# Using Quantified Voice Patterns to diagnose the severity of Parkinson's disease.

David Northey

2024-04-22

## Abstract

Diagnosis for Parkinson's Disease is a time-consuming endeavor requiring trained medical staff which can take years to perform an accurate assessment. The purpose of this report is to use quantified voice patterns to determine the severity of Parkinson's disease. If successful, this would be a less invasive diagnostic tool than conventional methods. To do this we used Principal Component Analysis (PCA) to reduce the dimensionality of the dataset. The output from this unsupervised learning technique meets the assumptions required to perform some of the supervised learning algorithms. These include logistic regression, random forest and K-Nearest Neighbours (kNN). Following these methods, a further unsupervised learning of cluster analysis using k-means was untilised to further explore the nature of the dataset.

The supervised learning models produced in this investigation performed poorly unfortunately. The random forest and K-NN models performed the best but given the small number of patients, the models have misdiagnosed patients too readily. When visualising the data using k means clustering, most of the data forms into one cluster. Further data collection and analysis will be required to develop better prediction models for Parkinson's disease.

## Introduction

Parkinson's Disease is a serious medical condition that affects the nervous system. Some diagnostic tools used to determine the severity of the condition include both motor and total Unified Parkinson's Disease Rating Scale (UPDRS). Diagnosing Parkinson's disease can take years too accurately diagnose (Tsanas. A, et al, 2009) requiring a lot of resources. Being able to develop new, quicker methods to aid in diagnosis would be advantageous.

The objective of this report is to create an accurate training model that can effectively predict whether or not a patient is in two categories of Parkinson's disease, these two categories are 'mild' and 'advanced'. If this can be performed effectively, then voice pattern analysis could be used to accompany or even replace conventional diagnostic techniques for Parkinson's disease.

## Data

The data was sourced from the University of California Irvine (UCI) machine learning repository (*UCI Machine Learning Repository, 2023*). The dataset was collected by recording the voice measurements of 42 patients over a six month period as part of an observational study. The patients have all been diagnosed with early-stage Parkinson's disease. Each of the recordings were captured in the patient's own homes. The data was donated to UCI on the 28/10/2009 from the university of Oxford working with ten medical centers in the US.

The Parkinsons dataset has a dimension of 5875 observations by 23 variables. There were no nulls contained within the dataset, so there will be no need to impute/remove missing data. An additional variable called 'severity' will be created which will be a factor of two classes converted from the "total_UPDRS" variable. This will be based on exploratory data analysis techniques and will be used to determine if the quantified voice patterns can identify the severity of Parkinson's disease.

Table 1: A table showing the variables within the dataset.

| # | Name | Type | Description |
|---|---|---|---|
| 1 | Subject# | Integer | Unique ID for each patient |
| 2 | Age | Integer | Patient age. |
| 3 | test_time | Continuous | The number of days since the patient has been recruited. |
| 4 | Jitter(%) | Continuous | A measurement of frequency |
| 5 | Jitter(ABS) | Continuous | A measurement of frequency |
| 6 | Jitter:RAP | Continuous | A measurement of frequency |
| 7 | JitterPPQ5 | Continuous | A measurement of frequency |
| 8 | Jitter:DDP | Continuous | A measurement of frequency |
| 9 | Shimmer | Continuous | A measurement of amplitude |
| 10 | Shimmer(dB) | Continuous | A measurement of amplitude |
| 11 | Shimmer:APQ3 | Continuous | A measurement of amplitude |
| 12 | Shimmer:APQ5 | Continuous | A measurement of amplitude |
| 13 | Shimmer:APQ11 | Continuous | A measurement of amplitude |
| 14 | Shimmer:DDA | Continuous | A measurement of amplitude |
| 15 | NHR | Continuous | Ratio of noise to tonal components in the voice |
| 16 | HNR | Continuous | Ratio of noise to tonal components in the voice |
| 17 | RPDE | Continuous | A non linear complexity measure. |
| 18 | DFA | Continuous | Signal fractal scaling exponent |
| 19 | PPE | Continuous | A nonlinear measure of fundamental frequency variation. |
| 20 | motor_UPDRS | Continuous | Doctor's motor UPDRS score. |
| 21 | total_UPDRS | Continuous | Doctor's total UPDRS score. |
| 22 | sex | Integer | Gender of the patient. |
| 23 | severity | Factor/character | This variable was created using the total_UPDRS variable. "mild", "advanced" |

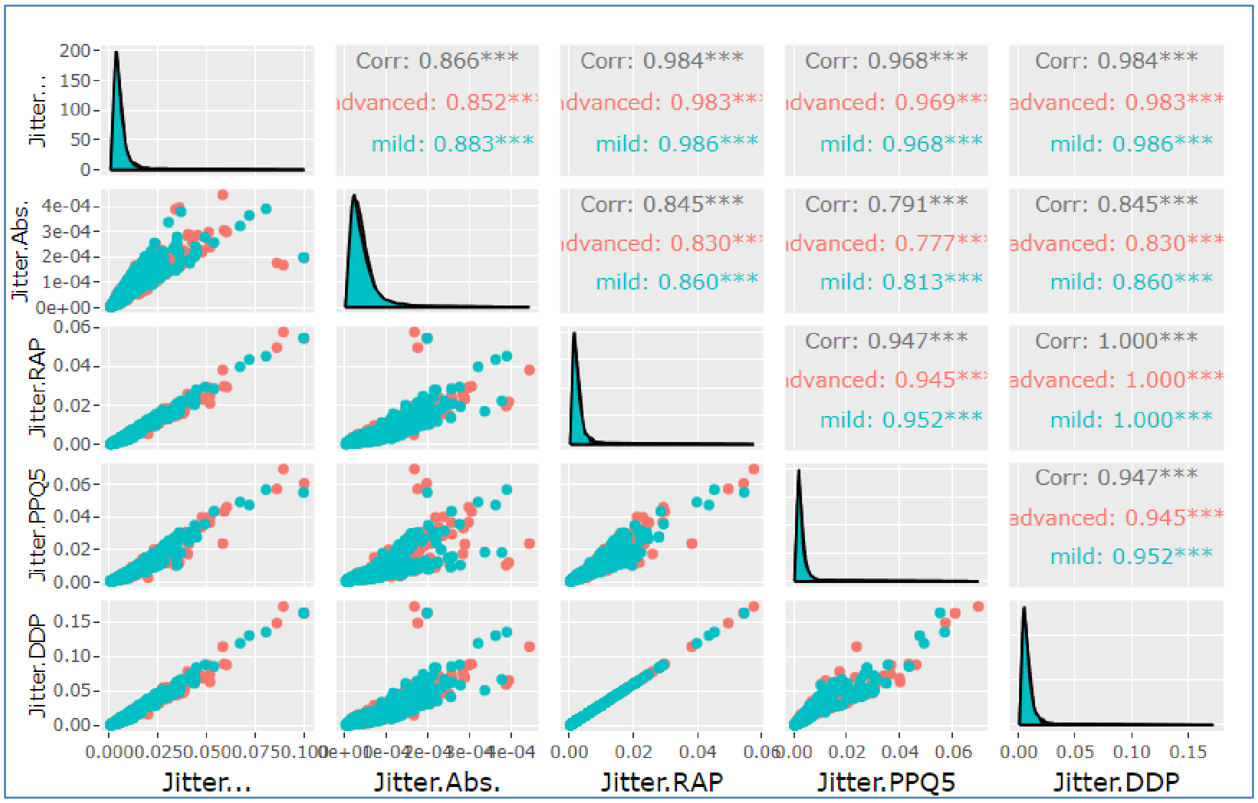
## Exploratory Data Analysis

When viewing the summary of each of the variables, we can confirm that there are no missing values within the dataset. The mean value for total_UPDRS is measuring at 29, therefore this shall be the point at which the severity factor variable will split the classes "mild" and "advanced".

When checking the proportion of the "severity" variable, we can see that there are two groups created that are roughly the same size. 45% are "advanced" and 55% are "mild" which is what we would expect.

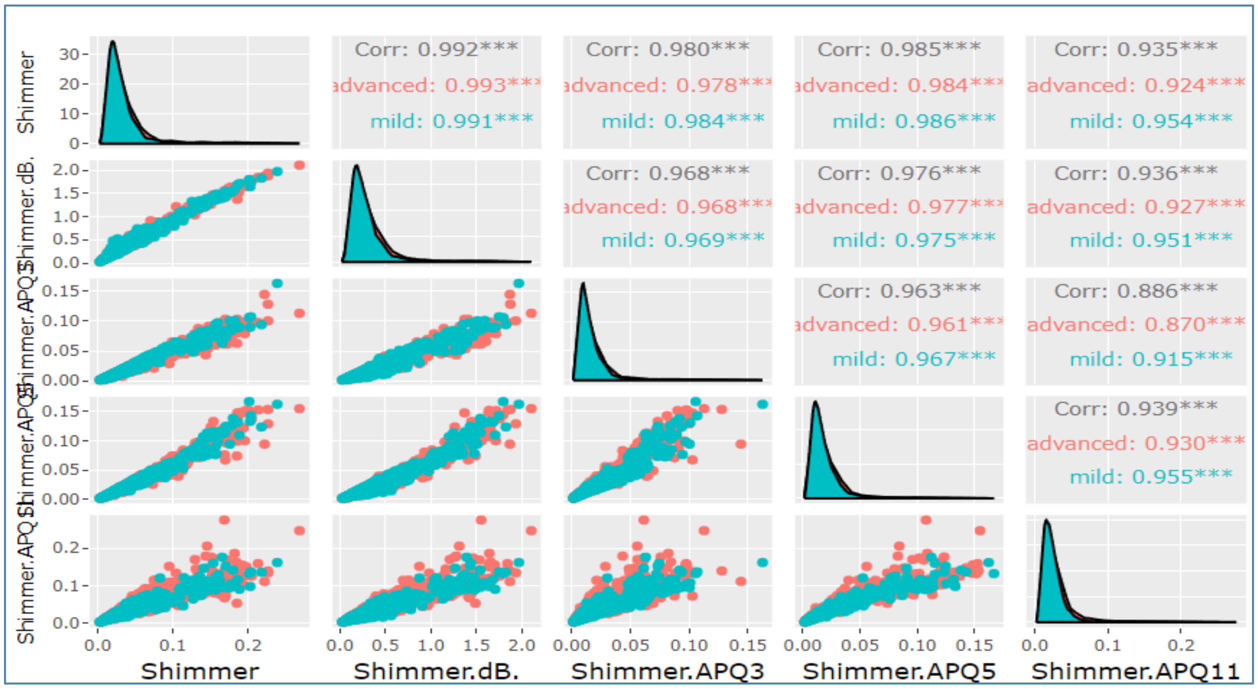
To further explore the dataset, some graphs were produced to demonstrate the amount of correlation and distribution type within the dataset.

Graph 1: A series of graphs showing the correlation and distribution between the Jitter variables within the dataset.



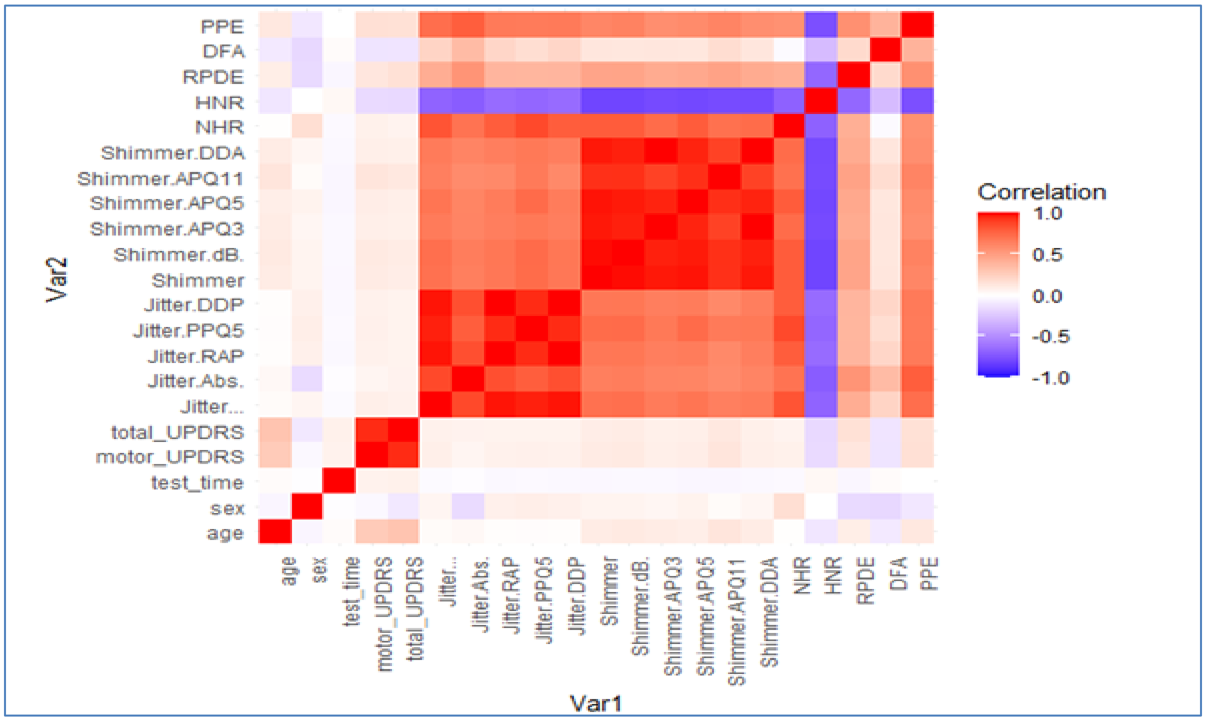
As we can see from the jitter variables, there is a high amount of correlation within the dataset. Also, it would seem that the distribution is not normally distributed. This would rule out certain classifiers such as Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) as the normal distribution assumption has been violated. A similar pattern can be seen for the shimmer variables also. Due to the high amount of correlation, it would also not be suitable to use other classifiers that require low multicollinearity to work well.

Graph 2: A series of graphs showing the correlation and distribution between the Shimmer variables within the dataset



To further assess the amount of multicollinearity within the dataset, we shall create a heat map.

Graph 3: A Heatmap showing the amount of correlation within the Parkinson's dataset



Looking at the heat map, there appears to be a strong correlation between the features for the shimmer and jitter variables.

## Methods

*R Studio version 2023.12.1.402 'Ocean Storm' was used for this investigation.*

To address the issue of multicollinearity and a high number of variables, let us perform Principal Component Analysis (PCA) to reduce the dimensionality of the dataset. PCA is in an Unsupervised Learning and is also good at removing inter-feature correlation by capturing the variation within the dataset. As we are looking to confirm if quantified voice patterns can be used to determine the severity of Parkinson's disease, we want to create an accurate model that only uses variables that can be easily obtained. Therefore, the 'severity', 'motor_UPDRS' and 'total_UPDRS' variables will be removed. Next, the data will be required to be scaled. This is as the variation between the variables are different, therefore the mean of each variable will be scaled to equal zero and the standard deviation will equal 1.

Next we will add the 'severity' variable into the PCA output by using the cbind() function and index the appropriate number of principal components based on the scree plot and summary of the principal components obtained. Using this unsupervised learning technique should provide us with some insights as to which variables account for the majority of the variation within the dataset, as well as providing data appropriate for the next stage for our analysis. Next, we will prepare different supervised learning models and assess which will perform the best using an AUC-ROC plot analysis to determine which is this best performing model. For this we will be using logistic regression, random forest and k-nearest neighbours to maximise the chance of determining Parkinson's disease severity using voice pattern analysis.

For each supervised learning algorithm, the null and alternate hypothesis is as follows:

**HO: There is no significant relationship between the severity of Parkinson's disease and the variables derived from voice pattern analysis**.

**HA: There is significant relationship between the severity of Parkinson's disease and the variables derived from voice pattern analysis.**

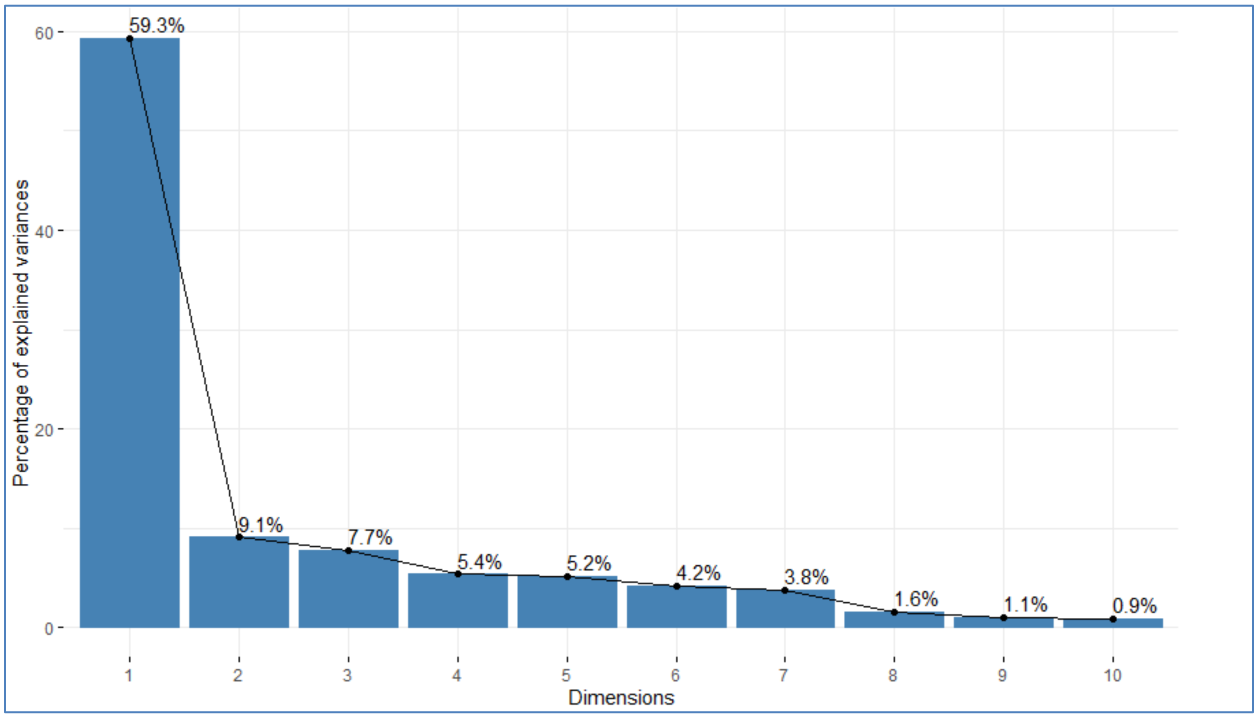
Due to the data now having a low multicollinearity, logistic regression and random forest techniques would be appropriate to use. Due to the scaling performed in the PCA step, the data will be appropriate for use for the kNN classifier. It will be interesting to see how well the kNN and random forest techniques perform as they should perform differently based on outliers present in the data. Generally, kNN will perform poorly in the presence of outliers, whereas random forest will be more robust (Phan, T, et al, 2018).

The data will need to be converted into a training and test set by dividing the data up into two portions. 80% of the data will be used for training and 20% will be used as the test. This provides around 4700 observations for training which is a good amount to provide an accurate model. Around 1200 observations will provide us with a good amount of feedback into how well the model is performing. If we had only a small amount of test data, we may find that no false negatives or false positives were detected, our test sample size will not have this issue.

For each of the supervised learning methods used, a 10 fold non overlapping cross validation procedure will be used. This way, multiple samples are taken from the data to create the models, improving the accuracy. We will set the seed of each model so that the same data is used, this will help when comparing the performance of each model. Next, we can view the confusion matrix formed from each model and view this information on the AUC-ROC curve. After obtaining results for the models created, we will continue to explore the data using another unsupervised learning method of k-means cluster analysis. The k-means analysis will be performed using the same selected variables that were used previously in this investigation (no UPDRS related variables). From this analysis, we will be able to see the number of clusters which have formed by using the elbow and silhouette methods.

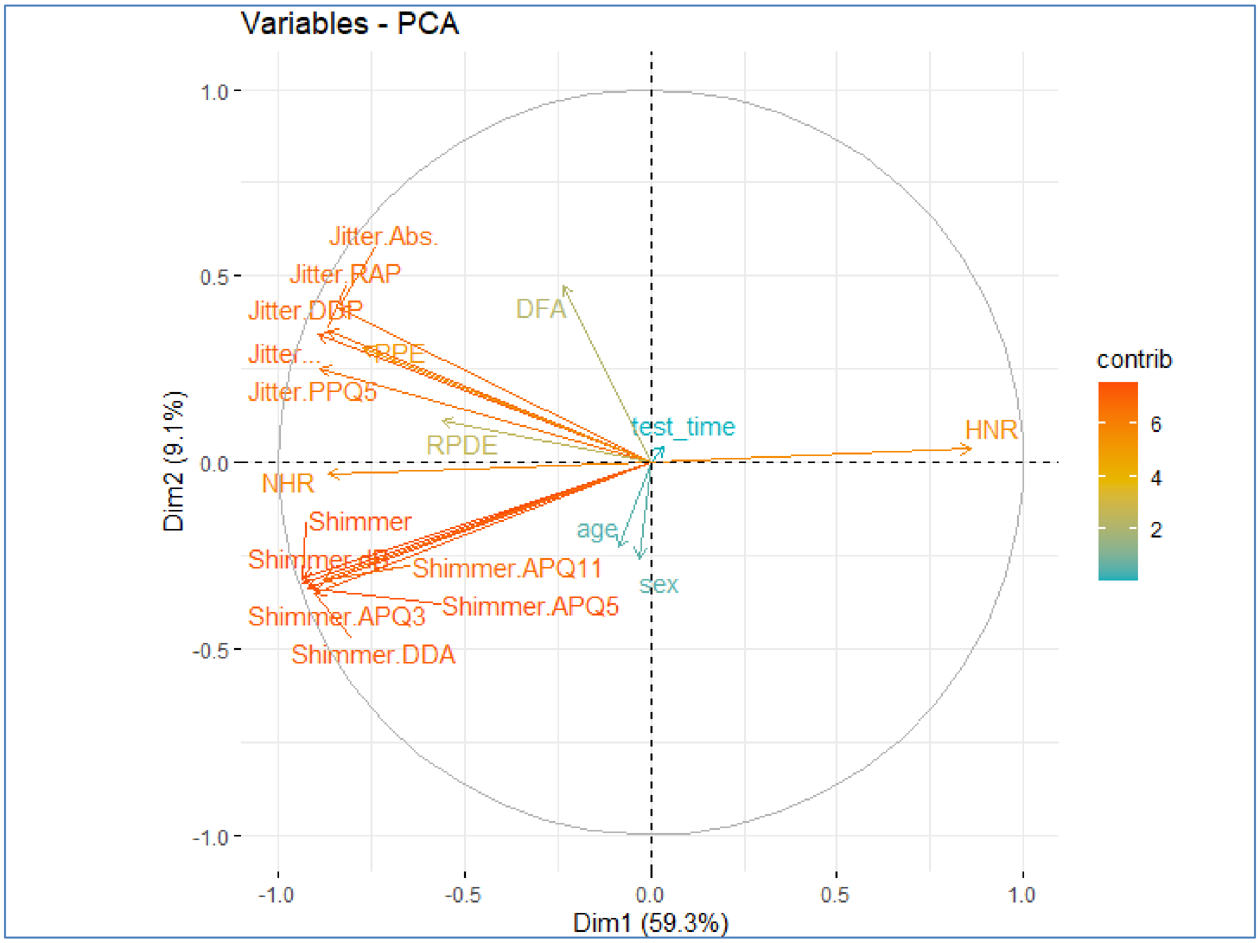
## Results

Graph 4: Scree plot showing the amount of variation present for each principal component.



Observing the scree plot (*graph 4*), there does seem to be a significant amount of variation captured within the first two principal components. Originally, only the first two principal components were going to be used due to there being a very distinctive elbow at 2 principal components. However, it could be seen that the performance of each model was quite poor. This could be due to the fact that only 68% of the total variation within the dataset has been captured. This could make the models formed inaccurate, missing key patterns integral to form a good prediction. From fine tuning by using a variety of principal components, it could be seen that the best performance was captured using 6 principal components. With this number of principal components, around 90% of the total data is attributed for this dataset. This could explain why the models perform well using this number of principal components while still addressing the original issue of high multicollinearity within the dataset.

Graph 5: A graph showing the relationship between the variables for principal components 1 and 2



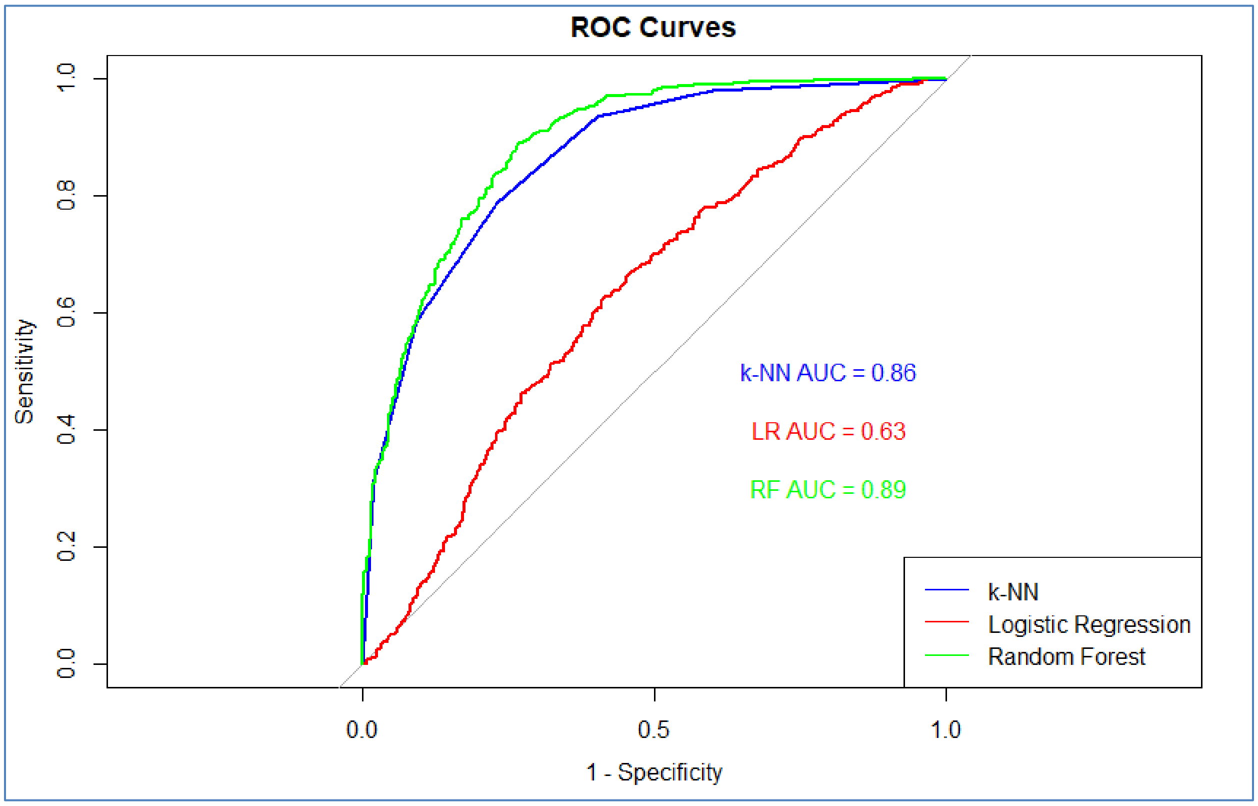
From the plot showing the eigenvectors for each variable *(graph 5)*, the jitter and shimmer variables all account for a similar amount of the variation measuring high eigenvalues compared to the other variables. This could potentially suggest that these variables will predict how much a patient is affected by Parkinson's disease well. Particularly as the first two principal components were responsible for most of the variation within the dataset.

After viewing the summary of the logistic regression model, although the p-values of each principal component are significant in rejecting the null hypothesis, the kappa (18%) and accuracy (60%) scores of the model are quite low. The specificity (73%) and sensitivity (46%) are quite low suggesting frequent misdiagnosis. Viewing the confusion matrix of the test data, a lot of false negatives and false positives can be identified demonstrating the lack of quality for this model.

The kappa (54%) and accuracy (77%) measurements of the k-nearest neighbours model have performed better than the logistic regression model. The kappa and accuracy values are higher indicating that this could be a respectable method to assess the severity of a patient in terms of Parkinson's disease. When viewing the confusion matrix of the test data, it can be observed that the accuracy measured at 78% with sensitivity of 79% and specificity of 78%. Although this model has performed better than the logistic regression model, it should be noted that there are still plenty of patients that have been misdiagnosed using the model from kNN.

From the random forest model, the results have performed similarly as the k-NN model. The kappa (57%) and accuracy (79%) scores of the random forest model are similar to the k-NN model. The confusion matrix of the test data demonstrates that the model has been able to classify the two classes quite well although there are still numerous false positives and false negatives which have been identified in the output. Overall the best performing algorithm for this dataset has been the random forest technique. This can also be seen when viewing the ROC-AUC curve plotted *(Graph 6).*

Graph 6: An AUC-ROC graph showing the performance of the three models created, k-NN, Logistic regression and Random Forests.



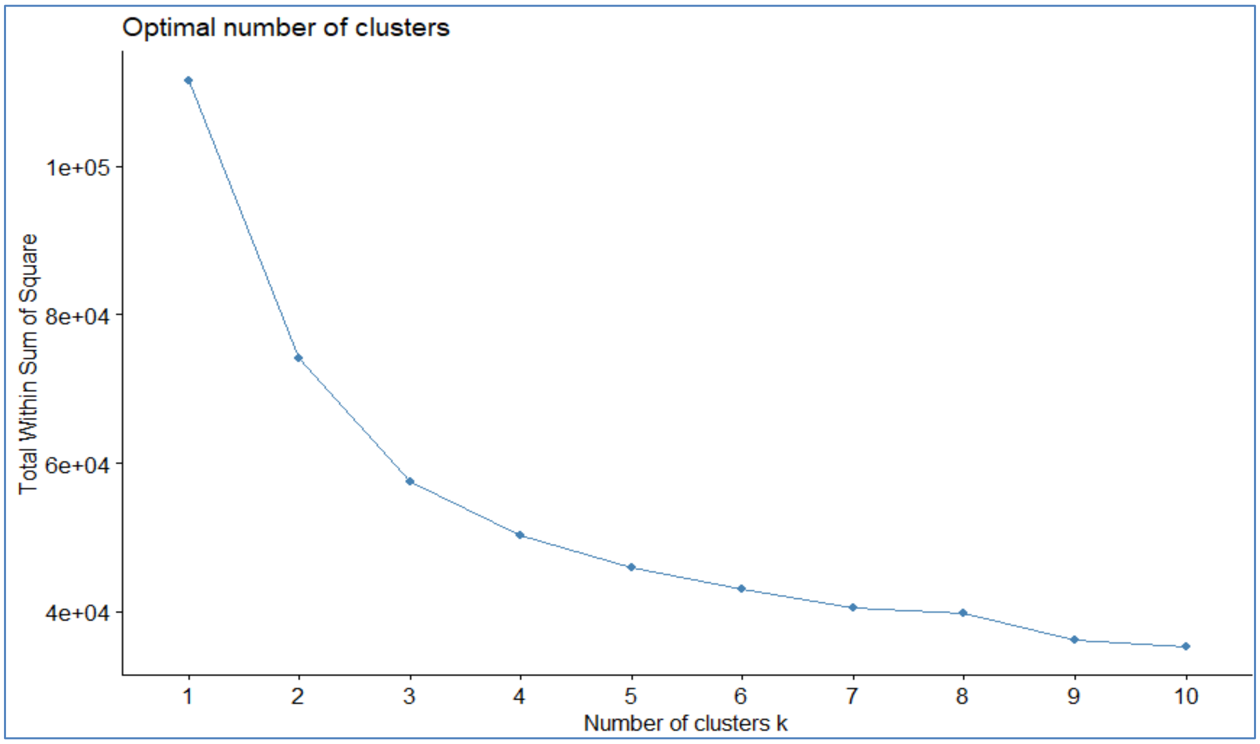
From the ROC-AUC curve, the best performing model was the random forest algorithm. Here we can see that the area under the curve value is the highest at 0.89. The k-NN also performed similarly with an area of 0.86. The worst performing was the logistic regression model with an area of 0.63. It is unusual that the k-NN and random forest algorithms performed similarly as the k-NN does not perform well with outliers, conversely the random forest can perform effectively. To further explore the data, an unsupervised learning technique of k-means was deployed. For the k-means analysis, we would like to test the following hypothesis:

**HO: The variables associated with voice pattern analysis will not form into two cluster types for parkinson's disease severity categorical type.**

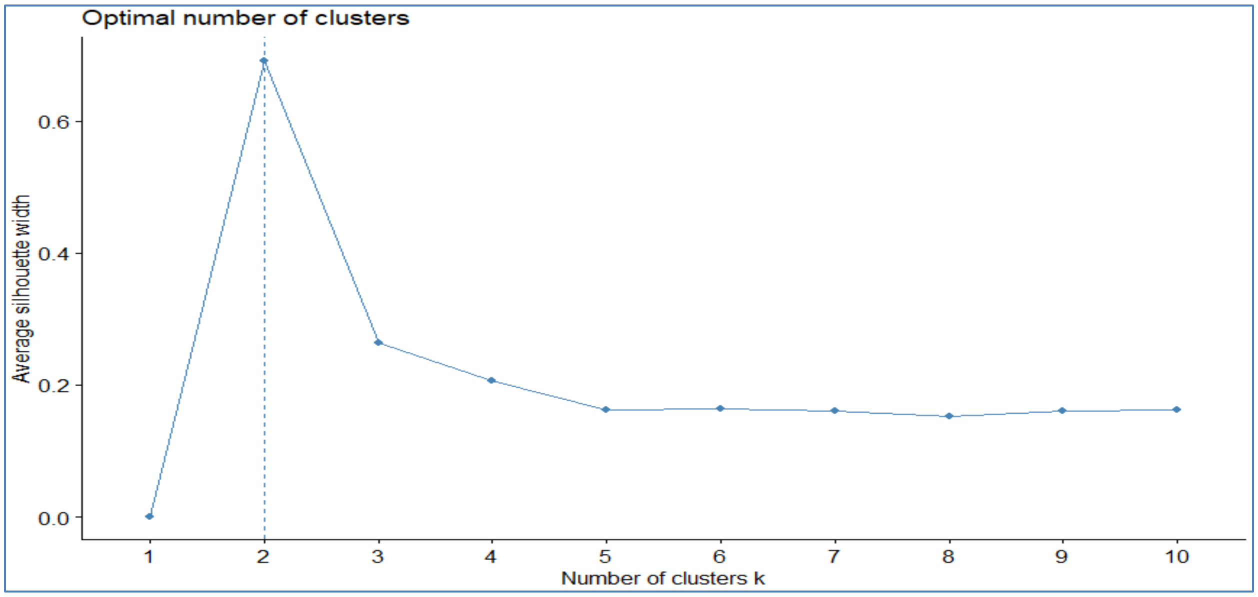
**HA: The variables associated with voice pattern analysis will form into two cluster types for parkinson's disease severity categorical type.**

When observing the amount of variation which is explained by the clusters within the dataset, the score is only 34% which indicates that the clusters are not very distinct. When viewing the ratio of the two severity classes, the ratio is 45:55. When viewing the cluster sizes obtained using k-means analysis, it could be observed that one cluster makes up 96% of the data and the other only makes up 4%. This would suggest that two distinct clusters have not formed. Let us identify if there are truly 2 clusters which form within the dataset using the elbow method and Silhouette Width Criterion (SWC) techniques.

Graph 7: A graph showing the number of clusters within the dataset.

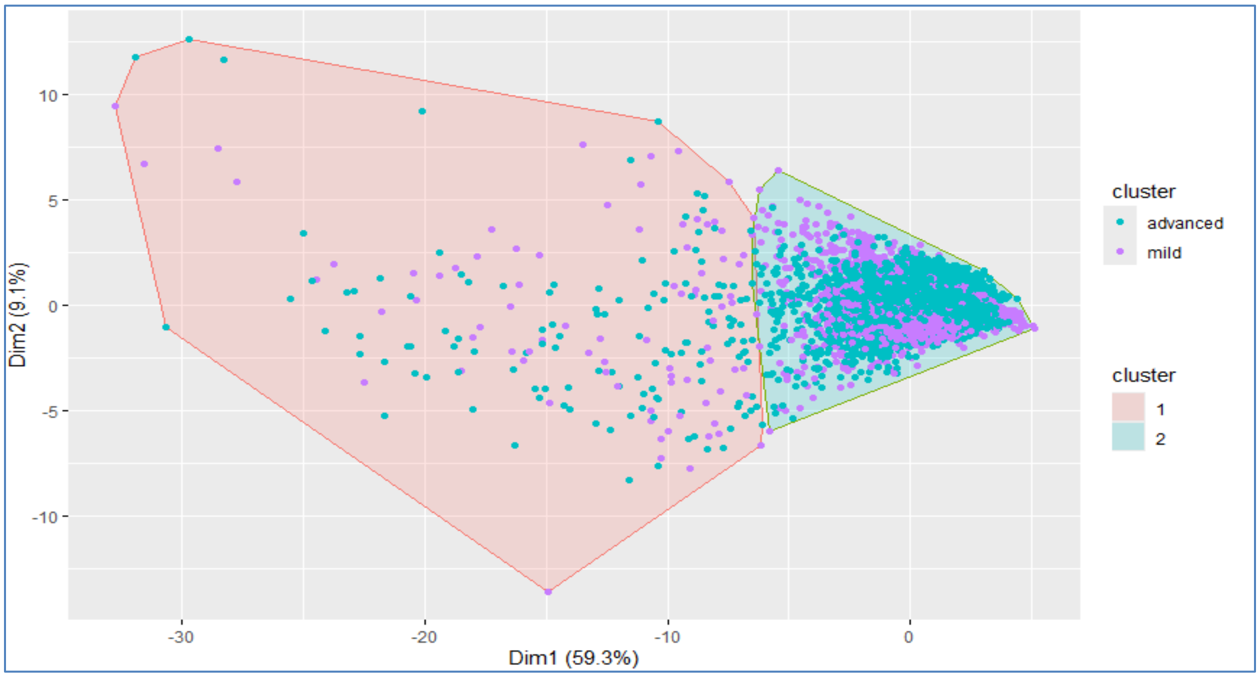


Graph 8: A graph showing the number of clusters within the dataset using the silhouette width criterion (SWC).



There was not a very distinctive elbow formed using the elbow method (*Graph 7*), but there did seem to be a slight elbow that formed between 2 and 3 clusters. When viewing the silhouette method (*Graph 8*), it was very clear that the number of clusters in the dataset is 2. Let us see the clusters visually using k = 2 to see if there is a pattern of the position of 'advanced' and 'mild' within the 2 clusters which have been identified.

Graph 9: A graph showing the clusters of the variables associated with voice pattern analysis using K-means analysis



When viewing the data *(Graph 9)*, there does seem to be two clusters of data present. However, the clusters which are formed are not ordered by the severity category. From observing this cluster analysis, we cannot observe any useful patterns within the dataset. Because of this we cannot reject the null hypothesis. Therefore, there is no evidence that the variables associated with voice pattern analysis form into two clusters based on the Parkinson severity classes.

## Conclusion

Although the random forest and kNN supervised learning algorithms could categorise the two classes of severity to a rough degree of accuracy, it should be noted that the data does have some limitations. This whole dataset was based only on 42 patients, each of whom recorded around 200 voice recordings. With this in mind, the supervised models should be performing much more accurately as the same patients would likely produce similar voice patterns.

Also, the patients used for this analysis were early onset parkinson cases. Therefore, the voice readings and total UPDRS scores would have only been from a small portion of the total scale. This could potentially explain why there were no distinct clusters which formed in the dataset when performing cluster analysis. It may also explain why the supervised learning algorithms did not perform as well as they should have. It would be interesting if it were possible to collect further data from patients who had more developed cases of Parkinson's disease and from a larger number of patients as well. All that we have been able to prove in this report, is that we can loosely diagnose the same 42 patients with slightly different voice patterns.

From this study, we are not able to reject the null hypothesis that there is no relationship between the severity of Parkinson's disease and voice pattern analysis. This method could be a useful tool to aid doctors in the diagnosis of parkinson's disease, but further study will be required to develop more robust algorithms trained, using a greater number of patients.

## References:

Ivey, F.M, Katzel, L,I. Sorkin, J.D, Macko, R. F, Shulman, L. M. (2012). The Unified Parkinson's Disease Rating Scale as a predictor of peak aerobic capacity and ambulatory function. *Journal of Rehabilitation Research and Development*, 49(8), 1269-1276. https://doi.org/10.1682/jrrd.2011.06.0103

James, G. Witten, D. Hastie, T. Tibshirani, R. (2021*), 'An introduction to Statistical Learning'*. Springer.

Little, M.A, McSharry, P.E, Hunter, E.J, Spielman, J, Ramig, L.O. (2009). Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. *IEEE Transactions on Biomedical Engineering*, 56(4), 1015. https://doi.org/10.1109/TBME.2008.2005954

Neurotoolkit. (2018). Unified Parkinson's Disease Rating Scale – *NeurologyToolKit*. https://neurotoolkit.com/updrs/

Phan, T. N, Kappas, M. (2018). Comparison of random forest, k-nearset neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery. *Sensors*, 18(1), https://doi.org/10.3390/s18010018

Tsanas, A. Little, M,A. McSharry, P.E. Ramig, L.O. (2010), 'Accurate telemonitoring of Parkinson's disease progression by non-invasive speech tests', *IEEE Transactions on Biomedical Engineering*, 57(4), 884-893. https://DOI:10.1109/TBME.2009.2036000

UCI Machine Learning Repository.(2023). *Parkinsons Telemonitoring.* https://archive.ics.uci.edu/dataset/189/parkinsons+telemonitoring

Verzani, J. (2014). *'Using R for Introductory statistics'*. CRC Press.

## R Code

```r
library(tidyverse)
library(mice)
library(MASS)
library(factoextra)
library(caret)
library(plotly)
library(GGally)
library(pROC)

parkinsons_data <- read.csv("parkinsons_updrs.data") #Load the
                                                    # parkinsons dataset

str(parkinsons_data) # View classes of the dataset
View(parkinsons_data)

summary(parkinsons_data) #Provide summary for each variable. The mean of the
                        #total_UPDRS is at 29. This will be the value used to
                        #seperate the two categories in the created 'severity
'                        #variable

parkinsons_data$severity <- ifelse(parkinsons_data$total_UPDRS > 29, "advance
d", "mild")
                              # This is to create the severity variable

table(parkinsons_data$severity) %>% prop.table()
                         # Here we are checking the proportion of each class

parkinsons_data %>%
  ggpairs(columns = c(7,8,9,10,11), aes(colour=severity)) %>%
  ggplotly()

parkinsons_data%>%
  ggpairs(columns = c(12,13,14,15,16), ggplot2::aes(colour=severity)) %>%
  ggplotly()

parkinsons_data_long <- parkinsons_data %>%


parkinsons_data$sex <- as.numeric(parkinsons_data$sex) #cor function requires
numeric data.
selected_variables <- parkinsons_data%>% select(-subject., -severity)


correlation_matrix <- cor(selected_variables)

ggplot(data = reshape2::melt(correlation_matrix), aes(Var1, Var2, fill = valu
e)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red",
                      midpoint = 0, limit = c(-1,1), space = "Lab",
                      name="Correlation") +
  theme_minimal() +
  coord_fixed() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

## PCA

```
selected_variables <- parkinsons_data %>% select(-subject., -motor_UPDRS, -to
tal_UPDRS, -severity) # Remove variables that would not allow the data to for
m patterns naturally. Remove -subject variable as this is not a characteristi
c of the data.

pr.out <- prcomp(selected_variables, center = T, scale=T) ## data is scaled w
hen performing PCA

pr.out$rotation

pca_data <- as.data.frame(pr.out$x)

comb_data <- cbind(severity = parkinsons_data$severity, pca_data) # adds the
severity column back into the PCA data.
comb_data$severity <- as.factor(comb_data$severity) # turns severity class in
to factor type. Necessary for algorithms.

fviz_eig(pr.out, addlabels = T)

selected_PC <- comb_data[1:7] # Select the number of principal components tha
t accounts for over 90% of the variation.
summary(pr.out)

fviz_pca_var(pr.out,
             col.var = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = T)
```

## Supervised Learning

```
count_obs <- dim(selected_PC)[1]
test_index <- sample(count_obs,
                              size = as.integer(count_obs*0.2),
                              replace = F) ## select test index of principal
components data

training_index <- -test_index ## select training index of principal component
s data

test_data <- selected_PC[test_index, ]
training_data <- selected_PC[-test_index, ]
```

## Logistic regression

```r
set.seed(10)
model_lr <- train(form = severity ~ .,
                  data = training_data,
                  trControl = trainControl(method = 'cv', number = 10),
                  method = 'glm',
                  family = 'binomial')

summary(model_lr)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6485  -1.1852   0.7639   1.0831   2.1308
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.205751   0.030201   6.813 9.58e-12 ***
## PC1          0.062233   0.009119   6.825 8.82e-12 ***
## PC2          0.130461   0.023318   5.595 2.21e-08 ***
## PC3         -0.160336   0.025445  -6.301 2.95e-10 ***
## PC4         -0.221933   0.030419  -7.296 2.97e-13 ***
## PC5          0.236647   0.030652   7.721 1.16e-14 ***
## PC6          0.181090   0.033969   5.331 9.76e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6474.3  on 4699  degrees of freedom
## Residual deviance: 6213.8  on 4693  degrees of freedom
## AIC: 6227.8
##
## Number of Fisher Scoring iterations: 4
```

```r
model_lr
```

```
## Generalized Linear Model
##
## 4700 samples
##    6 predictor
##    2 classes: 'advanced', 'mild'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4230, 4230, 4230, 4230, 4230, 4230, ...
## Resampling results:
##
```

```
##   Accuracy   Kappa
##   0.5987234  0.1723027
```

```
set.seed(10)
prediction_lr <- predict(model_lr, newdata = test_data)
confusionMatrix(prediction_lr, test_data$severity)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction advanced mild
##   advanced      211  148
##   mild          324  492
##
##                Accuracy : 0.5983
##                  95% CI : (0.5696, 0.6265)
##     No Information Rate : 0.5447
##     P-Value [Acc > NIR] : 0.0001187
##
##                   Kappa : 0.1677
##
##  Mcnemar's Test P-Value : 7.946e-16
##
##             Sensitivity : 0.3944
##             Specificity : 0.7688
##          Pos Pred Value : 0.5877
##          Neg Pred Value : 0.6029
##              Prevalence : 0.4553
##          Detection Rate : 0.1796
##    Detection Prevalence : 0.3055
##       Balanced Accuracy : 0.5816
##
##        'Positive' Class : advanced
##
```

**kNN**

```
set.seed(10)
model_knn <- train(form = severity ~ .,
                   data = training_data,
                   trControl = trainControl(method = 'cv', number = 10),
                   method = 'knn')

summary(model_knn)
```

```
##               Length Class      Mode
## learn         2      -none-     list
## k             1      -none-     numeric
## theDots       0      -none-     list
## xNames        6      -none-     character
## problemType 1      -none-     character
## tuneValue   1      data.frame list
## obsLevels   2      -none-     character
## param       0      -none-     list
```

```
model_knn
```

```
## k-Nearest Neighbors
##
## 4700 samples
##    6 predictor
##    2 classes: 'advanced', 'mild'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4230, 4230, 4230, 4230, 4230, 4230, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.8234043  0.6442656
##   7  0.8168085  0.6306773
##   9  0.8123404  0.6214972
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

```
set.seed(10)
prediction_knn <- predict(model_knn, newdata = test_data)
confusionMatrix(prediction_knn, test_data$severity)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction advanced mild
##   advanced      446  105
##   mild           89  535
##
##                Accuracy : 0.8349
##                  95% CI : (0.8124, 0.8557)
##     No Information Rate : 0.5447
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6679
##
##  Mcnemar's Test P-Value : 0.2815
##
##             Sensitivity : 0.8336
##             Specificity : 0.8359
##          Pos Pred Value : 0.8094
##          Neg Pred Value : 0.8574
##              Prevalence : 0.4553
##          Detection Rate : 0.3796
##    Detection Prevalence : 0.4689
##       Balanced Accuracy : 0.8348
##
##        'Positive' Class : advanced
##
```

### Random forest

```
set.seed(10)
model_rf <- train(form = severity ~ .,
                  data = training_data,
                  trControl = trainControl(method = 'cv', number = 10),
                  method = 'rf')

summary(model_rf)
```

```
##                 Length Class      Mode
## call               4   -none-     call
## type               1   -none-     character
## predicted       4700   factor     numeric
## err.rate        1500   -none-     numeric
## confusion          6   -none-     numeric
## votes           9400   matrix     numeric
## oob.times       4700   -none-     numeric
## classes            2   -none-     character
## importance         6   -none-     numeric
## importanceSD       0   -none-     NULL
## localImportance    0   -none-     NULL
## proximity          0   -none-     NULL
## ntree              1   -none-     numeric
## mtry               1   -none-     numeric
## forest            14   -none-     list
## y               4700   factor     numeric
## test               0   -none-     NULL
## inbag              0   -none-     NULL
## xNames             6   -none-     character
## problemType        1   -none-     character
## tuneValue          1   data.frame list
## obsLevels          2   -none-     character
## param              0   -none-     list
```

```
model_rf
```

```
## Random Forest
##
## 4700 samples
##    6 predictor
##    2 classes: 'advanced', 'mild'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4230, 4230, 4230, 4230, 4230, 4230, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.8334043  0.6637267
##   4     0.8346809  0.6663648
##   6     0.8400000  0.6769773
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 6.
```

```
set.seed(10)
prediction_rf <- predict(model_rf, newdata = test_data)
confusionMatrix(prediction_rf, test_data$severity)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction advanced mild
##   advanced      464   80
##   mild           71  560
##
##                Accuracy : 0.8715
##                  95% CI : (0.851, 0.8901)
##     No Information Rate : 0.5447
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7413
##
##  Mcnemar's Test P-Value : 0.515
##
##             Sensitivity : 0.8673
##             Specificity : 0.8750
##          Pos Pred Value : 0.8529
##          Neg Pred Value : 0.8875
##              Prevalence : 0.4553
##          Detection Rate : 0.3949
##    Detection Prevalence : 0.4630
##       Balanced Accuracy : 0.8711
##
##        'Positive' Class : advanced
##
```

### AUC-ROC graph

```
set.seed(10)
test_data$severity_numeric <- ifelse(test_data$severity == "advanced", 1, 0)
# Insert a variable that is
# in numeric format for severity
```

```r
prob_knn <- predict(model_knn, newdata = test_data, type = "prob")

# Create probability values for the ROC curve
# for each supervised learning model.
prob_rf <- predict(model_rf, newdata = test_data, type = "prob")

prob_lr <- predict(model_lr, newdata = test_data, type = "prob")

# Creates the roc curve for each model
roc_knn <- roc(test_data$severity_numeric, prob_knn[,"advanced"])

roc_lr <- roc(test_data$severity_numeric, prob_lr[,"advanced"])

roc_rf <- roc(test_data$severity_numeric, prob_rf[,"advanced"])



plot(roc_knn, col = "blue", main = "ROC Curves", legacy.axes = TRUE, print.au
c = F) # plot the curves on one graph
plot(roc_lr, col = "red", add = TRUE, print.auc = F, legacy.axes = TRUE)
plot(roc_rf, col = "green", add = TRUE, print.auc = F, legacy.axes = TRUE)


legend("bottomright", legend = c("k-NN", "Logistic Regression", "Random Fores
t"), col = c("blue", "red", "green"), lty = c(1, 1, 1))

# Add a legend to the graph


# print the AUC scores on the graph
text(0.2, 0.5, paste("k-NN AUC =", round(auc(roc_knn), 2)), col = "blue")
text(0.2, 0.4, paste("LR AUC =", round(auc(roc_lr), 2)), col = "red")
text(0.2, 0.3, paste("RF AUC =", round(auc(roc_rf), 2)), col = "green")
```

### K-means Cluster analysis

```r
selected_variables <- parkinsons_data %>% select(-subject., -motor_UPDRS, -to
tal_UPDRS, -severity)

std_cluster <- scale(selected_variables) ## scale the dataset
set.seed(10)
k2 <- kmeans(std_cluster, centers = 2, nstart = 10)

k2$betweenss/k2$totss ## calculated to show variation explained by clustering

parkinsons_data$severity <- as.character(parkinsons_data$severity)

table(k2$cluster) %>% prop.table()
table(parkinsons_data$severity) %>% prop.table()
```

```r
set.seed(10)
fviz_nbclust(std_cluster, kmeans, method='wss')
```

```r
set.seed(10)
fviz_nbclust(std_cluster, kmeans, method='silhouette')
```

```r
fviz_cluster(list(data = std_cluster, cluster = k2$cluster),
             geom = "point",
             shape = 1,
             show.clust.cent = F) +
  geom_point(aes(colour = parkinsons_data$severity)) +
  ggtitle("K-means Clusters of the variables asssociated with voice pattern a
nalysis")
```