) - Average vocal fundamental frequency
MDVP:Fhi(Hz) - Maximum vocal fundamental frequency
MDVP:Flo(Hz) - Minimum vocal fundamental frequency
[- MDVP:Jitter(%)
- MDVP:Jitter(Abs)
- MDVP:RAP
- MDVP:PPQ
- jitter:DDP] : Several measures of variation in fundamental frequency

[- MDVP:Shimmer
- MDVP:Shimmer(dB)
- Shimmer:APQ3
- Shimmer:APQ5
- MDVP:APQ
- Shimmer:DDA] - Several measures of variation in amplitude

NHR,HNR - Two measures of ratio of noise to tonal components in the voice
status - Health status of the subject (one) - Parkinson's, (zero) - healthy
RPDE,D2 - Two nonlinear dynamical complexity measures
DFA - Signal fractal scaling exponent
spread1, spread2, PPE - Three nonlinear measures of fundamental frequency variation

Import data and data preprocessing

Load data and install packages

```r
#install.packages(" ")
library(data.table)
library(visdat)
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::between()     masks data.table::between()
## x dplyr::filter()      masks stats::filter()
## x dplyr::first()       masks data.table::first()
## x lubridate::hour()    masks data.table::hour()
## x lubridate::isoweek() masks data.table::isoweek()
## x dplyr::lag()         masks stats::lag()
## x dplyr::last()        masks data.table::last()
## x lubridate::mday()    masks data.table::mday()
## x lubridate::minute()  masks data.table::minute()
## x lubridate::month()   masks data.table::month()
## x lubridate::quarter() masks data.table::quarter()
## x lubridate::second()  masks data.table::second()
## x purrr::transpose()   masks data.table::transpose()
## x lubridate::wday()    masks data.table::wday()
## x lubridate::week()    masks data.table::week()
## x lubridate::yday()    masks data.table::yday()
## x lubridate::year()    masks data.table::year()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(moments)
library(dplyr)
library(ggcorrplot)
library(knitr)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
library(mlbench)
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
```

```
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
setwd("C:/Users/0&1/OneDrive/Documents/Parkinson")
data <- read.csv("PD_data.csv")
```

Exploratory Data Analysis

```
# about the dataset
dim(data) # dimension
```

```
## [1] 195  24
```

```
head(data) # content
```

```
##             name MDVP.Fo.Hz. MDVP.Fhi.Hz. MDVP.Flo.Hz. MDVP.Jitter...
## 1 phon_R01_S01_1     119.992      157.302       74.997        0.00784
## 2 phon_R01_S01_2     122.400      148.650      113.819        0.00968
## 3 phon_R01_S01_3     116.682      131.111      111.555        0.01050
## 4 phon_R01_S01_4     116.676      137.871      111.366        0.00997
## 5 phon_R01_S01_5     116.014      141.781      110.655        0.01284
## 6 phon_R01_S01_6     120.552      131.162      113.787        0.00968
##   MDVP.Jitter.Abs. MDVP.RAP MDVP.PPQ Jitter.DDP MDVP.Shimmer MDVP.Shimmer.dB.
## 1          0.00007  0.00370  0.00554    0.01109      0.04374            0.426
## 2          0.00008  0.00465  0.00696    0.01394      0.06134            0.626
## 3          0.00009  0.00544  0.00781    0.01633      0.05233            0.482
## 4          0.00009  0.00502  0.00698    0.01505      0.05492            0.517
## 5          0.00011  0.00655  0.00908    0.01966      0.06425            0.584
## 6          0.00008  0.00463  0.00750    0.01388      0.04701            0.456
##   Shimmer.APQ3 Shimmer.APQ5 MDVP.APQ Shimmer.DDA     NHR    HNR status     RPDE
## 1      0.02182      0.03130  0.02971     0.06545 0.02211 21.033      1 0.414783
## 2      0.03134      0.04518  0.04368     0.09403 0.01929 19.085      1 0.458359
## 3      0.02757      0.03858  0.03590     0.08270 0.01309 20.651      1 0.429895
## 4      0.02924      0.04005  0.03772     0.08771 0.01353 20.644      1 0.434969
## 5      0.03490      0.04825  0.04465     0.10470 0.01767 19.649      1 0.417356
## 6      0.02328      0.03526  0.03243     0.06985 0.01222 21.378      1 0.415564
##        DFA   spread1  spread2       D2      PPE
## 1 0.815285 -4.813031 0.266482 2.301442 0.284654
## 2 0.819521 -4.075192 0.335590 2.486855 0.368674
## 3 0.825288 -4.443179 0.311173 2.342259 0.332634
## 4 0.819235 -4.117501 0.334147 2.405554 0.368975
## 5 0.823484 -3.747787 0.234513 2.332180 0.410335
## 6 0.825069 -4.242867 0.299111 2.187560 0.357775
```

```
str(data) # structure
```

```
## 'data.frame':    195 obs. of  24 variables:
##  $ name            : chr  "phon_R01_S01_1" "phon_R01_S01_2" "phon_R01_S01_3" "phon_R01_S01_4" ...
##  $ MDVP.Fo.Hz.     : num  120 122 117 117 116 ...
##  $ MDVP.Fhi.Hz.    : num  157 149 131 138 142 ...
##  $ MDVP.Flo.Hz.    : num  75 114 112 111 111 ...
##  $ MDVP.Jitter...  : num  0.00784 0.00968 0.0105 0.00997 0.01284 ...
##  $ MDVP.Jitter.Abs.: num  0.00007 0.00008 0.00009 0.00009 0.00011 0.00008 0.00003 0.00003 0.00006 0.0
##  $ MDVP.RAP        : num  0.0037 0.00465 0.00544 0.00502 0.00655 0.00463 0.00155 0.00144 0.00293 0.00
##  $ MDVP.PPQ        : num  0.00554 0.00696 0.00781 0.00698 0.00908 0.0075 0.00202 0.00182 0.00332 0.00
##  $ Jitter.DDP      : num  0.0111 0.0139 0.0163 0.015 0.0197 ...
##  $ MDVP.Shimmer    : num  0.0437 0.0613 0.0523 0.0549 0.0643 ...
##  $ MDVP.Shimmer.dB.: num  0.426 0.626 0.482 0.517 0.584 0.456 0.14 0.134 0.191 0.255 ...
##  $ Shimmer.APQ3    : num  0.0218 0.0313 0.0276 0.0292 0.0349 ...
##  $ Shimmer.APQ5    : num  0.0313 0.0452 0.0386 0.0401 0.0483 ...
##  $ MDVP.APQ        : num  0.0297 0.0437 0.0359 0.0377 0.0447 ...
##  $ Shimmer.DDA     : num  0.0654 0.094 0.0827 0.0877 0.1047 ...
##  $ NHR             : num  0.0221 0.0193 0.0131 0.0135 0.0177 ...
##  $ HNR             : num  21 19.1 20.7 20.6 19.6 ...
##  $ status          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ RPDE            : num  0.415 0.458 0.43 0.435 0.417 ...
##  $ DFA             : num  0.815 0.82 0.825 0.819 0.823 ...
##  $ spread1         : num  -4.81 -4.08 -4.44 -4.12 -3.75 ...
##  $ spread2         : num  0.266 0.336 0.311 0.334 0.235 ...
##  $ D2              : num  2.3 2.49 2.34 2.41 2.33 ...
##  $ PPE             : num  0.285 0.369 0.333 0.369 0.41 ...
```

```
summary(data) # summary
```

```
##      name            MDVP.Fo.Hz.      MDVP.Fhi.Hz.    MDVP.Flo.Hz.
##  Length:195          Min.   : 88.33   Min.   :102.1   Min.   : 65.48
##  Class :character   1st Qu.:117.57   1st Qu.:134.9   1st Qu.: 84.29
##  Mode  :character   Median :148.79   Median :175.8   Median :104.31
##                     Mean   :154.23   Mean   :197.1   Mean   :116.32
##                     3rd Qu.:182.77   3rd Qu.:224.2   3rd Qu.:140.02
##                     Max.   :260.11   Max.   :592.0   Max.   :239.17
##  MDVP.Jitter...     MDVP.Jitter.Abs.       MDVP.RAP           MDVP.PPQ
##  Min.   :0.001680   Min.   :7.000e-06   Min.   :0.000680   Min.   :0.000920
##  1st Qu.:0.003460   1st Qu.:2.000e-05   1st Qu.:0.001660   1st Qu.:0.001860
##  Median :0.004940   Median :3.000e-05   Median :0.002500   Median :0.002690
##  Mean   :0.006220   Mean   :4.396e-05   Mean   :0.003306   Mean   :0.003446
##  3rd Qu.:0.007365   3rd Qu.:6.000e-05   3rd Qu.:0.003835   3rd Qu.:0.003955
##  Max.   :0.033160   Max.   :2.600e-04   Max.   :0.021440   Max.   :0.019580
##    Jitter.DDP        MDVP.Shimmer     MDVP.Shimmer.dB.  Shimmer.APQ3
##  Min.   :0.002040   Min.   :0.00954   Min.   :0.0850   Min.   :0.004550
##  1st Qu.:0.004985   1st Qu.:0.01650   1st Qu.:0.1485   1st Qu.:0.008245
##  Median :0.007490   Median :0.02297   Median :0.2210   Median :0.012790
##  Mean   :0.009920   Mean   :0.02971   Mean   :0.2823   Mean   :0.015664
##  3rd Qu.:0.011505   3rd Qu.:0.03789   3rd Qu.:0.3500   3rd Qu.:0.020265
##  Max.   :0.064330   Max.   :0.11908   Max.   :1.3020   Max.   :0.056470
##   Shimmer.APQ5        MDVP.APQ        Shimmer.DDA           NHR
##  Min.   :0.00570   Min.   :0.00719   Min.   :0.01364   Min.   :0.000650
##  1st Qu.:0.00958   1st Qu.:0.01308   1st Qu.:0.02474   1st Qu.:0.005925
##  Median :0.01347   Median :0.01826   Median :0.03836   Median :0.011660
##  Mean   :0.01788   Mean   :0.02408   Mean   :0.04699   Mean   :0.024847
##  3rd Qu.:0.02238   3rd Qu.:0.02940   3rd Qu.:0.06080   3rd Qu.:0.025640
##  Max.   :0.07940   Max.   :0.13778   Max.   :0.16942   Max.   :0.314820
##       HNR             status            RPDE             DFA
##  Min.   : 8.441   Min.   :0.0000   Min.   :0.2566   Min.   :0.5743
##  1st Qu.:19.198   1st Qu.:1.0000   1st Qu.:0.4213   1st Qu.:0.6748
##  Median :22.085   Median :1.0000   Median :0.4960   Median :0.7223
##  Mean   :21.886   Mean   :0.7538   Mean   :0.4985   Mean   :0.7181
##  3rd Qu.:25.076   3rd Qu.:1.0000   3rd Qu.:0.5876   3rd Qu.:0.7619
##  Max.   :33.047   Max.   :1.0000   Max.   :0.6852   Max.   :0.8253  
##     spread1          spread2                D2             PPE         
##  Min.   :-7.965   Min.   :0.006274   Min.   :1.423   Min.   :0.04454  
##  1st Qu.:-6.450   1st Qu.:0.174350   1st Qu.:2.099   1st Qu.:0.13745  
##  Median :-5.721   Median :0.218885   Median :2.362   Median :0.19405  
##  Mean   :-5.684   Mean   :0.226510   Mean   :2.382   Mean   :0.20655  
##  3rd Qu.:-5.046   3rd Qu.:0.279234   3rd Qu.:2.636   3rd Qu.:0.25298  
##  Max.   :-2.434   Max.   :0.450493   Max.   :3.671   Max.   :0.52737  
```

```r
# Check for missing values
library(naniar)

miss_scan_count(data = data, search = list("N/A","Unknown","Other"))
```

```
## # A tibble: 24 x 2
##    Variable             n
##    <chr>            <int>
##  1 name                 0
##  2 MDVP.Fo.Hz.          0
##  3 MDVP.Fhi.Hz.         0
##  4 MDVP.Flo.Hz.         0
##  5 MDVP.Jitter...       0
##  6 MDVP.Jitter.Abs.     0
##  7 MDVP.RAP             0
##  8 MDVP.PPQ             0
##  9 Jitter.DDP           0
## 10 MDVP.Shimmer         0
## # i 14 more rows
```

```r
#about variables
## check unique values of Status variable
#checking entries with status 0 and status 1

#checking only 'status' column
#using a new variable called 'status_val'
status_val<-data[,c("status")]
print(status_val)
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1
##  [38] 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1
##  [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0
## [186] 0 0 0 0 0 0 0 0 0 0
```

```r
#number of entries with status = 0 i.e. Healthy People
sum(status_val==0) #48
```

```
## [1] 48
```

```r
#number of entries with status = 1 i.e. People with Parkinson's Disease
sum(status_val==1) #147
```

```
## [1] 147
```

Observations:

Upon initial analysis of the Parkinson's Disease Dataset we see:
1. There are no null values in the Parkinson's Dataset
2. All the record inputs in the dataset are unique.
3. There are 48 healthy people and 147 patients with Parkinson's Disease; a total of 195 entries (as shown in the figure below).

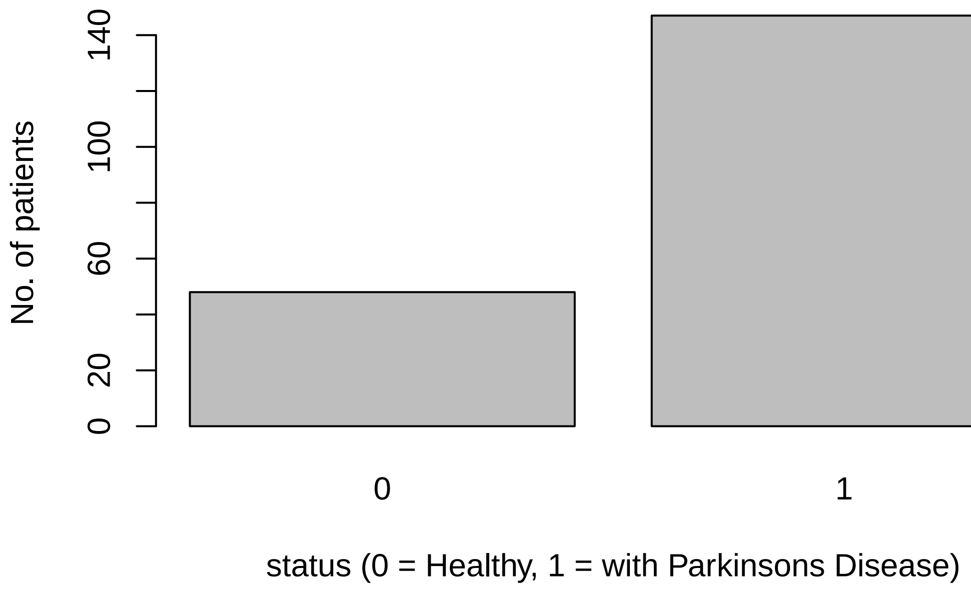


Figure 1: Barplot of Patient Healthy to Patient ratio

Main Parkinsons Data Analysis

This section includes the different techniques performed to analyze the Parkinson's Data. These techniques include:
1. Correlation
2. Understanding Variable Importance
3. Principal Component Analysis

Checking for repeated object values in the column "name" in data; redundancy

```
record_name <- data[,c("name")]
uniq_record_name <- unique(record_name)
length(uniq_record_name)
```

```
## [1] 195
```

Therefore, all the objects in column "name" (i.e. people tested for Parkinson's) and their observations for parkinson's are unique.

Checking correlation

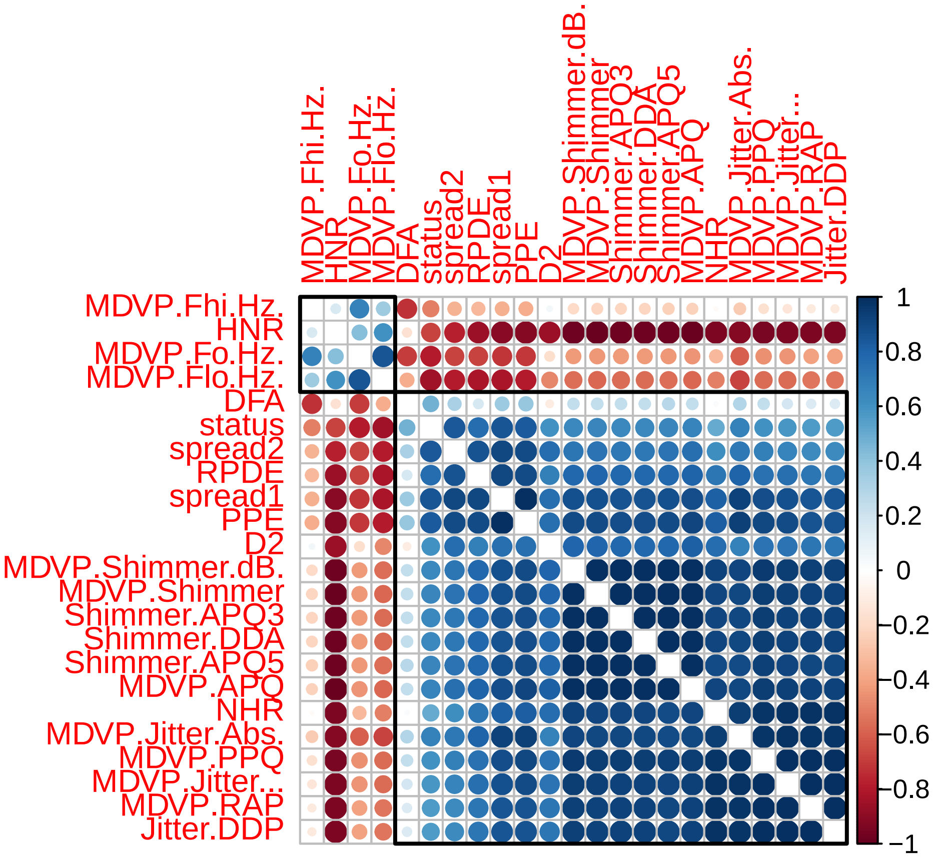
```
#removing the name attribute for correlation
data1 <- data[c(2:24)]

colnames(data1)
```

```
##  [1] "MDVP.Fo.Hz."      "MDVP.Fhi.Hz."     "MDVP.Flo.Hz."     "MDVP.Jitter..."
##  [5] "MDVP.Jitter.Abs." "MDVP.RAP"         "MDVP.PPQ"         "Jitter.DDP"
##  [9] "MDVP.Shimmer"     "MDVP.Shimmer.dB." "Shimmer.APQ3"     "Shimmer.APQ5"
## [13] "MDVP.APQ"         "Shimmer.DDA"      "NHR"              "HNR"
## [17] "status"           "RPDE"             "DFA"              "spread1"
## [21] "spread2"          "D2"               "PPE"
```

```
#creating correlation data
data2 <- transform(data1, status = as.numeric(status))
cor_data <- cor(data2, method = c("pearson"))
```

```
#creating correlation matrix
cor_matrix <- round(cor(cor_data),2)
```

```
corrplot::corrplot(cor(cor_data), order="hclust", addrect=2, diag=F)
```



```
#printing attrbutes that are highly correlated with a cutoff of 0.9
highlyCorrelated <- findCorrelation(cor_matrix, cutoff=0.9)
print(highlyCorrelated)
```

```
##  [1] 23 20 13 16  9  5 10 11 14 12  7  4  6  8
```

```
#The highly correlated attribute no.s are: 23 20  13  16  9 5  10  11 14  12 7 4 6 8
```

To understand highly correlated features easily, we used the function ‘findCorrelation()’ to find correlation from our already created correlation matrix with a cut-off of 0.9 and printing those attribute/column values as below:

i.e., PPE, spread 1, MDVP.APQ, HNR, MDVP.Shimmer,MDVP.Jitter.Abs.,MDVP.Shimmer.dB., Shimmer APQ3, Shimmer DDA, Shimmer APQ5, MDVP.PPQ, MDVP.Jitter…, MDVP.RAP, Jitter.DDP.

Understanding the importance of variables(feature selection)

We calculate the importance of variables in predicting the patient status in the Parkinson’s Dataset. This is done by creating a Feature Model using a classifier and specifying the dependent viariable and the data to be used.This Feature Model is then fed to the ‘varImp()’ function to find the importance of the variables. We can also view the plot of variable importance using the ‘varImpPlot()’ function. The importance of variables according to dependent attribute ‘status’ in Parkinson’s Disease Dataset can be shown in the plot given below:

```
#converting list "data1" to data frame
data3 <- as.data.frame(data1)

#fitting a random forest model
if(!require(randomForest)) install.packages("randomForest",repos = "http://cran.us.r-project.org")
```

```
## Loading required package: randomForest

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

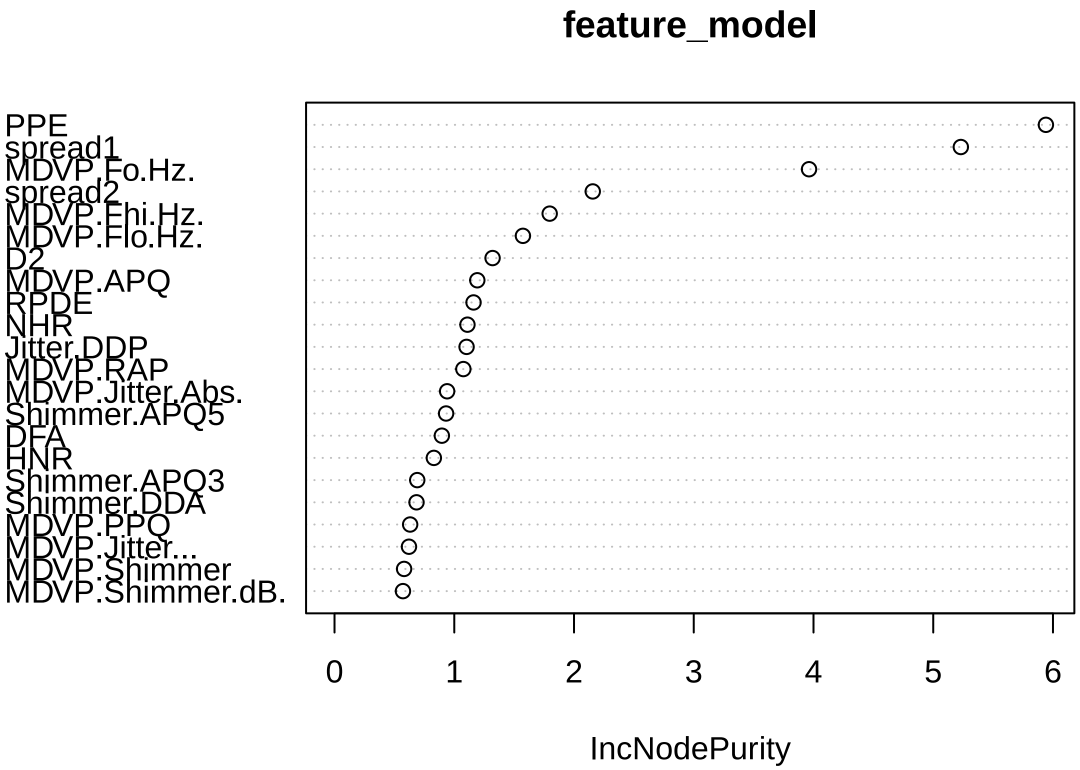
```
library(randomForest)
feature_model = randomForest(data$status~., data3)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values.  Are you sure you want to do regression?
```

```
#estimate variable importance
importance <- varImp(feature_model)

#summarize importance
print(importance)
```

```
##                    Overall
## MDVP.Fo.Hz.      3.9627743
## MDVP.Fhi.Hz.     1.7970006
## MDVP.Flo.Hz.     1.5732547
## MDVP.Jitter...   0.6216449
## MDVP.Jitter.Abs. 0.9405031
## MDVP.RAP          1.0758246
## MDVP.PPQ          0.6315170
## Jitter.DDP        1.1027265
## MDVP.Shimmer      0.5808849
## MDVP.Shimmer.dB.  0.5714747
## Shimmer.APQ3      0.6907165
## Shimmer.APQ5      0.9316959
## MDVP.APQ          1.1923255
## Shimmer.DDA       0.6847282
## NHR               1.1101542
## HNR               0.8295209
## RPDE              1.1609327
## DFA               0.8963282
## spread1           5.2304288
## spread2           2.1564757
## D2                1.3199506
## PPE               5.9409169
```

```
#plot importance
varImpPlot(feature_model)
```



Hence, the top 3 attribute features are: PPE, spread 1, MDVP.Fo.Hz

But other features in this data also play important roles in some way. Therefore, we use PCA to check it out.

Principal Component Analysis

Principle Component Analysis (PCA) is a mathematical procedure that transforms a number of (possibly) correlated variables into a smaller number of uncorrelated variables called **Principal Components**.
It is a method of analysis which involves finding the linear combination of a set of variables that has maximum variance and removing its effect, repeating this successively.

PCA is defined as an 'orthogonal linear transformation' that transforms the data to a new coordinate system such that the greatest variance by some scalar projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.

Applying PCA on Parkinson's Disease Dataset Here we apply PCA on Parkinson's Disease Dataset by ensuring that the data is centered and scaled.

The summary of the Principal Component Analysis done on the dataset is shown below:

```
#install.packages('*factoextra', dependencies = TRUE)

#installing packages to apply PCA in Parkinson's Dataset
if(!require(factoextra)) install.packages("factoextra",repos="http://cran.us.r-project.org", dependenci
```

```
## Loading required package: factoextra
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(factoextra)
if(!require(FactoMineR)) install.packages("FactoMineR",repos="http://cran.us.r-project.org", dependenci
```

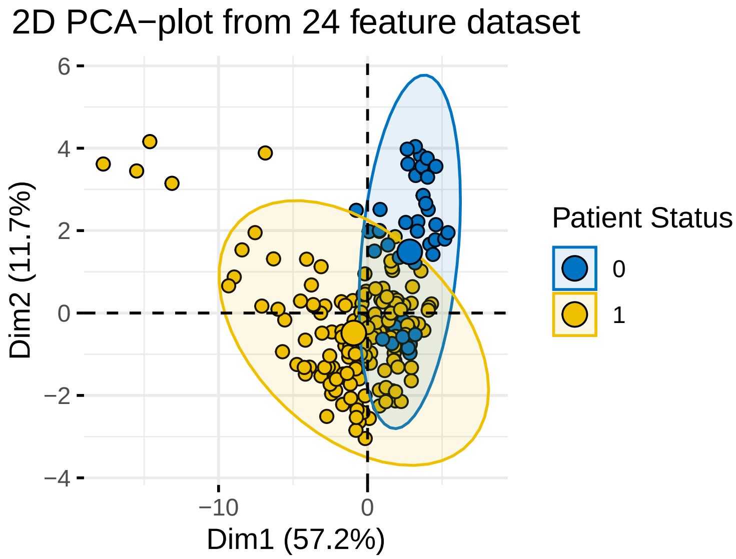
```
## Loading required package: FactoMineR
```

```
library(FactoMineR)
```

```
#Doing Principle Component Analysis on the Dataset
pd.pca <- prcomp(data2, center = TRUE, scale = TRUE)
summary(pd.pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.6256 1.6410 1.25590 1.21260 1.00533 0.85649 0.80032
## Proportion of Variance 0.5715 0.1171 0.06858 0.06393 0.04394 0.03189 0.02785
## Cumulative Proportion  0.5715 0.6886 0.75719 0.82113 0.86507 0.89696 0.92481
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.66946 0.59816 0.53667 0.47149 0.37331 0.32377 0.26406
## Proportion of Variance 0.01949 0.01556 0.01252 0.00967 0.00606 0.00456 0.00303
## Cumulative Proportion  0.94430 0.95985 0.97238 0.98204 0.98810 0.99266 0.99569
##                           PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     0.18947 0.14777 0.13253 0.11150 0.08288 0.05868 0.03288
## Proportion of Variance 0.00156 0.00095 0.00076 0.00054 0.00030 0.00015 0.00005
## Cumulative Proportion  0.99725 0.99820 0.99896 0.99950 0.99980 0.99995 1.00000
##                             PC22     PC23
## Standard deviation     0.0006015 0.000182
## Proportion of Variance 0.0000000 0.000000
## Cumulative Proportion  1.0000000 1.000000
```

The 2D-Plot for PCA on a 23 feature dataset is shown below:



Obtaining the eigenvalues, variance percentage and cumulative variance percentage for different dimensions or principal components:

```
##          eigenvalue variance.percent cumulative.variance.percent
## Dim.1  1.314527e+01     5.715333e+01                    57.15333
## Dim.2  2.692943e+00     1.170845e+01                    68.86178
## Dim.3  1.577273e+00     6.857709e+00                    75.71949
## Dim.4  1.470409e+00     6.393083e+00                    82.11257
## Dim.5  1.010689e+00     4.394301e+00                    86.50687
## Dim.6  7.335692e-01     3.189431e+00                    89.69631
## Dim.7  6.405124e-01     2.784837e+00                    92.48114
## Dim.8  4.481805e-01     1.948611e+00                    94.42975
## Dim.9  3.577979e-01     1.555643e+00                    95.98540
## Dim.10 2.880117e-01     1.252225e+00                    97.23762
## Dim.11 2.223062e-01     9.665486e-01                    98.20417
## Dim.12 1.393597e-01     6.059116e-01                    98.81008
## Dim.13 1.048291e-01     4.557785e-01                    99.26586
## Dim.14 6.972919e-02     3.031704e-01                    99.56903
## Dim.15 3.589816e-02     1.560790e-01                    99.72511
## Dim.16 2.183532e-02     9.493616e-02                    99.82004
## Dim.17 1.756358e-02     7.636340e-02                    99.89641
## Dim.18 1.243327e-02     5.405769e-02                    99.95047
## Dim.19 6.868404e-03     2.986262e-02                    99.98033
## Dim.20 3.443165e-03     1.497028e-02                    99.99530
## Dim.21 1.080936e-03     4.699721e-03                   100.00000
## Dim.22 3.618178e-07     1.573121e-06                   100.00000
## Dim.23 3.312204e-08     1.440088e-07                   100.00000
```

Plotting cos2 of variables to first 3 dimensions/PCs

Checking Quality of Representation of Variables in PCs on the factor map:

The cos2 of Variables to both the dimensions show the following:

1. A high cos2 indicates a good representation of the variable on the Principal Component. In this case, the variable is positioned close to the circumference of the correlation circle.

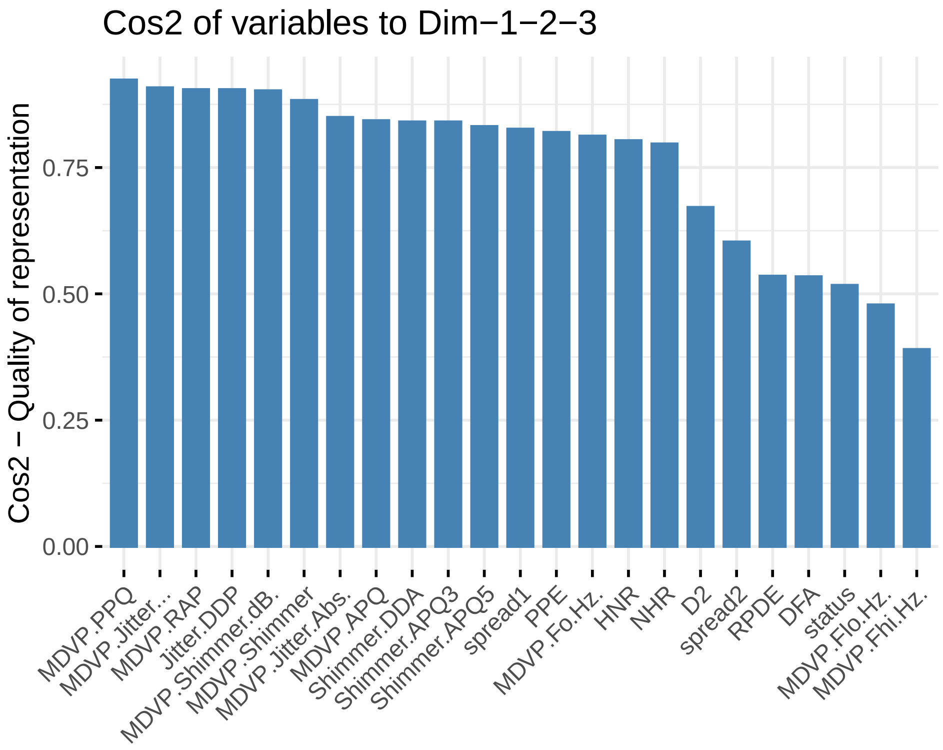


Figure 2: cos2 QoR of Variables in first 3 PCs

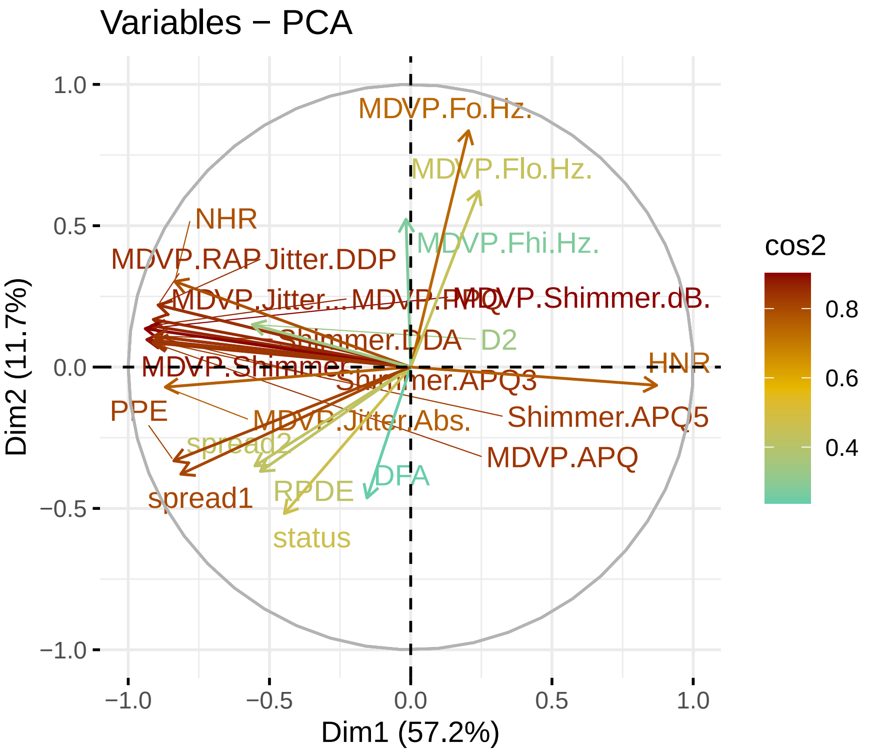


Figure 3: Variable QoR in Factor Map

2. A low cos2 value indicates that the variable is not perfectly represented by the PCs. In this case, the variable is close to the centre of the correlation circle.
   Hence, the variable with high cos2 value is more important for interpretation in the multivariate data.

Build the prediction model

In order to predict the people in 2 categories i.e., 0 for healthy and 1 for patients with Parkinson's Disease, our classification model utlizes **Random Forest Classifier** of the **COREIearn Package** to accurately predict the validation/test data after the model has been trained with 70% of the dataset in random fashion.

Here, we have trained our model against the attribute 'status' (dependent variable) with 136 inputs of our training data using **CoreModel** for **Random Forest Classifier** and then tested our model with 45 inputs of the test/validation data to obtain our results.

Random Forest

Data Preprocessing

Comparison of Real and Predicted counts for patient status:

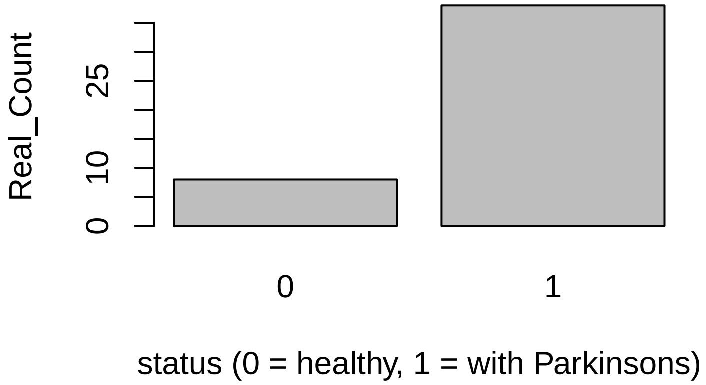


Figure 4: Real Count of Patient Status

Classification Evaluation Metrics

There are different classification evaluation metrics to evaluate classification models like Acuuracy, Precision, Recall, F1 score, etc.
Here, we have used the 'modelEval()' function from the COREIearn package to evaluate the classification-based prediction system.

The evaluation of classification-based prediction system is as shown below:

i. Prediction Matrix (confusion matrix)

```
##    0  1
## 0 6  2
```

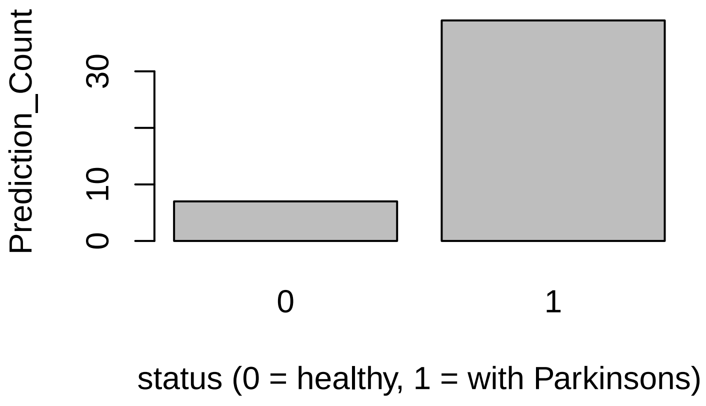


Figure 5: Predicted Count of Patient Status

```
## 1 1 37
```

ii. Accuracy

```
## [1] 0.9347826
```

iii. AUC

```
## [1] 0.9703947
```

iv. Recall

```
## [1] 0.75
```

v. Precision

```
## [1] 0.8571429
```

vi. F1 Score

```
## [1] 0.8
```

Findings and Conclusions

Precision and recall are indeed critical metrics in medical diagnosis, as false positive and false negative predictions can have serious consequences. In the context of Parkinson prediction, it is important to accurately identify Parkinson cases to ensure appropriate interventions and timely treatment.

First, we did exploratory data analysis and discovered that PPE, spread 1, MDVP.APQ, HNR, MDVP.Shimmer,MDVP.Jitter.Abs.,MDVP.Shimmer.dB., Shimmer APQ3, Shimmer DDA, Shimmer APQ5, MDVP.PPQ, MDVP.Jitter…, MDVP.RAP, Jitter.DDP. are highly correlated features.

The top 3 attribute features are: PPE, spread 1, MDVP.Fo.Hz. But other features in this data also play important roles in some way. Therefore, we used PCA to check it out.

We applied PCA on Parkinson's Disease Dataset by ensuring that the data is centered and scaled. The cos2 of Variables to both the dimensions show the following:

1. A high cos2 indicates a good representation of the variable on the Principal Component. In this case, the variable is positioned close to the circumference of the correlation circle.

2. A low cos2 value indicates that the variable is not perfectly represented by the PCs. In this case, the variable is close to the centre of the correlation circle.
   Hence, the variable with high cos2 value is more important for interpretation in the multivariate data.

In order to predict the people in 2 categories i.e., 0 for healthy and 1 for patients with Parkinson's Disease, our classification model utlizes **Random Forest Classifier** of the **COREelearn Package** to accurately predict the validation/test data after the model has been trained with 70% of the dataset in random fashion.

We have trained our model against the attribute 'status' (dependent variable) with 136 inputs of our training data using **CoreModel** for **Random Forest Classifier** and then tested our model with 45 inputs of the test/validation data to obtain our results.The variable with high cos2 value is more important for interpretation in the multivariate data.

The results of the models in terms of precision, recall, F1-score, indicate that they faced challenges in correctly identifying stroke cases. This can be attributed to the significant class imbalance between non-stroke and stroke instances in the test set, with a much larger number of non-stroke instances compared to stroke instances. This class imbalance creates a bias in the models towards predicting the majority class, which in this case is non-stroke.

The Random Forest model has a high recall of 0.7. This suggests that the model was successful in correctly identifying a large proportion of the actual cases with Parkinson in the dataset. Moreover, the high precision of 0.87 indicates that the model also classified a few number of non-parkinson cases as parkinson, resulting in a low rate of false positive predictions.

In conclusion, the results of this random forest model was successful and has implications for healthcare providers, as accurate prediction of Parkinson can help in early identification, and appropriate allocation of resources for controlling the disease.