DA 5030 Project

Puja Mehta

2020-04-19

Business Understanding

Background:

The dataset includes 30 patients with early untreated Parkinson's disease (PD), 50 patients with REM sleep behavior disorder (RBD), which are at high risk developing Parkinson's disease or other synucleinopathies; and 50 healthy controls (HC). All patients were scored clinically by a well-trained professional neurologist with experience in movement disorders. All subjects were examined during a single session with a speech specialist. All subjects performed reading of standardized, phonetically-balanced text of 80 words and monologue about their interests, job, family or current activities for approximately 90 seconds.

Project Plan:

The goal is to predict the disease categories based on different predictors in the dataset. Visualization of the data distribution, obtaining correlation between variables and detection of outliers.

The data would be split into training and test set for hold out validation and k-fold cross validation method would be used while training the data.

Data processing , variable tranformation, normalization, binning and variable selection would be the next steps. PCA analazyis would be carried out on the variables to analyze the variables.

I would include a minimum of 3 Machine Learning models, ensemble them and compare their performace. The models I choose will be classification models and hence the comparision would be based on Accuracy and Kappa value.

I also plan to include a model just to analyse the feature that are most important in identification of the disease category and which could then be used as biomarkers in Disease Prediction.

#Data Understanding

Collection of Initial data:

Original paper: Automated analysis of connected speech reveals early biomarkers of Parkinson's disease in patients with rapid eye movement sleep behaviour disorder by Jan Hlavnička, Roman Čmejla, Tereza Tykalová, Karel Šonka, Evžen Růžička & Jan Rusz @nature.com

```
# DATA ACQUISITION
library(readr)
datasetn <- read.csv("Downloads/early-biomarkers-of-parkinsons-disease/dataset.csv",
                     stringsAsFactors = F)

# Removing the redundant columns
dataset <- datasetn[,1:53]
# Renaming the columns
names(dataset)[1:53] <- c("Category","Age","Gender","History","OnsetAge",
```

```
                    "Duration","AntidepressantTherapy","medication",
                    "AntipsychoticMed","BenzodiazepineMed","LevodopaEquivalent",
                    "Clonazepam","MotorOverviewHY","MotorOverViewUPDRS3",
                    "Speech","FaceExpression","TremorHeadR","TremorRUER","TremorLUER",
                    "TermorRLER","TremorLLER","TremorRUEA","TremorLUEA",
                    "RigidityNeck","RigidityRUE","RigidityLUE","RigidityRLE","RigidityLLE",
                    "FTapsRUE","FtapsLUE","HMoveRUE","HMoveLUE","RAMoveRUE",
                    "RAMoveLUE","LegAgilityRLE","LegAgilityLLE","ArisingChair",
                    "Posture","Gait","PStability","BodyBradykinesiaHypokinesia",
                    "SpeechEntropyTime","RateofSpeech","AccSpeechTime","PauseDuration",
                    "VoicedIntervalDuration","Gapping",
                    "UnvoicedStopDuration","Decay","RelativeLoudness","PauseIntervalResp",
                    "RateofSpeechResp","LatencyRespExchange" )

# Obtaining the dimension of the dataset
dim(dataset)
```

```
## [1] 130  53
```

Data Description:

```
# The dataset contains features which could be used as biomarkers in
#the disease category
# I have named the columns which makes it self explanatory
# I will also be stressing the features which I obtain in the end of
#my analysis because it will be only those which have the most importance.
colnames(dataset)
```

```
##  [1] "Category"                    "Age"
##  [3] "Gender"                      "History"
##  [5] "OnsetAge"                    "Duration"
##  [7] "AntidepressantTherapy"       "medication"
##  [9] "AntipsychoticMed"            "BenzodiazepineMed"
## [11] "LevodopaEquivalent"          "Clonazepam"
## [13] "MotorOverviewHY"             "MotorOverViewUPDRS3"
## [15] "Speech"                      "FaceExpression"
## [17] "TremorHeadR"                 "TremorRUER"
## [19] "TremorLUER"                  "TermorRLER"
## [21] "TremorLLER"                  "TremorRUEA"
## [23] "TremorLUEA"                  "RigidityNeck"
## [25] "RigidityRUE"                 "RigidityLUE"
## [27] "RigidityRLE"                 "RigidityLLE"
## [29] "FTapsRUE"                    "FtapsLUE"
## [31] "HMoveRUE"                    "HMoveLUE"
## [33] "RAMoveRUE"                   "RAMoveLUE"
## [35] "LegAgilityRLE"               "LegAgilityLLE"
## [37] "ArisingChair"                "Posture"
## [39] "Gait"                        "PStability"
## [41] "BodyBradykinesiaHypokinesia" "SpeechEntropyTime"
## [43] "RateofSpeech"                "AccSpeechTime"
## [45] "PauseDuration"               "VoicedIntervalDuration"
## [47] "Gapping"                     "UnvoicedStopDuration"
## [49] "Decay"                       "RelativeLoudness"
```

```
## [51] "PauseIntervalResp"           "RateofSpeechResp"
## [53] "LatencyRespExchange"
```

Data Exploration:

```
head(dataset)
```

```
##   Category Age Gender History OnsetAge Duration AntidepressantTherapy
## 1     PD01  58      F      No       56        2                    No
## 2     PD02  68      F      No       67        1                    No
## 3     PD03  68      M      No       67        1                    No
## 4     PD04  75      M      No       73        2                    No
## 5     PD05  61      M     Yes       60      0.7                    No
## 6     PD06  58      M      No       58        1                    No
##   medication AntipsychoticMed BenzodiazepineMed LevodopaEquivalent Clonazepam
## 1         No               No                No                  0          0
## 2         No               No                No                  0          0
## 3         No               No                No                  0          0
## 4         No               No                No                  0          0
## 5         No               No                No                  0          0
## 6         No               No                No                  0          0
##   MotorOverviewHY MotorOverViewUPDRS3 Speech FaceExpression TremorHeadR
## 1             1.5                   8      0              1           0
## 2             2.5                  22      1              1           0
## 3               2                  19      0              2           0
## 4               2                  24      0              2           0
## 5             2.5                  54      1              3           3
## 6               2                  29      1              2           0
##   TremorRUER TremorLUER TermorRLER TremorLLER TremorRUEA TremorLUEA
## 1          0          2          0          2          0          0
## 2          0          0          0          0          1          1
## 3          0          0          0          0          0          0
## 4          1          0          1          0          1          1
## 5          2          1          1          0          1          2
## 6          0          0          0          0          1          1
##   RigidityNeck RigidityRUE RigidityLUE RigidityRLE RigidityLLE FTapsRUE
## 1            0           0           1           0           0        0
## 2            1           0           1           1           2        1
## 3            2           0           0           2           2        1
## 4            1           1           1           1           1        1
## 5            3           3           3           1           1        3
## 6            3           2           1           1           1        3
##   FtapsLUE HMoveRUE HMoveLUE RAMoveRUE RAMoveLUE LegAgilityRLE LegAgilityLLE
## 1        1        0        0         0         1             0             0
## 2        2        0        1         1         2             1             2
## 3        1        0        1         0         0             1             2
## 4        1        2        2         1         2             1             0
## 5        4        2        4         2         3             1             3
## 6        3        2        1         1         0             1             1
##   ArisingChair Posture Gait PStability BodyBradykinesiaHypokinesia
## 1            0       0    0          0                           0
## 2            0       1    0          1                           1
## 3            0       3    0          0                           2
```

```
## 4            0       1    1          0                          1
## 5            1       2    1          1                          2
## 6            0       1    1          0                          2
##   SpeechEntropyTime RateofSpeech AccSpeechTime PauseDuration
## 1             1.564          354          6.05           146
## 2             1.564          340         27.52           173
## 3             1.550          211         11.97           377
## 4             1.519          140         -2.49           360
## 5             1.543          269          6.72           211
## 6             1.553          317         24.19           186
##   VoicedIntervalDuration Gapping UnvoicedStopDuration  Decay RelativeLoudness
## 1                    264   58.65                31.38 -2.101           -22.47
## 2                    253   48.26                22.38 -1.745           -24.59
## 3                    322   47.54                38.12  2.657           -16.89
## 4                    663   13.72                44.88 -0.934           -25.54
## 5                    328   42.90                47.12 -0.973           -22.61
## 6                    286   43.83                33.63  0.921           -25.00
##   PauseIntervalResp RateofSpeechResp LatencyRespExchange
## 1              4.50            21.14                167
## 2              7.00            15.28                163
## 3              3.00            20.76                372
## 4              1.00            18.71                119
## 5              5.00            16.26                 78
## 6              2.75            27.07                124
```

```
tail(dataset)
```

```
##     Category Age Gender History OnsetAge Duration AntidepressantTherapy
## 125     HC45  46      M       -        -        -                    No
## 126     HC46  69      M       -        -        -                    No
## 127     HC47  68      M       -        -        -                    No
## 128     HC48  53      M       -        -        -                    No
## 129     HC49  44      M       -        -        -                    No
## 130     HC50  54      M       -        -        -                    No
##     medication AntipsychoticMed BenzodiazepineMed LevodopaEquivalent Clonazepam
## 125         No               No                No                  0          0
## 126         No               No                No                  0          0
## 127         No               No                No                  0          0
## 128         No               No                No                  0          0
## 129         No               No                No                  0          0
## 130         No               No                No                  0          0
##     MotorOverviewHY MotorOverViewUPDRS3 Speech FaceExpression TremorHeadR
## 125               -                   -      -              -          -
## 126               -                   -      -              -          -
## 127               -                   -      -              -          -
## 128               -                   -      -              -          -
## 129               -                   -      -              -          -
## 130               -                   -      -              -          -
##     TremorRUER TremorLUER TermorRLER TremorLLER TremorRUEA TremorLUEA
## 125          -          -          -          -          -          -
## 126          -          -          -          -          -          -
## 127          -          -          -          -          -          -
## 128          -          -          -          -          -          -
## 129          -          -          -          -          -          -
```

```
## 130          -          -          -          -          -          -
##     RigidityNeck RigidityRUE RigidityLUE RigidityRLE RigidityLLE FTapsRUE
## 125            -           -           -           -           -        -
## 126            -           -           -           -           -        -
## 127            -           -           -           -           -        -
## 128            -           -           -           -           -        -
## 129            -           -           -           -           -        -
## 130            -           -           -           -           -        -
##     FtapsLUE HMoveRUE HMoveLUE RAMoveRUE RAMoveLUE LegAgilityRLE LegAgilityLLE
## 125        -        -        -         -         -              -             -
## 126        -        -        -         -         -              -             -
## 127        -        -        -         -         -              -             -
## 128        -        -        -         -         -              -             -
## 129        -        -        -         -         -              -             -
## 130        -        -        -         -         -              -             -
##     ArisingChair Posture Gait PStability BodyBradykinesiaHypokinesia
## 125            -       -    -          -                          -
## 126            -       -    -          -                          -
## 127            -       -    -          -                          -
## 128            -       -    -          -                          -
## 129            -       -    -          -                          -
## 130            -       -    -          -                          -
##     SpeechEntropyTime RateofSpeech AccSpeechTime PauseDuration
## 125             1.530          457         17.62           125
## 126             1.564          265          3.58           198
## 127             1.547          291          6.31           183
## 128             1.540          298        -13.66           177
## 129             1.560          359         -2.44           169
## 130             1.552          264          6.49           171
##     VoicedIntervalDuration Gapping UnvoicedStopDuration  Decay RelativeLoudness
## 125                    197   44.38                20.13 -5.649           -16.49
## 126                    365   40.25                26.88 -1.872           -28.04
## 127                    359   39.59                31.37 -1.517           -22.87
## 128                    283   53.01                50.50 -1.111           -22.91
## 129                    256   50.68                17.88 -0.823           -23.82
## 130                    354   35.59                29.13 -0.469           -28.26
##     PauseIntervalResp RateofSpeechResp LatencyRespExchange
## 125              8.75            10.91                133
## 126              6.50            10.24                158
## 127              5.00            13.46                224
## 128              4.50            19.11                251
## 129              6.50            18.14                226
## 130              4.50            17.57                158
```

```
 # It is observed that the data contains missing values and it is
#mainly categorical features and few numeric features.
```

Data Prepartion

Cleaning and Formatting Data:

```r
# Converting the Disease categories into factors and making the
#categories numeric
dataset$Category <- as.factor(dataset$Category)
levels(dataset$Category)[1:50] <- "1" #HC
levels(dataset$Category)[2:31] <- "2" #PD
levels(dataset$Category)[3:52] <- "3" #RBD
```

```r
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
# Binning the Age feature
min(dataset$Age)
```

```
## [1] 34
```

```r
max(dataset$Age)
```

```
## [1] 83
```

```r
dataset$Age <- as.factor(findInterval(dataset$Age, c(20, 40, 60,80)))

# Converting the Gender feature into numeric categorical data
dataset$Gender <- as.factor(dataset$Gender)
dataset$Gender <- as.numeric(dataset$Gender)
# F =1
# M =2


# Assumption that I have considered for the missing data that there
#was no presence of history for the healthy individuals
dataset[81:130,4] <- "No"
dataset$History <- as.factor(dataset$History)
dataset$History <- as.numeric(dataset$History)

# Imputing the missing data with the mean of the existing data and
```

```
#binning it into numeric categorical data
dataset[81:130,5] <- NA
mean.age <- round(mean(as.numeric(dataset$OnsetAge), na.rm = T))
dataset$OnsetAge <- impute(dataset$OnsetAge,mean.age)
dataset$OnsetAge <- as.factor(findInterval(dataset$OnsetAge, c(30,40,50,60,70)))

# Imputing the missing data with the mean of the existing data and
#binning it into numeric categorical data
dataset[81:130,6] <- NA
mean.duration <- round(mean(as.numeric(dataset$Duration), na.rm = T))
dataset$Duration <- impute(dataset$Duration,mean.duration)
dataset$Duration <- as.factor(findInterval(dataset$Duration, c(0,10, 20)))

# Re-assigning the factor levels for a uniform data format
dataset$AntidepressantTherapy <- as.factor(dataset$AntidepressantTherapy)
levels(dataset$AntidepressantTherapy)[1:9]<- "1"
levels(dataset$AntidepressantTherapy)[2]<- "0"

# Re-assigning the factor levels for a uniform data format
dataset$BenzodiazepineMed <- as.factor(dataset$BenzodiazepineMed)
levels(dataset$BenzodiazepineMed)[1:3] <- "1"
levels(dataset$BenzodiazepineMed)[2] <- "0"

# Assumption that I have considered for the missing data that there
#was no presence of Speech issues for the healthy individuals
dataset[81:130,15] <- NA
dataset$Speech <- impute(dataset$Speech,0)

# Re-assigning the factor levels for a uniform data format
dataset$FaceExpression <- as.factor(dataset$FaceExpression)
levels(dataset$FaceExpression)[1:2] <- 0
```

Select Data:

There is a big chunk of data which is missing in the dataset in columns between 17 and 41. Since, removing the rows gives only one disease category which would not make sense. There wasnt a time series present so the values could be carried forward. The best way to handle this chuck of missing data was to remove it. Yes, there would be a loss of data but the data quality would not be hampered this way.

```
# Selecting columns which have no significant data
rm.col <- c(5,8,9,11,12,13,14,17:41)
# Removing the columns with no significant data
dataset <- dataset[-rm.col]
# Checking for N/A values
sum(is.na.data.frame(dataset))
```

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

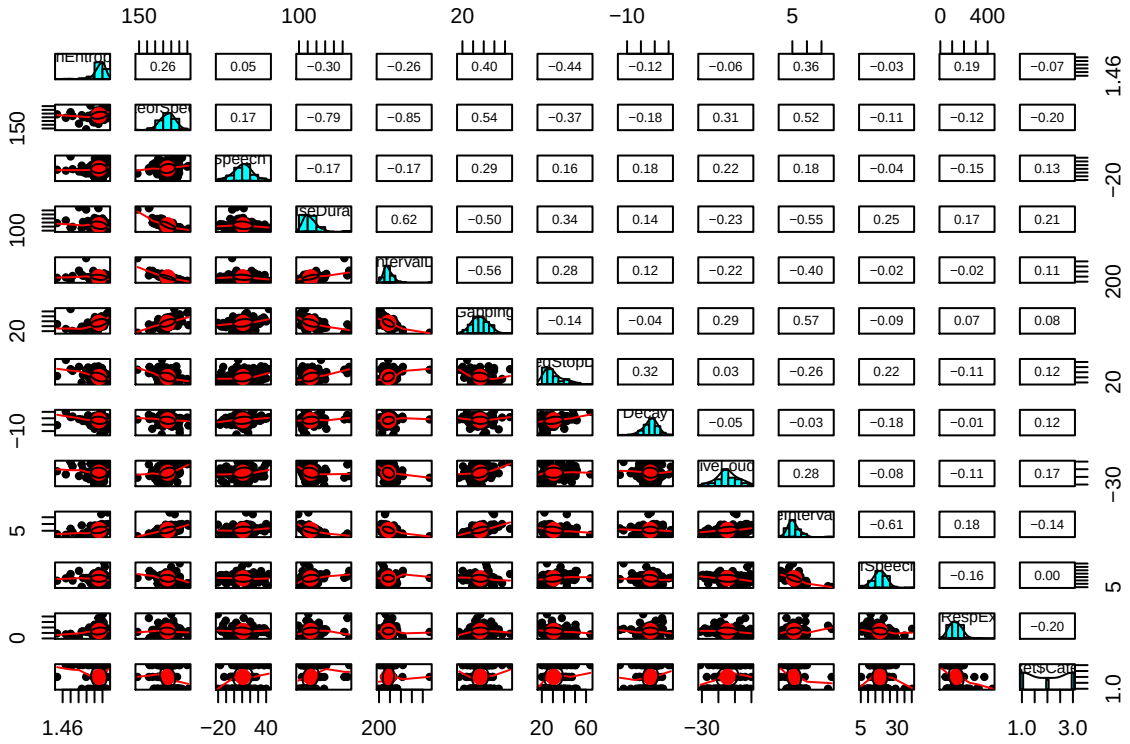
Integrating Data:

```
# Integrating categorical and numeric data in different sets
data.cat <- dataset[1:9]
data.num <- dataset[10:21]
```

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following object is masked from 'package:Hmisc':
##
##     describe

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
d.num <- cbind(data.num,dataset$Category)
# Analysing the data based on correlation and the distribution
pairs.panels(d.num)
```



It is observed that there is no high correlation observered in the numeric data set. The distribution is close to normal distribution but skewed in some features which wil be analysed in the later section

Normalizing the Data:

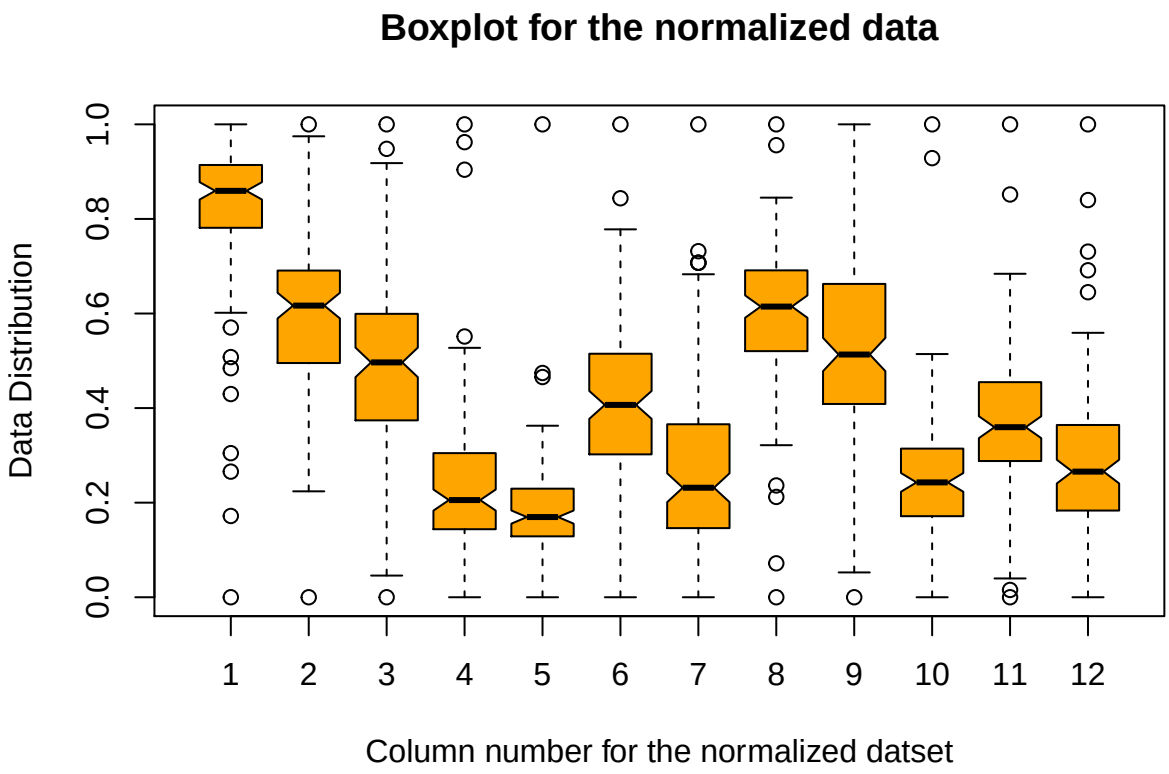
```
library(gplots)
```

```
##
## Attaching package: 'gplots'

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

```
library(taRifx)
library(Hmisc)
library(psych)

# Min-max Normalization will be considered
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x))) }
# Since the categorical data cannot be normalised only the numeric
#dataset is normalized
data.n <- as.data.frame(lapply(data.num, normalize))
summary(data.n)
```

```
##  SpeechEntropyTime  RateofSpeech    AccSpeechTime    PauseDuration
##  Min.   :0.0000    Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.7812    1st Qu.:0.4961   1st Qu.:0.3743   1st Qu.:0.1447
##  Median :0.8594    Median :0.6167   Median :0.4966   Median :0.2055
##  Mean   :0.8216    Mean   :0.5908   Mean   :0.4907   Mean   :0.2419
##  3rd Qu.:0.9141    3rd Qu.:0.6901   3rd Qu.:0.5993   3rd Qu.:0.3048
##  Max.   :1.0000    Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  VoicedIntervalDuration    Gapping       UnvoicedStopDuration     Decay
##  Min.   :0.0000         Min.   :0.0000   Min.   :0.0000       Min.   :0.0000
##  1st Qu.:0.1288         1st Qu.:0.3027   1st Qu.:0.1462       1st Qu.:0.5220
##  Median :0.1695         Median :0.4067   Median :0.2316       Median :0.6147
##  Mean   :0.1905         Mean   :0.4129   Mean   :0.2761       Mean   :0.5967
##  3rd Qu.:0.2296         3rd Qu.:0.5139   3rd Qu.:0.3658       3rd Qu.:0.6904
##  Max.   :1.0000         Max.   :1.0000   Max.   :1.0000       Max.   :1.0000
##  RelativeLoudness PauseIntervalResp RateofSpeechResp LatencyRespExchange
##  Min.   :0.0000   Min.   :0.0000    Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.4098   1st Qu.:0.1714    1st Qu.:0.2882   1st Qu.:0.1833
##  Median :0.5133   Median :0.2429    Median :0.3596   Median :0.2657
##  Mean   :0.5324   Mean   :0.2612    Mean   :0.3678   Mean   :0.2892
##  3rd Qu.:0.6614   3rd Qu.:0.3143    3rd Qu.:0.4542   3rd Qu.:0.3643
##  Max.   :1.0000   Max.   :1.0000    Max.   :1.0000   Max.   :1.0000
```

```
boxplot(data.n[1:12], col = "orange", notch = T , horizontal = F,
        names = c(1:12),
        xlab= "Column number for the normalized datset",
        ylab = "Data Distribution",
        main = "Boxplot for the normalized data")
```



It is observed that there are few outliers that are present. Since, there are not many outliers, I choose to not eliminate them. The data is mainly normally distibuted. When I try to transform the data with square-root, log or inverse transform, there is not much improvement in the overall distribution of the data. Hence, there is no transformation carried out.

Shaping the Data:

```
# Obtaining the mergered, cleaned and formatted dataset for the models
newdata <- cbind(data.cat,data.n)
newdata <- as.data.frame(sapply(remove.factors(newdata), as.numeric))
str(newdata)
```

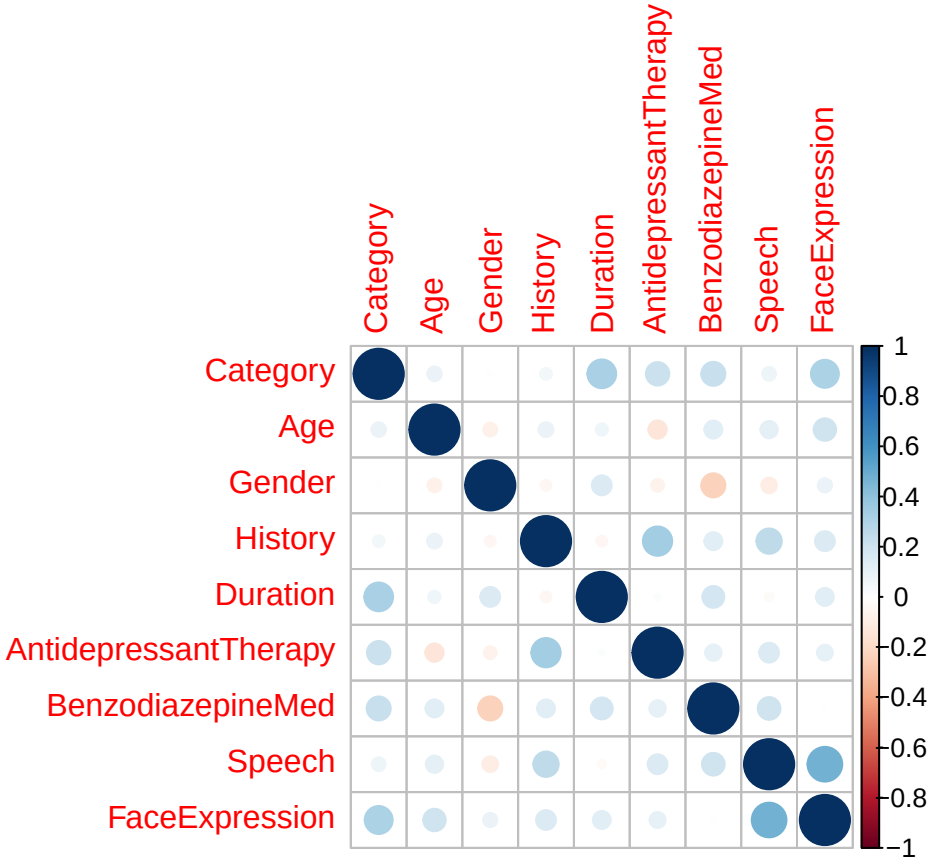
```
## 'data.frame':    130 obs. of  21 variables:
##  $ Category              : num  2 2 2 2 2 2 2 2 2 2 ...
##  $ Age                   : num  2 3 3 3 3 2 3 2 3 3 ...
##  $ Gender                : num  1 1 2 2 2 2 2 1 2 2 ...
##  $ History               : num  1 1 1 1 2 1 1 1 1 1 ...
##  $ Duration              : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ AntidepressantTherapy : num  0 0 0 0 0 0 0 1 0 1 ...
##  $ BenzodiazepineMed     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Speech                : num  0 1 0 0 1 1 0 1 0 1 ...
##  $ FaceExpression        : num  1 1 2 2 3 2 0 2 0 1 ...
##  $ SpeechEntropyTime     : num  0.922 0.922 0.812 0.57 0.758 ...
##  $ RateofSpeech          : num  0.675 0.631 0.224 0 0.407 ...
##  $ AccSpeechTime         : num  0.414 0.751 0.507 0.28 0.425 ...
##  $ PauseDuration         : num  0.171 0.264 0.962 0.904 0.394 ...
##  $ VoicedIntervalDuration: num  0.144 0.12 0.268 1 0.281 ...
```

```
##  $ Gapping                : num  0.469 0.361 0.353 0 0.305 ...
##  $ UnvoicedStopDuration   : num  0.2927 0.0976 0.4389 0.5854 0.634 ...
##  $ Decay                  : num  0.57 0.589 0.828 0.633 0.631 ...
##  $ RelativeLoudness       : num  0.519 0.381 0.88 0.32 0.51 ...
##  $ PauseIntervalResp      : num  0.2 0.343 0.114 0 0.229 ...
##  $ RateofSpeechResp       : num  0.455 0.287 0.444 0.385 0.315 ...
##  $ LatencyRespExchange    : num  0.364 0.355 0.84 0.253 0.158 ...
```

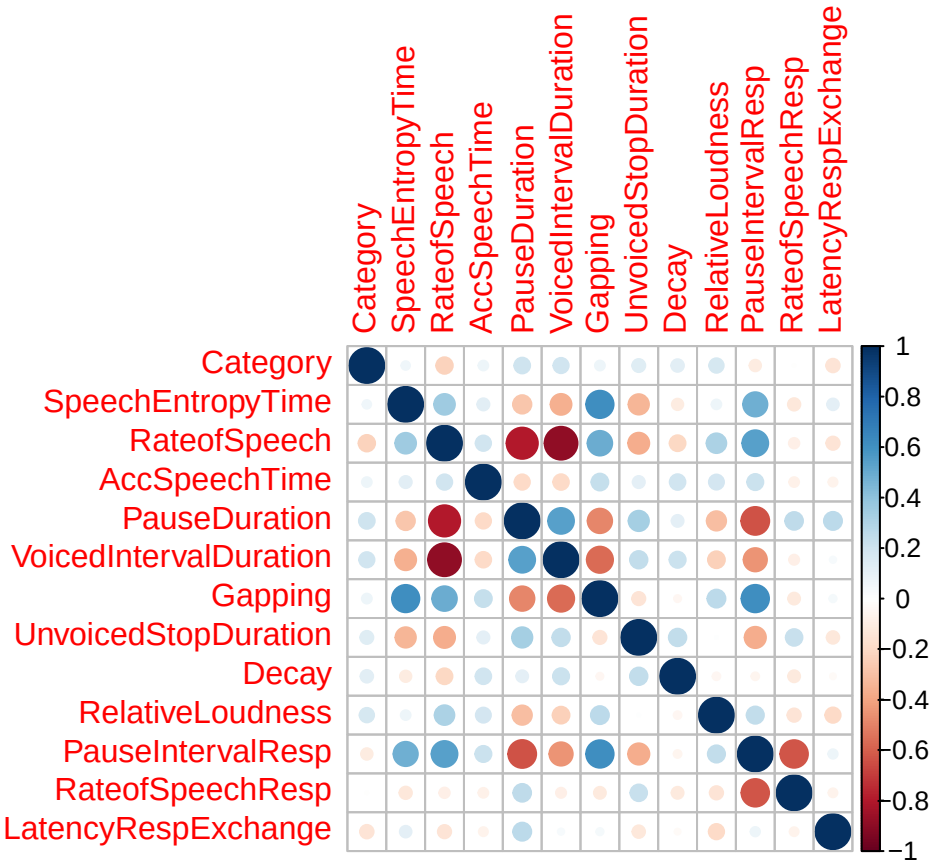
Selecting Features

Correlation:

```
# Using the method kendall for the categorical data
cor.cat <- cor(newdata[1:9], method = "kendall")
corrplot::corrplot(cor.cat)
```




```
# Using the method spearman for the numeric data
cor.num <- cor(cbind(Category = newdata$Category,newdata[10:21]),
               method = "spearman")
corrplot::corrplot(cor.num)
```

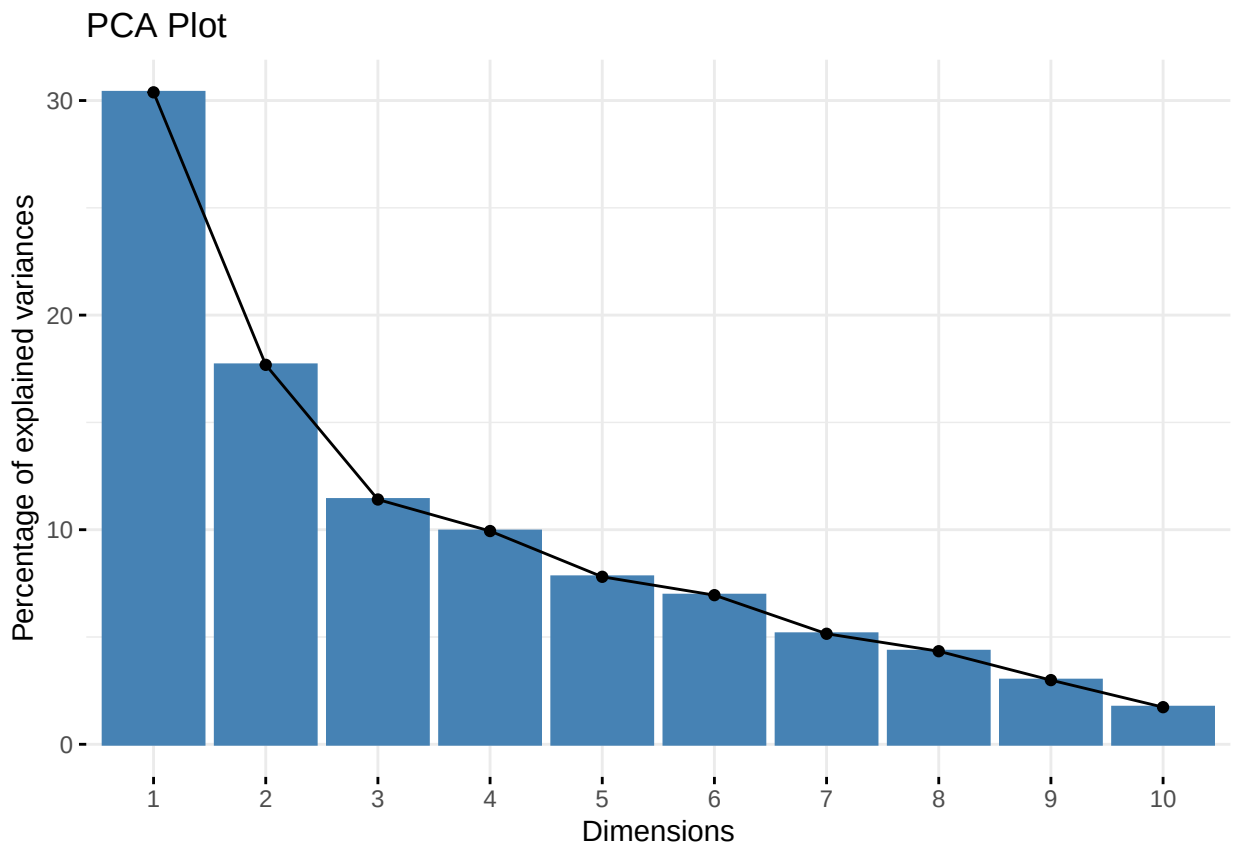


PCA Analysis:

```
library("factoextra")
```

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

```
PCA.comp<- prcomp(data.n)
fviz_eig(PCA.comp, main = "PCA Plot")
```




```
summary(PCA.comp)
```

```
## Importance of components:
##                            PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     0.3079 0.2349 0.1887 0.17611 0.15607 0.14728 0.12679
## Proportion of Variance 0.3038 0.1768 0.1141 0.09938 0.07805 0.06951 0.05151
## Cumulative Proportion  0.3038 0.4807 0.5947 0.69412 0.77217 0.84168 0.89319
##                            PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.11635 0.09661 0.07348 0.05898 0.03977
## Proportion of Variance 0.04338 0.02991 0.01730 0.01115 0.00507
## Cumulative Proportion  0.93657 0.96648 0.98378 0.99493 1.00000
```

It is observed that the first 3 features have above 10% variance and that is the reason why I have included them even thought they have very low correlation.

```
#Selecting variable columns, ignoring the columns with very high and very
#low correlation
vars <- c(1:5,9,10,12,13,14,16,17,18)
```

```
# Obtaining data for the models with the extracted features
data.var <- as.data.frame(newdata[vars])
data.var$Category <- as.factor(data.var$Category)
```

Modeling

I have focused on the caret package for builing the models as they have the ease to cross-validate and tune the models while training them.

Splitting the data:

The data is splt into 80-20 proportion with each disease category equally represented in each set. The validation data will be used in the hold-out validation.

```
set.seed(1010)
library(caret)
```

```
##
## Attaching package: 'caret'

## The following object is masked from 'package:survival':
##
##     cluster
```

```
sample <- createDataPartition(data.var$Category, p = 0.8, list = FALSE)
train <- data.var[sample,]
valid <- data.var[-sample,]
```

```
#Creating factors or the disease category in the training and validation set
train$Category <- as.factor(as.character(train$Category))
valid$Category <- as.factor(as.character(valid$Category))
```

Model Selection

Since, the majority of the features in the dataset are categorical and I have not dummy coded the features. The best suited models would be naive bayes, decision trees and neural network. These models are good when it comes to handling categorical vairables.

k-fold cross validation: All the models have implementation of k-fold cross validation with 10 folds

Metric: All the models are compared on the Accuracy metric. Since, the models are used for classification they cannot be compared on the basis of RMSE/MAD. The models will be compared on the value of Kappa and Accuracy.

Naive Bayes Classifier

```
#NAIVE BAYES
set.seed(1010)
library(caret)

# Naive Bayes model
nb.mod <- train(Category ~ ., data=train, method = "naive_bayes", metric = "Accuracy",
                trControl= trainControl(method = "cv", number = 10 ))
# Hold out validation
nb.pred <- predict(nb.mod, valid)

nb.output <- confusionMatrix(valid$Category,nb.pred)
nb.output
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3
##          1 10  0  0
##          2  2  2  2
##          3  7  2  1
##
## Overall Statistics
##
##                Accuracy : 0.5
##                  95% CI : (0.2993, 0.7007)
##     No Information Rate : 0.7308
##     P-Value [Acc > NIR] : 0.99673
##
##                   Kappa : 0.2176
##
##  Mcnemar's Test P-Value : 0.02929
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            0.5263  0.50000  0.33333
## Specificity            1.0000  0.81818  0.60870
## Pos Pred Value         1.0000  0.33333  0.10000
## Neg Pred Value         0.4375  0.90000  0.87500
## Prevalence             0.7308  0.15385  0.11538
## Detection Rate         0.3846  0.07692  0.03846
## Detection Prevalence   0.3846  0.23077  0.38462
## Balanced Accuracy      0.7632  0.65909  0.47101
```

```
nb.accuracy <- nb.output$overall[[1]]
nb.kappa <- nb.output$overall[[2]]
nb.lower.ci <- nb.output$overall[[3]]
nb.upper.ci <- nb.output$overall[[4]]
```

It is observed that the accuracy of the model is just 50% and that the model is not a good model based on the kappa value.

Decision Tree Classifier

```
set.seed(1010)

tune_grid <- expand.grid(cp=seq(0,0.5,0.05))

# Decision tree using rpart
t.mod <- train(Category ~ ., data=train,
               method = "rpart", metric = "Accuracy",
               trControl = trainControl(method = "cv", number = 10 ),
               tuneGrid = tune_grid)

#Hold-out validation
t.pred <- predict(t.mod,valid)
```

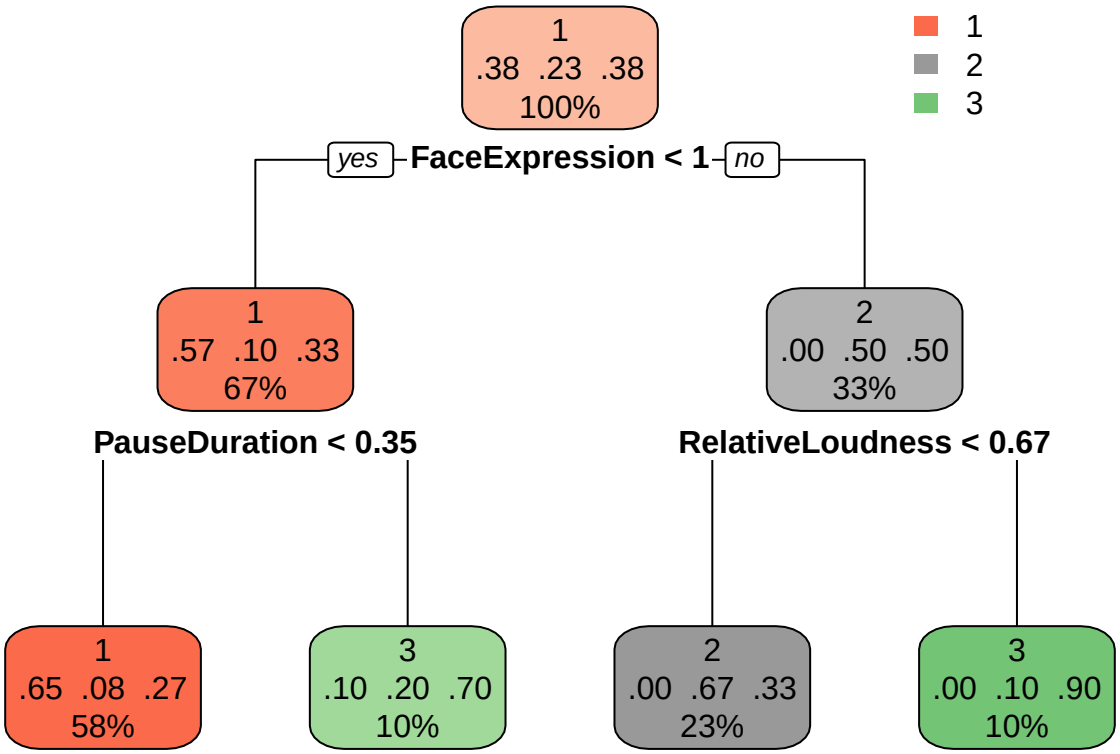
```
t.output <- confusionMatrix(valid$Category,t.pred)
t.output
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3
##          1 10  0  0
##          2  0  4  2
##          3  4  4  2
##
## Overall Statistics
##
##                Accuracy : 0.6154
##                  95% CI : (0.4057, 0.7977)
##     No Information Rate : 0.5385
##     P-Value [Acc > NIR] : 0.2791
##
##                   Kappa : 0.4196
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            0.7143   0.5000  0.50000
## Specificity            1.0000   0.8889  0.63636
## Pos Pred Value         1.0000   0.6667  0.20000
## Neg Pred Value         0.7500   0.8000  0.87500
## Prevalence             0.5385   0.3077  0.15385
## Detection Rate         0.3846   0.1538  0.07692
## Detection Prevalence   0.3846   0.2308  0.38462
## Balanced Accuracy      0.8571   0.6944  0.56818
```

```
tree.accuracy <- t.output$overall[[1]]
tree.kappa <- t.output$overall[[2]]
t.lower.ci <- t.output$overall[[3]]
t.upper.ci <- t.output$overall[[4]]
```

It is observed that the accuracy of the model is 61.54% and that the model is a fair model based on the kappa value.

```
# Decision tree Plot
rpart.plot::rpart.plot(t.mod$finalModel)
```



It is observed here that the FaceExpression feature is the one with the highest feature importance and is the root of the tree. The interior nodes consists of PauseDuration and RelativeLoudness with decreaing feature importance. The leaves give the predicted diease category.

Neural Network Classifier

```
set.seed(100)
# Neural network classifier
t.grid <- expand.grid(size=5,decay=0.2)
nnmodel <- train(train[,-1], train$Category, method = "nnet", metric = "Accuracy",
                 trControl= trainControl(method = "cv", number = 10 ),
                 tuneGrid = t.grid)
```

```
## # weights:  83
## initial  value 112.830321
## iter  10 value 85.050209
## iter  20 value 74.496183
## iter  30 value 73.140833
## iter  40 value 73.035293
## iter  50 value 73.028262
## iter  60 value 73.026817
## final  value 73.026783
## converged
## # weights:  83
## initial  value 129.973721
## iter  10 value 79.511027
```

```
## iter  20 value 73.789256
## iter  30 value 73.498526
## iter  40 value 73.119427
## iter  50 value 73.007508
## iter  60 value 73.000735
## iter  70 value 72.999460
## final  value 72.999448
## converged
## # weights:  83
## initial  value 112.649114
## iter  10 value 84.773482
## iter  20 value 74.087147
## iter  30 value 72.042441
## iter  40 value 71.764765
## iter  50 value 71.600544
## iter  60 value 71.584190
## iter  70 value 71.576447
## iter  80 value 71.565669
## iter  90 value 71.564968
## iter  90 value 71.564967
## iter  90 value 71.564967
## final  value 71.564967
## converged
## # weights:  83
## initial  value 106.927538
## iter  10 value 83.568790
## iter  20 value 76.749444
## iter  30 value 75.527904
## iter  40 value 74.725463
## iter  50 value 74.623144
## iter  60 value 74.590104
## iter  70 value 74.569137
## iter  80 value 74.564206
## iter  90 value 74.561840
## final  value 74.561813
## converged
## # weights:  83
## initial  value 116.688840
## iter  10 value 86.421129
## iter  20 value 75.480163
## iter  30 value 73.512399
## iter  40 value 73.216123
## iter  50 value 73.043457
## iter  60 value 73.039812
## iter  70 value 73.039032
## iter  80 value 73.039001
## iter  90 value 73.038977
## final  value 73.038948
## converged
## # weights:  83
## initial  value 103.484367
## iter  10 value 83.158931
## iter  20 value 75.814798
## iter  30 value 75.091883
```

```
## iter  40 value 74.814128
## iter  50 value 74.223270
## iter  60 value 73.791379
## iter  70 value 73.709241
## iter  80 value 73.694601
## iter  90 value 73.689215
## iter 100 value 73.687905
## final  value 73.687905
## stopped after 100 iterations
## # weights:  83
## initial  value 114.265793
## iter  10 value 83.452826
## iter  20 value 74.119515
## iter  30 value 72.753627
## iter  40 value 72.207007
## iter  50 value 71.921132
## iter  60 value 71.852851
## iter  70 value 71.783128
## iter  80 value 71.772189
## iter  90 value 71.770773
## iter 100 value 71.770523
## final  value 71.770523
## stopped after 100 iterations
## # weights:  83
## initial  value 104.056258
## iter  10 value 78.533228
## iter  20 value 75.316330
## iter  30 value 75.114204
## iter  40 value 75.075480
## iter  50 value 75.052603
## iter  60 value 75.051295
## final  value 75.051241
## converged
## # weights:  83
## initial  value 109.449136
## iter  10 value 83.927121
## iter  20 value 77.251344
## iter  30 value 76.228082
## iter  40 value 75.810966
## iter  50 value 75.701259
## iter  60 value 75.677996
## iter  70 value 75.673001
## iter  80 value 75.672076
## iter  90 value 75.672019
## final  value 75.672004
## converged
## # weights:  83
## initial  value 107.781619
## iter  10 value 83.442706
## iter  20 value 75.587324
## iter  30 value 73.322778
## iter  40 value 73.145391
## iter  50 value 73.039829
## iter  60 value 73.008636
```

```
## iter  70 value 73.006810
## iter  80 value 73.006288
## iter  90 value 73.006137
## final  value 73.006130
## converged
## # weights:  83
## initial  value 119.691805
## iter  10 value 90.153267
## iter  20 value 82.644292
## iter  30 value 81.535279
## iter  40 value 81.086980
## iter  50 value 80.952121
## iter  60 value 80.918451
## iter  70 value 80.917014
## final  value 80.916951
## converged
```

```
#Hold-out validation
nnprediction <- predict(nnmodel, valid)

nnet.output <- confusionMatrix(nnprediction, valid$Category)
nnet.output
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 1 2 3
##          1 9 1 3
##          2 0 4 0
##          3 1 1 7
##
## Overall Statistics
##
##                Accuracy : 0.7692
##                  95% CI : (0.5635, 0.9103)
##     No Information Rate : 0.3846
##     P-Value [Acc > NIR] : 7.573e-05
##
##                   Kappa : 0.6389
##
##  Mcnemar's Test P-Value : 0.3916
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            0.9000   0.6667   0.7000
## Specificity            0.7500   1.0000   0.8750
## Pos Pred Value         0.6923   1.0000   0.7778
## Neg Pred Value         0.9231   0.9091   0.8235
## Prevalence             0.3846   0.2308   0.3846
## Detection Rate         0.3462   0.1538   0.2692
## Detection Prevalence   0.5000   0.1538   0.3462
## Balanced Accuracy      0.8250   0.8333   0.7875
```

```
nnet.accuracy <- nnet.output$overall[[1]]
nnet.kappa <- nnet.output$overall[[2]]
nnet.lower.ci <- nnet.output$overall[[3]]
nnet.upper.ci <- nnet.output$overall[[4]]
```

It is observed that the accuracy of the model is just 76.92% and that the model is a good model based on the kappa value.

Feature Importance using Random forest model

```
set.seed(100)
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
##
## Attaching package: 'randomForest'
```

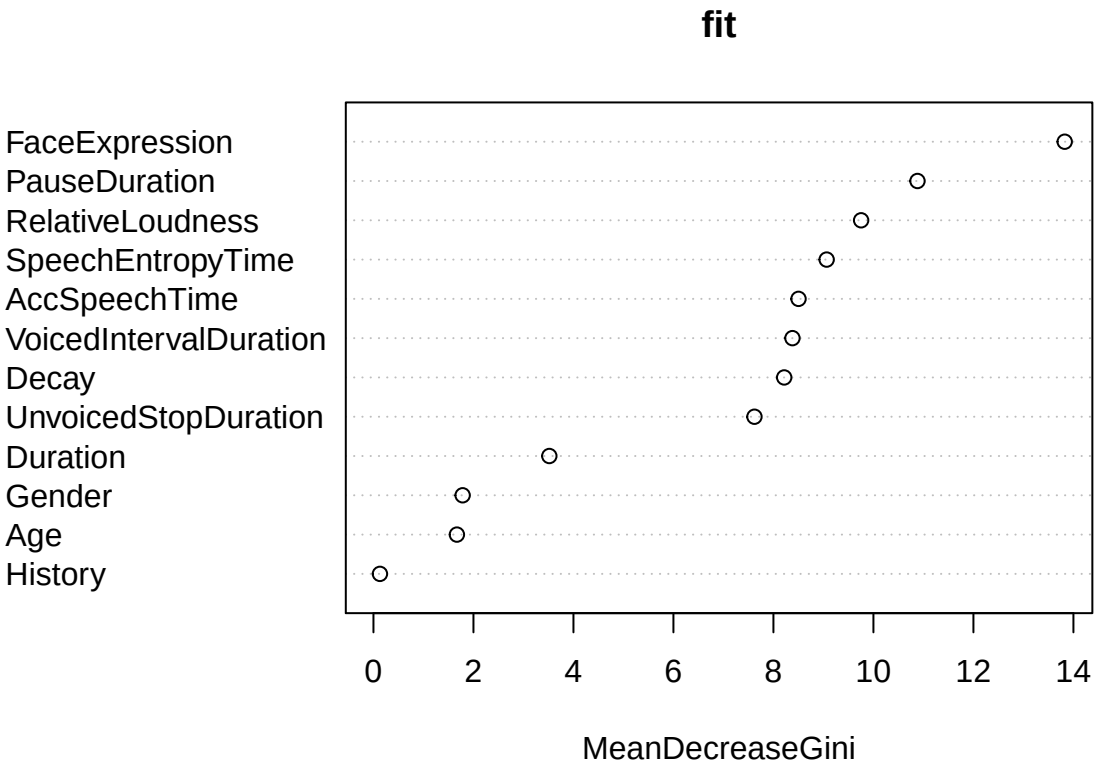
```
## The following object is masked from 'package:psych':
##
##     outlier
```

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

```
fit <- randomForest(Category ~ .,data = data.var)
# Feature importance value
importance(fit)
```

```
##                         MeanDecreaseGini
## Age                            1.6691313
## Gender                         1.7838051
## History                        0.1296881
## Duration                       3.5175698
## FaceExpression                13.8258035
## SpeechEntropyTime              9.0646303
## AccSpeechTime                  8.5018383
## PauseDuration                 10.8813244
## VoicedIntervalDuration         8.3803601
## UnvoicedStopDuration           7.6202264
## Decay                          8.2152165
## RelativeLoudness               9.7538341
```

```
# Plot for the importance of features
varImpPlot(fit)
```



It is observed that the feature: FaceExpression, PauseDuration, SpeechEntropyTime, RelativeLoudness , AccSpeechTime, VoicedIntervalDuration, Decay, RateOfSpeech and UnvoicedStopDuration are the most important features which could be used as Early Biomarkers of prediction of Parkinsons disease.

When comparing the top three features to that obtained by the decision tree model are the same and those obtained from the random forest model.

Stacked Ensemble Model

```
# The predicted data from all the models is ensembled
ensemble.data <- data.frame(nb.pred,t.pred,nnprediction,
                            Category = valid$Category,
                            stringsAsFactors = F)
# The random forest model is used as an ensemble model with 10 fold cross validation
modelStack <- train(Category ~ ., data = ensemble.data, method = "rf",
                    trControl= trainControl(method = "cv", number = 10))
# Hold- out validation
combPred <- predict(modelStack, ensemble.data)

ensemble.output <- confusionMatrix(combPred, valid$Category)
ensemble.output
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3
##          1 10  1  3
```

```
##          2  0  4  0
##          3  0  1  7
##
## Overall Statistics
##
##                Accuracy : 0.8077
##                  95% CI : (0.6065, 0.9345)
##     No Information Rate : 0.3846
##     P-Value [Acc > NIR] : 1.298e-05
##
##                   Kappa : 0.6991
##
##  Mcnemar's Test P-Value : 0.1718
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            1.0000   0.6667   0.7000
## Specificity            0.7500   1.0000   0.9375
## Pos Pred Value         0.7143   1.0000   0.8750
## Neg Pred Value         1.0000   0.9091   0.8333
## Prevalence             0.3846   0.2308   0.3846
## Detection Rate         0.3846   0.1538   0.2692
## Detection Prevalence   0.5385   0.1538   0.3077
## Balanced Accuracy      0.8750   0.8333   0.8187
```

```
ensemble.accuracy <- ensemble.output$overall[[1]]
ensemble.kappa <- ensemble.output$overall[[2]]
ensemble.lower.ci <- ensemble.output$overall[[3]]
ensemble.upper.ci <- ensemble.output$overall[[4]]
```

Outcome

```
accuracy <- c(nb.accuracy,tree.accuracy,nnet.accuracy,ensemble.accuracy)
kappa <- c(nb.kappa,tree.kappa,nnet.kappa,ensemble.kappa)
CI.range <- c((nb.upper.ci-nb.lower.ci),
              (t.upper.ci-t.lower.ci),
              (nnet.upper.ci-nnet.lower.ci),
              (ensemble.upper.ci-ensemble.lower.ci))

compared.data <- cbind(Accuracy = accuracy, Kappa = kappa, CIRange = CI.range)
colnames(compared.data) <- c("Accuracy", "Kappa", "CI Range")
rownames(compared.data) <- c("Naive Bayes", "Decision Tree", "Neural Network", "Ensembled model")
compared.data <- data.frame(compared.data)
compared.data
```

```
##                  Accuracy     Kappa  CI.Range
## Naive Bayes     0.5000000 0.2175926 0.4014556
## Decision Tree   0.6153846 0.4196429 0.3920322
## Neural Network  0.7692308 0.6388889 0.3467350
## Ensembled model 0.8076923 0.6990741 0.3279574
```

It is observed that Naive Bayes is not a good classifier for this dataset even though it is known to handle categorical data well. Decision tree is observed to be better than Naive Bayes and Neural Network is observed to be the best classifer in this dataset as it has the highest accuracy, kappa value and a small confidence interval range. Since, we are dealing with data from human study a kappa value of 0.41 is an acceptable value as per certain studies.

When the ensembled model is considered, it gives a good accuracy of 80.76% and a kappa value of 0.70 which a model model and the confidence interval range is the least.

References

Automated analysis of connected speech reveals early biomarkers of Parkinson's disease in patients with rapid eye movement sleep behaviour disorder by Jan Hlavnička, Roman Čmejla, Tereza Tykalová, Karel Šonka, Evžen Růžička & Jan Rusz

https://www.nature.com/articles/s41598-017-00047-5

Interrater reliability: the kappa statistic by Mary L. McHugh

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3900052/