**Alexandros Christoforos Mitronikas**
**ST 20152336**



# ASSIGNMENT COVER

| | |
|---|---|
| Course code: | CIS6005 |
| Course name: | Computational Intelligence |
| Assignment title: | Prediction and Classification of Parkinson's Disease using Machine Learning |
| Instructor's name: | Dr. K. Veropoulos |
| Student's name: | Alexandros – Christoforos Mitronikas |
| Date: | 21/1/2021 |
| Word count | 4447 |


Cardiff Metropolitan University | Prifysgol Metropolitan Caerdydd

**DECLARATION**

This work is the result of my own investigations, except where otherwise stated. This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed Alexandros – Christoforos Mitronikas (Candidate)

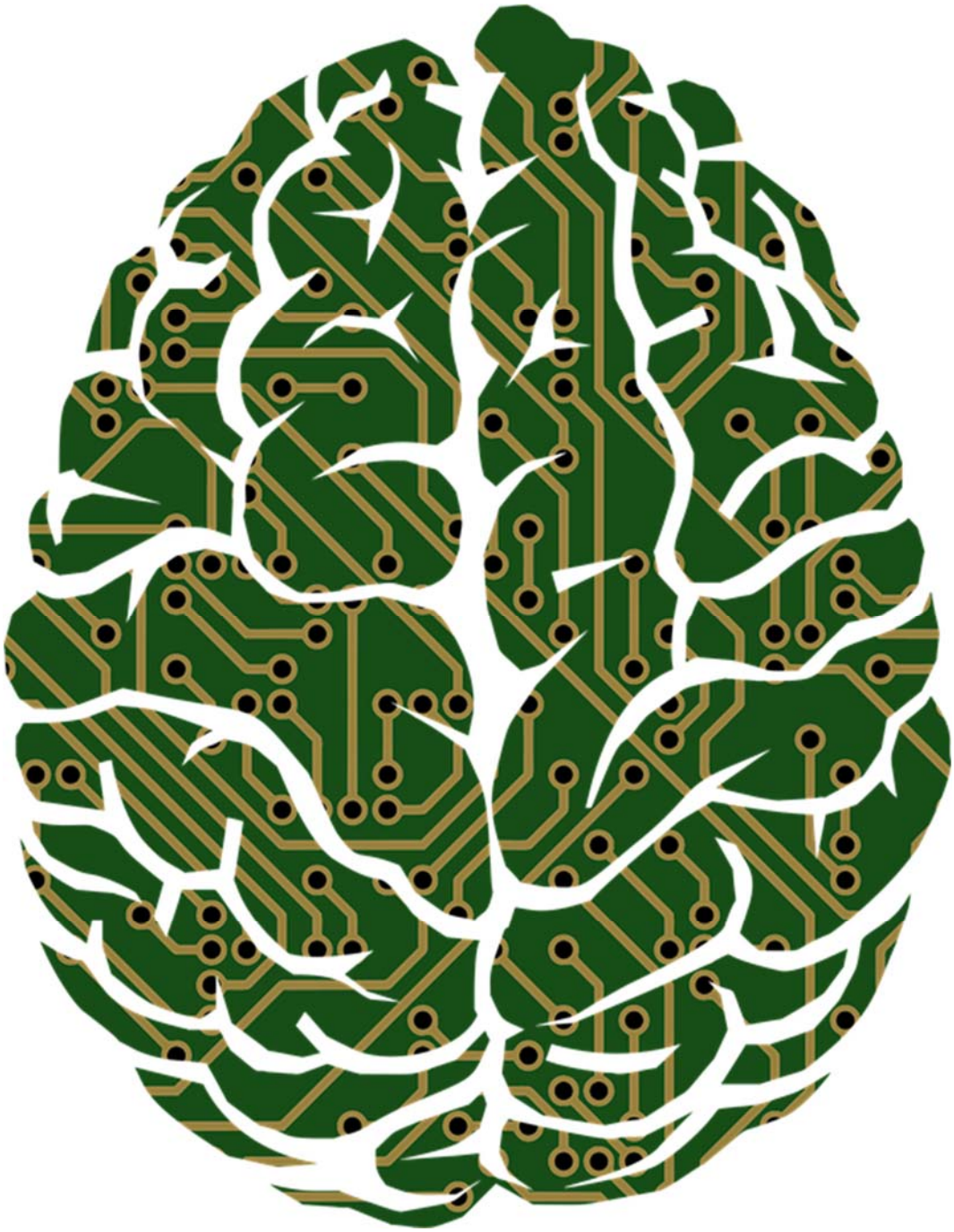
Date 21/1/2021

**Alexandros Christoforos Mitronikas**
**ST 20152336**

City unity College

Cardiff Metropolitan University

# Prediction and Classification of Parkinson's Disease using Machine Learning



## Contents

## Abstract

This paper, is the continuation of WRIT1 report which focuses on exploring different approaches for the binary classification problem to individuals, determining weather they have Parkinson's Disease with the help of machine learning algorithms. In this work, multiple machine learning algorithms have been tested using the Parkinson's Disease voice sample dataset retrieved for UCI, with numerous variability tests conducted before finalizing to the most fitting one, based on computing and accuracy performance. All dataset attributes are descripted, while many feature selection stages take place where uncorrelated attributes are removed based on the correlation matrix heatmap plotted. Finalizing on the K Nearest Neighbor model, further testing is implemented for comparison of results with noisy data and finally concluding to most optimal configuration achieving great results. Through the observations, K Nearest Neighbor does not seem to be the only suitable high result driven model for selection on the specified dataset, leaving a wide area for further investigation on the topic. Based on the report, machine learning algorithms may have much to offer to the health sector, assisting both patient and doctors' perspective in means of time consumption, costs and practical accessibility.

## 1. Introduction

Nowadays, the power of machines has advanced, more complex problems demand solutions not achievable before. Such demand is met in the health sector, where many yet undiscovered ways of disease treatments and diagnosis could eventually help millions to overcome their complications. A chronic disease that has endorsed such benefits and has still more to come is the Parkinson's Disease (PD) as mentioned in the previous report, which at an early stage identification and correct supply of medication could show significant improvements towards a patient's symptoms (McDermott, 2019). This chronic disease has more obstacles and clues yet to be discovered through algorithms and machine learning as no fully accurate or reliable way is there to determine whether a person suffers from PD.

For this case, the dataset used is based on speech impairments from voice samples from a number of individuals in which a portion suffers from Parkinson's disease. This dataset is retrieved from the UCI database library containing 195 measurements from 31 people, 23 diagnosed with PD. The data includes 24 attributes from which "status" with binary output indicating weather the patient is diagnosed with disease or is clean and "name" are initially dropped off the table, leaving it with another 22 numerical measurement-based features. "status" is saved as our output to be compared with the analogous attributes' performance.

*Table 1 Feature description*

| # | Attribute | Description |
|---|---|---|
| 1 | Fo(Hz) | Average vocal fundamental frequency |
| 2 | Fhi(Hz) | Maximum vocal fundamental frequency |
| 3 | Flo(Hz) | Minimum vocal fundamental frequency |
| 4 | Jitter(%) | Jitter in percentage |
| 5 | Jitter(Abs) | Absolute jitter in microseconds |
| 6 | Jitter:DDP | Difference of differences of periods |
| 7 | RAP | Relative average perturbation |
| 8 | PPQ | Five-point period perturbation quotient |
| 9 | Shimmer | Local shimmer |
| 10 | Shimmer(dB) | Local shimmer in decibels |
| 11 | Shimmer:APQ3 | Three-point amplitude perturbation quotient |

| | | |
|---|---|---|
| 12 | Shimmer:APQ5 | Five-point amplitude perturbation quotient |
| 13 | Shimmer:DDA | Difference of differences of amplitutes |
| 14 | APQ | Amplitude perturbation quotient |
| 15 | NHR | Noise-to-harmonics ratio |
| 16 | HNR | Harmonics-to-noise ratio |
| 17 | RPDE | Recurrence period density entropy |
| 18 | DFA | Detrended fluctuation analysis |
| 19 | spread1 | Nonlinear measure of fundamental frequency |
| 20 | spread2 | Nonlinear measure of fundamental frequency |
| 21 | D2 | Correlation dimension |
| 22 | PPE | Pitch period entropy |

Data mining, gives us the opportunity to manipulate the information retrieved by large databases and further handle it through machine learning algorithms, extracting valuable insights such as predicting an occurrence or classifying a binary problem (Alpayden, 2009). Of course, different ways of approach allow a variety of outputs based on the machine learning application type. Choosing the most fitting type is key for effective and efficient results, as each category has distinct methods of data processing, hence giving different outputs under diverse circumstances.

Intelligent systems can learn from data in two distinct ways, Supervised and Unsupervised learning (Soni, 2018). For Parkinson's dataset, supervised learning will be used as the model consists of an input variable, an output variable and the algorithm is to map their relation. Based on WRIT1 report's approaches, most fitting machine learning algorithms to follow are 'Stacking' with 94.6% and SVM with Radial Kernel (RBF) (Gaussian processes) with 96.9% accuracy on a similar PD dataset for telemonitoring. In this process, Support Vector Machines (SVM) both radial and linear and K Nearest Neighbors classifiers are tested in aspects of performance and results before finalizing on a machine learning process to use.

Once an algorithm is tested and results are retrieved, confusion matrixes are used to describe the performance of each classifier focusing on the predictive capabilities of the model. Initially, to prevent misleading outcome from our models it is necessary to pre-process the data avoiding noisy information.

## 2. Data Pre-processing

Before analyzing our data samples, it is necessary to ensure that the dataset is clear and in well-defined shape so that the algorithms output exact and precise results. If data is not in an understandable format for our machine learning system, errors are likely to appear stopping the process. Unprocessed data may contain aggregated data, errors or irrational values out of bounds, missing values or inconsistencies between values. The pre-process stage of data is to avoid such issues and only use high quality information in order to give a clean and normalized output.

To train and test our dataset, it is first obligatory to understand what the likely obstacles to face are and how to prevent them stopping the algorithms from delivering the desired accurate results. All missing and noisy data, duplicated or aggregated values are likely to affect the algorithms decisions and functions making it unlikely to extract clear and accurate results (Alexandropoulos, 2019).

After empty or duplicated values are removed from the table to prevent error, we move onto feature selection stage. To help distinguish between correlation of data we first make numerical data distribution in order to plot them and visually examine our numbers. This could lead us to a better explanation of the relationship between data possibly giving new insights.

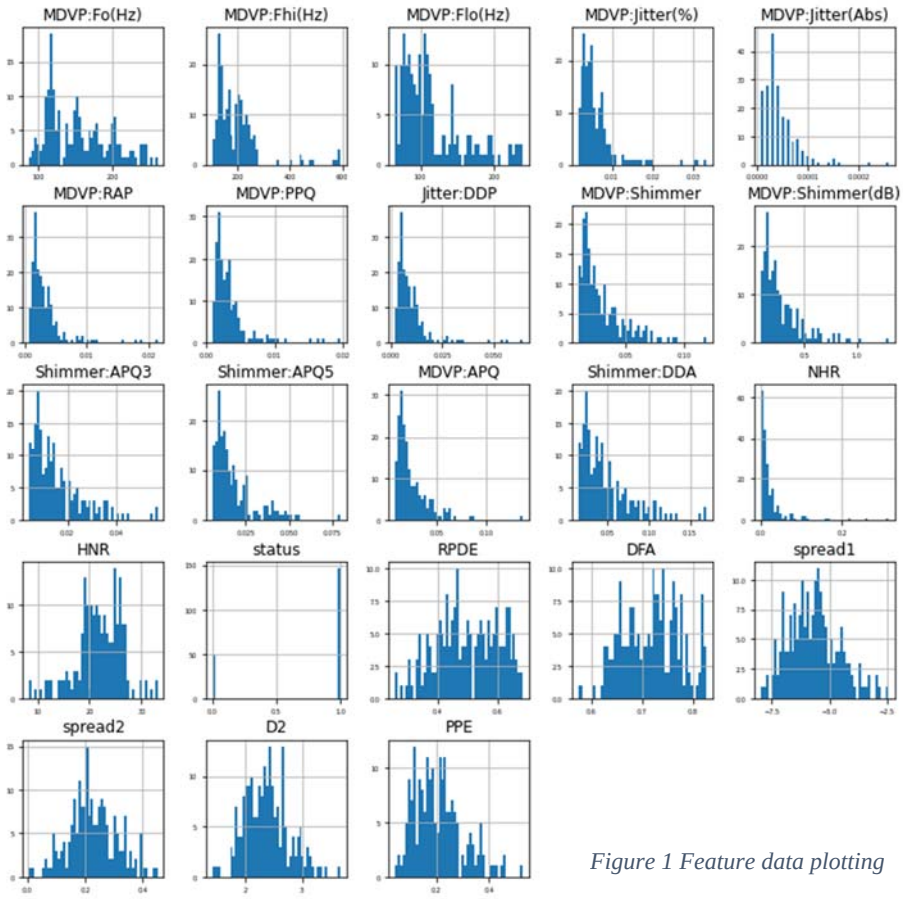



*Figure 1 Feature data plotting*

Through this plotting we can visually determine which of these attributes are having a linear or non-linear data likely to suggest for linear relation with other attributes as well. Other distinguished elements are the bounds which vary with big differences. To better simulate the algorithm, it is best to scale the data setting the mean for all features to 0 with bounds of 1 for lower or higher. This pre-process is also known as data normalization and is commonly used basically for simplifying the numbers in such a table.

```
# Remove mean
data_scaled = preprocessing.scale(X)
print("\nAFTER:")
print("Mean =", data_scaled.mean(axis=0))
print("Std deviation =", data_scaled.std(axis=0))

# Min max scaling
data_scaler_minmax = preprocessing.MinMaxScaler(feature_range=(-1, 1))
data_scaled_minmax = data_scaler_minmax.fit_transform(X)
print("\nMin max scaled data:\n", data_scaled_minmax)
```

*Figure 2 Data normalization*

Now the data has been scaled with mean 0 and saved as an array. In order to be able to further manipulate the data and create the correlation matrix heatmap it is obligatory to convert it back to a 'DataFrame' adding the data back to their correct columns.

```
#array to panda
scaled = pd.DataFrame(data_scaled_minmax, columns=['MDVP:Fo(Hz)',
scaled['status']=y
```

*Figure 3 Table configuration*

As mentioned earlier, 'status' column is the binary answer to weather the individual is diagnosed with PD. It is also added to the 'DataFrame' with the rest of values to print the correlation matrix and find most correlating attributes to it. For the matrix, corelating values below 0.3 and higher than -0.3 are not printed, giving a clear view for which features are best to collect and use within our model testing.

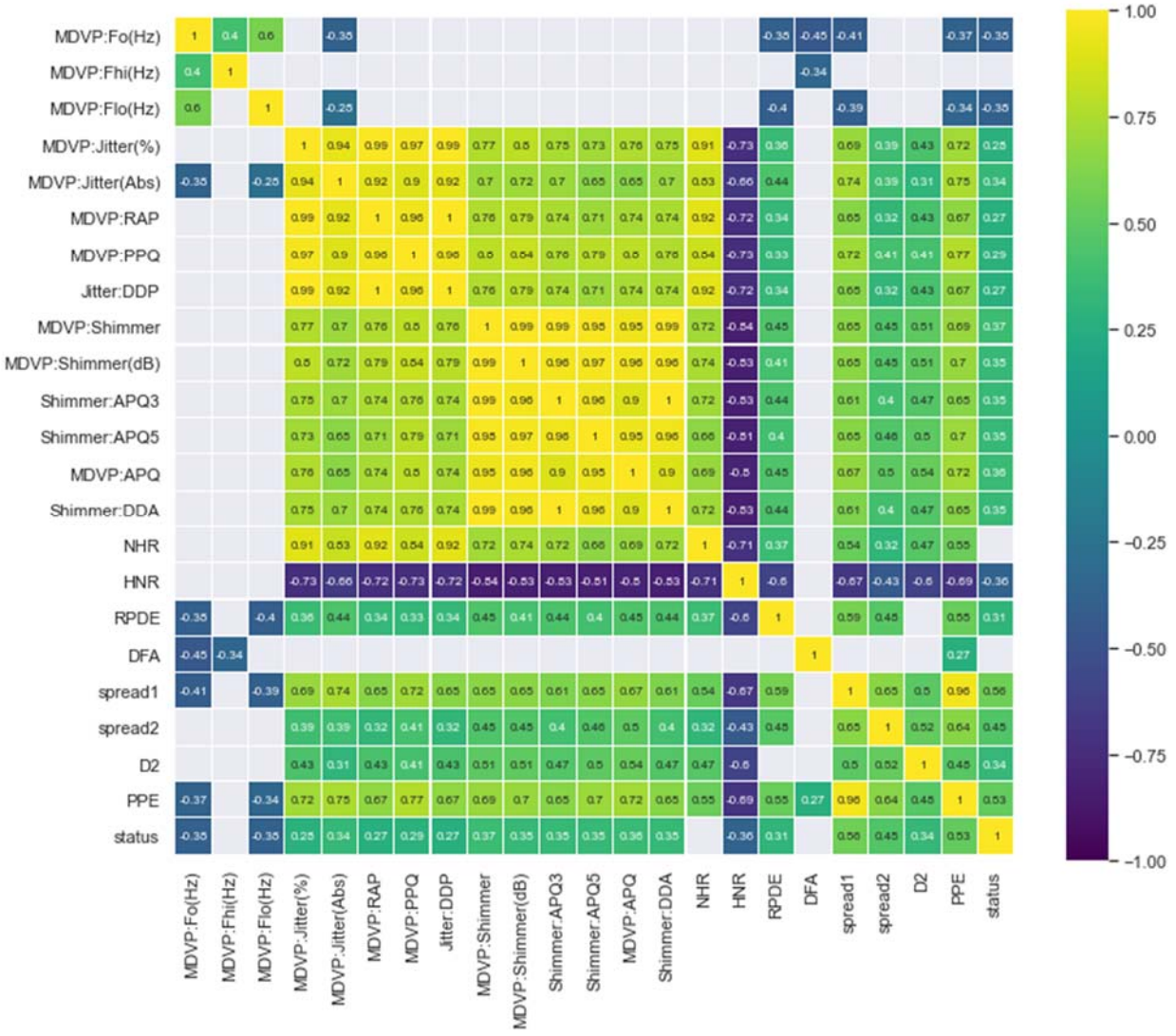



*Figure 4 Correlation matrix heatmap*

Now that the correlation matrix heatmap is visualized, we can visually identify which attributes have very little to no correlation with 'status', making it easy to choose which attributes are to be dropped off the table first. While testing through our models, confusion matrix diagrams are plotted to check and confirm the algorithm's performance between every cycle, continuing to attribute removal stage and noting every outcome into a table.

Data multicollinearity can be dangerous at times, analyzing the correlation matrix many attributes show very high correlation between them, meaning that even though they are viewed as independent variables, they are likely to be codependent of their results, leading to risks when removing attributes. It is usually preferred to remove such features in blocks, avoiding noisy process development.

## 3. Design model performances

To test our models, the process is separated in three complete cycles. As the 195 samples are comparatively few, testing and training split will remain at the optimal 80% / 20% through all stages. For our first cycle, no attribute is initially dropped off the table. Having this in mind, we will be able to compare the results later on, confirming the correlation based heatmap and concluding to how important noisy data extraction is. Furthermore, uncorrelated data does not necessarily mean low accuracy results, the first cycle is also an important step in configuring the algorithms and determining which is best to use for our dataset.

After running each algorithm, two outputs are to be given, an accuracy score based on successful runs of classification and the F1 score. The F1 score is a function of Precision and Recall and is needed when seeking a balance between them. F1 score may be a better measurement when there is an uneven class distribution while it also focuses on true negatives. This is why F1 measurements can indicate even higher results than accuracy (Shung, 2018).

### 3.1 First cycle | No features out of table

*Table 2 First Testing Cycle Results*

| Algorithms | Accuracy | F1 | Split Random Variable | Train / Test Split |
|---|---|---|---|---|
| Gaussian | 82,05 | 74 | 42 | 80/20 |
| Linear Kernel SVM | 82,05 | 74 | 42 | 80/20 |
| Sigmoid Kernel SVM | 82,05 | 74 | 42 | 80/20 |
| Polynomial Kernel SVM | 92,31 | 91,4 | 42 | 80/20 |
| K Nearest Neighbors (5) | 94.87 | 94,5 | 42 | 80/20 |

| Algorithms | Accuracy | F1 | Split Random variable | Acc AVG | F1 AVG |
|---|---|---|---|---|---|
| Gaussian | 82,05 | 73,96 | 17 | 82,05 | 73,96 |
| Linear Kernel SVM | 82,05 | 73,96 | 17 | 82,05 | 73,96 |
| Sigmoid Kernel SVM | 82,05 | 73,96 | 17 | 82,05 | 73,96 |
| Polynomial Kernel SVM | 87,18 | 84,08 | 17 | 89,745 | 87,755 |
| K Nearest Neighbors (5) | 92,31 | 91,43 | 17 | 92,31 | 92,975 |

Having the first cycle completed we can identify the same measurement index of 82,05 accuracy and 73,96 F1 scores throughout Gaussian, Linear and Sigmoid Kernel SVM, in both random train test split variables. This occurrence is suggesting either that the algorithms ran in a very similar manner or that they only support the dataset to an extent, with probable error in the output.

### 3.1.1 Gaussian Linear and Sigmoid SVM algorithms

To support this statement, the plotted confusion matrix shows that no 0 outputs are predicted in the algorithm. This is viewed in all three algorithms in the same way and is most likely occurred by the lack of samples ran within the training split.

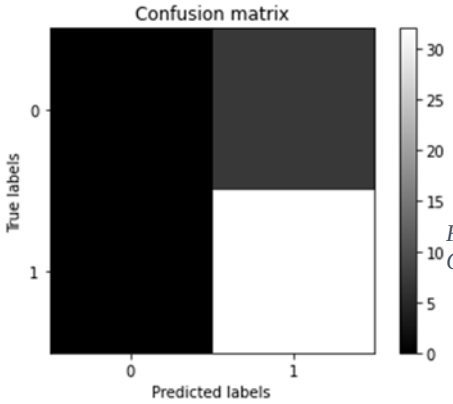



*Figure 5 Cycle 1 Gaussian Linear and Sigmoid SVM Confusion matrix*

### 3.1.2 Polynomial Kernel SVM algorithm

Polynomial Kerner SVM stands with higher accuracy and F1 results suggesting it as a more accurate method to follow. Of course, we cannot be certain of such recommendation without viewing the confusion matrix and completing all three cycles. Through polynomial's confusion matrix plot we can easily distinguish the difference in number of 0 predicted visually, as the color shade has changed.

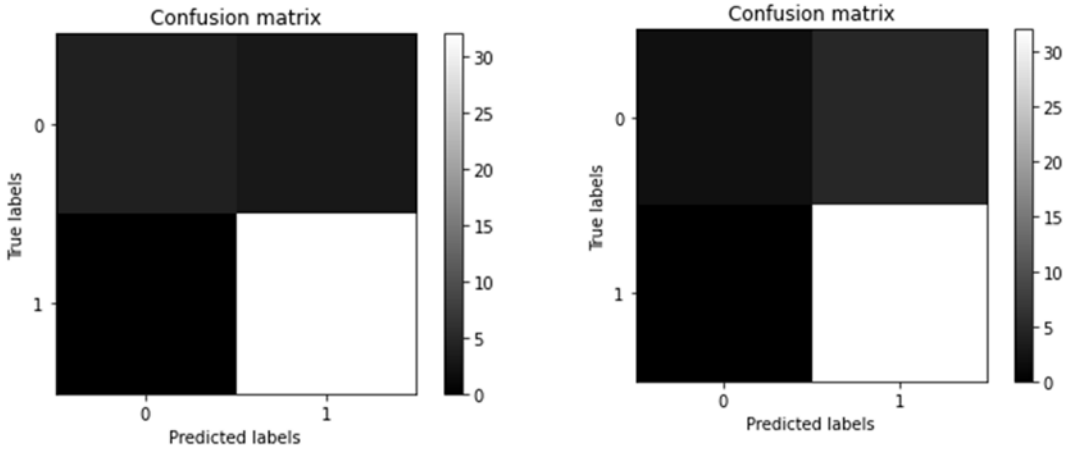



*Figure 6 Cycle 1 Polynomial Kernel SVM Confusion Matrix*

### 3.1.3 K Nearest Neighbors SVM algorithm

At highest placement, K Nearest Neighbours algorithm is met using '5' neighbour checks indicating the number neighbouring values chosen for processing. The average scores for accuracy and F1 are 92,31 and 92,975 respectively. The confusion matrixes is plotted and shown:

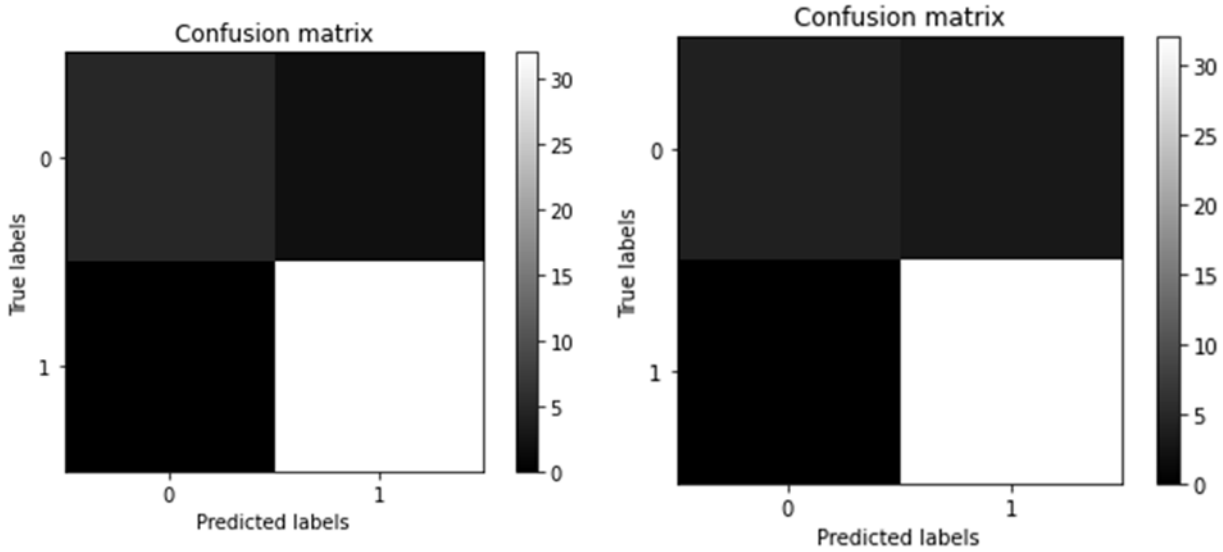



*Figure 7 Cycle 1 K Nearest Neighbors SVM Confusion Matrix*

The desired output is to have is for both diagonal squares from top left to bottom right to have bright color, through such observation we find the plethora of complete predictions on output zero, or individual diagnosed without PD.

## 3.2 Second Cycle | Feature selection stage

The first cycle is complete and all necessary data output is observed. Now the process continues in the same manner keeping the same two random train test split variables, while configuring the models. In the feature selection stage, all attributes with correlation to 'status' scoring under 0.3 are removed from the table. The features dropped off the table are; 'Fhi(Hz)', 'NHR', 'DFA', 'Jitter(%)', 'RAP', 'PPQ', 'Jitter:DDP'.

*Table 3 Second Testing Cycle Results*

| Algorithms | Accuracy | F1 | Split Random Variable |
|---|---|---|---|
| Gaussian | 82,05 | 74 | 42 |
| Linear Kernel SVM | 82,05 | 74 | 42 |
| Sigmoid Kernel SVM | 82,05 | 74 | 42 |
| Polynomial Kernel SVM | 92,31 | 91,4 | 42 |
| K Nearest Neighbors (5) | 94.87 | 94,5 | 42 |

| Train / Test Split |
|---|
| 80/20 |
| 80/20 |
| 80/20 |
| 80/20 |
| 80/20 |

| Algorithms | Accuracy | F1 | Split Random variable | Acc AVG | F1 AVG |
|---|---|---|---|---|---|
| Gaussian | 82,05 | 73,96 | 17 | 82,05 | 73,96 |
| Linear Kernel SVM | 82,05 | 73,96 | 17 | 82,05 | 73,96 |
| Sigmoid Kernel SVM | 82,05 | 73,96 | 17 | 82,05 | 73,96 |
| Polynomial Kernel SVM | 87,18 | 84,08 | 17 | 89,745 | 87,755 |
| K Nearest Neighbors (5) | 92,31 | 91,43 | 17 | 92,31 | 92,975 |

All algorithms and plots show the same result outputs even after 7 out of 22 features are dropped, leaving the current dataset with only 15 of its attributes. This observation, firstly confirms the choice of attribute removal, as multicollinearity between the attributes could have shown lower results, and secondly, it enhances the importance of randomization between test and train splits directly affecting the algorithm capabilities.

## 3.3 Third Cycle | Train Test Split variance

In the third and final cycle, train and test split ratio will be changed testing this different approach to possibly observe higher accuracies. Based on the observation of the matrixes in the first cycle, it was concluded that not enough 0 were predicted. Adding more data to feed the training stage of the algorithm is likely to improve that statement, while adversely, as the testing split is reduced the algorithm has less examples to work on, resulting to low output probabilities. Taking these insights in consideration, the optimal split is set to 75% training / 25% testing. The random test split variable will have no changes.

*Table 4 Third Testing Cycle Results*

| Algorithms | Accuracy | F1 | Split Random Variable | Train / Test Split |
|---|---|---|---|---|
| Gaussian | 77,55 | 67,75 | 42 | 75/25 |
| Linear Kernel SVM | 77,55 | 67,75 | 42 | 75/25 |
| Sigmoid Kernel SVM | 77,55 | 67,75 | 42 | 75/25 |
| Polynomial Kernel SVM | 91,84 | 91,13 | 42 | 75/25 |
| K Nearest Neighbors (5) | 93,88 | 93,51 | 42 | 75/25 |

| Algorithms | Accuracy | F1 | Split Random variable | Acc AVG | F1 AVG |
|---|---|---|---|---|---|
| Gaussian | 83,67 | 76,24 | 17 | 80,61 | 71,995 |
| Linear Kernel SVM | 83,67 | 76,24 | 17 | 80,61 | 71,995 |
| Sigmoid Kernel SVM | 83,67 | 76,24 | 17 | 80,61 | 71,995 |
| Polynomial Kernel SVM | 85,71 | 80,72 | 17 | 88,775 | 85,925 |
| K Nearest Neighbors (5) | 95,92 | 95,68 | 17 | 94,9 | 94,595 |

Observing the tables, we can see the impact the test and train split variance has made in the algorithms' results. The Gaussian, Linear and Sigmoid SVM algorithms remain with equaling outputs between them but with lower average scores to previous cycles, confirming the first observation at which the data manipulation methods of each are stated. This also suggests that the bigger train split did not help for better results in the particular algorithmic process. Polynomial Kernel SVM was also affected by the split variance lowering its accuracy results. Lastly but adversely, K Nearest Neighbors learning algorithm sustains the first placement of accuracy results averaging an even greater score than met in previous cycles.

### 3.3.1 Gaussian Linear and Sigmoid SVM algorithms

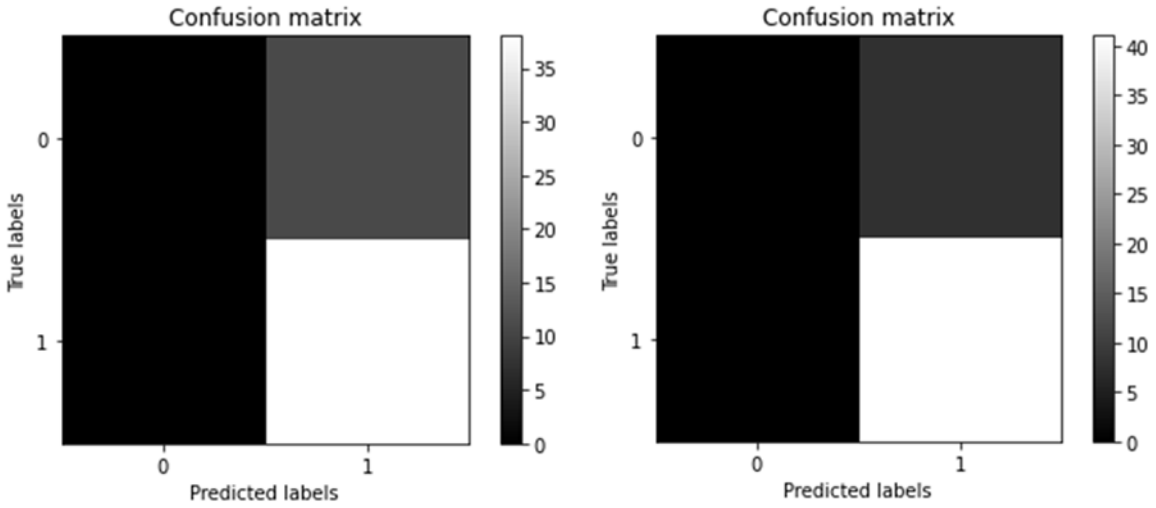



*Figure 8 Cycle 3 Gaussian Linear and Sigmoid SVM Confusion matrix*

As expected, the confusion matrixes for the Gaussian, Linear and Sigmoid SVM algorithms show poor results, even after the training split was given more data to train, the results do not show any improvement for 'status' 0 prediction.

### 3.3.2 Polynomial Kerner SVM algorithm

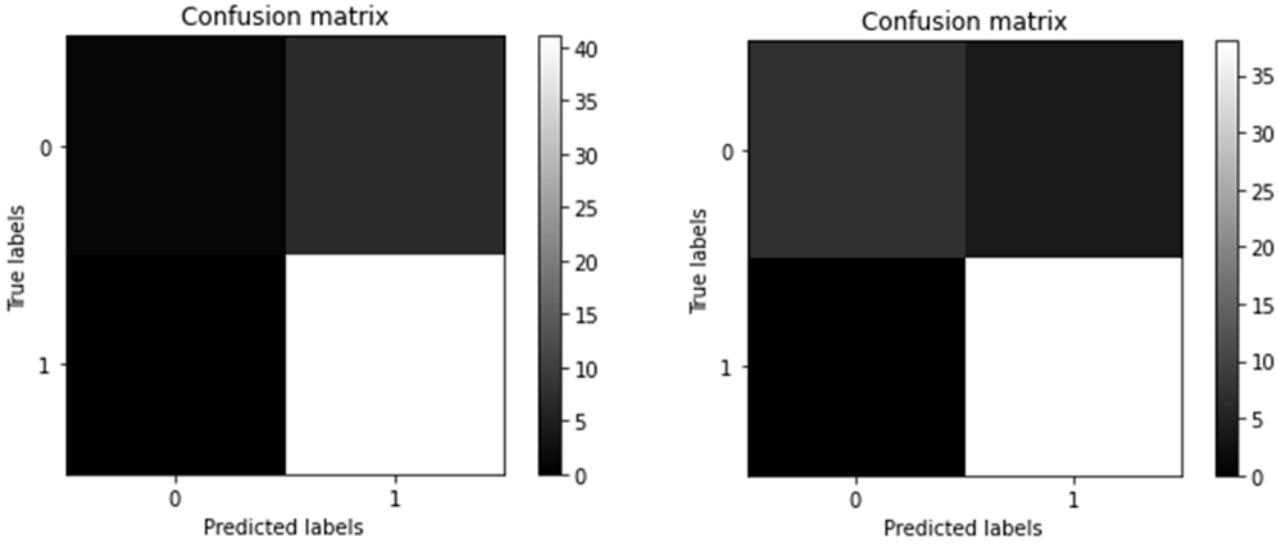



*Figure 9 Cycle 3 Polynomial Kernel SVM Confusion Matrix*

The same occurrence is seen in the confusion matrixes in the Polynomial Kernel SVM algorithm. As the training split magnification lowered the accuracy results, the successful predictions are relatively lower.

### 3.3.3 K Nearest Neighbors algorithm

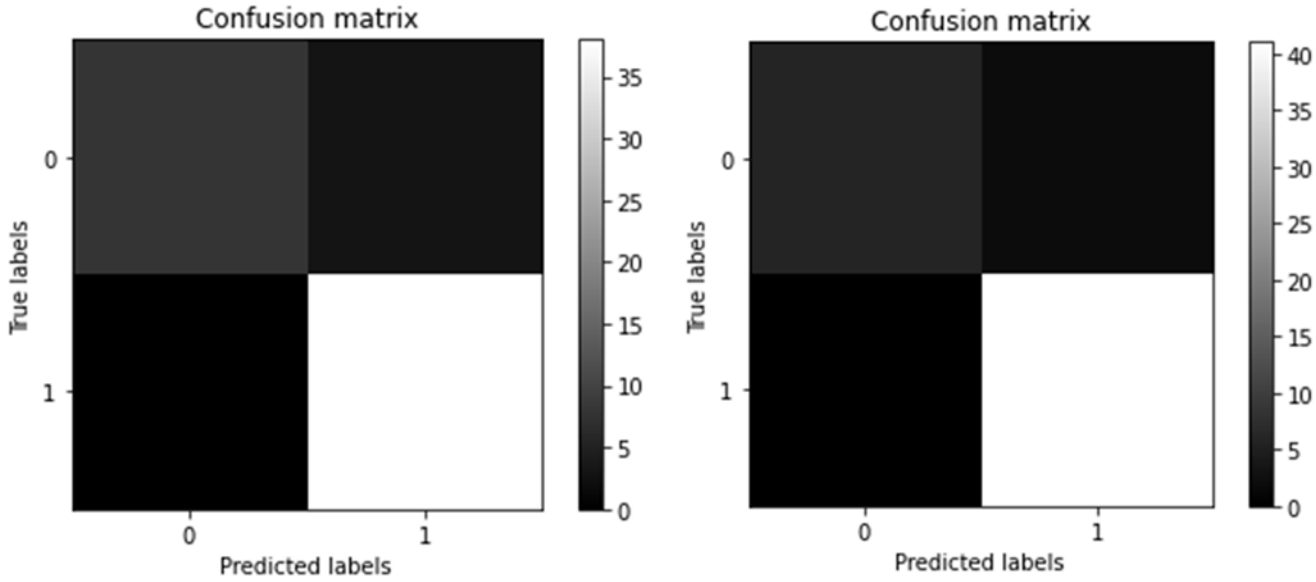



*Figure 10 Cycle 3 K Nearest Neighbors confusion matrix*

Inversely for K Nearest Neighbors, the average results scored even higher that previous runs, suggesting that a bigger training dataset can ultimately lead an algorithm to better results. Furthermore, the plots confirm the number of correct predictions.

## 3.4 Models' performance conclusion and model selection

The model selection is based on the overall performance of the learning process, including their costs and their result accuracy. The computing power needed to process this Parkinson's voice sample dataset is very minimal, as it is a comparatively small table. This makes it difficult to compare between computing performance between the algorithms, but also unnecessary. Ultimately, based on the accuracy and F1 results, the K Nearest Neighbor design model learning algorithm is chosen for further configuration. Another advantage of this model is the accessible parameter configuration that can be tested before concluding to its best average outcome achieved. In the next section, hyperparameter tuning on the Parkinson's dataset using the K Nearest Neighbors machine learning model is further implemented.

# 4. K Nearest Neighbors model configuration & results

Hyperparameters express important properties of algorithmic models such as how fast the process should learn or its complexity level. Configuring the parameters of K Nearest Neighbors model is essential to find the most optimal way to run it and obtain best results. In the specific algorithm, very important role for complexity level has the choice of 'Neighbors', which variation can achieve big changes towards performance. Other ways for tuning the model are through the testing and training split, and even further attribute elimination. Lastly, in order to define the most optimal value of 'K' – to choose how many clusters the data should be divided into – elbow method is used via grid search process to finalize the results.

## 4.1 Feature selection stage

Firstly, due to the lack of changes after the first attribute selection stage, further selection for elimination is executed and the algorithm is tested. Through the correlation matrix heatmap, three attributes scoring below 0.35 correlation with 'status' are; 'Jitter(Abs)', 'RPDE' and 'D2' also dropped off the table. Feature elimination will continue until lower algorithm performance is shown, giving clear insight to which attributes are noisy and which highly correlated. For testing and comparing with previous results, no other variables are configured.

*Table 5 Final design testing results*

| Previously highest measurements | | | |
|---|---|---|---|
| Algorithms | Accuracy | F1 | Split Random Variable |
| K Nearest Neighbors (5) | 93,88 | 93,51 | 42 |
| Further Feature selection (Jitter(Abs), RPDE, D2) < 0.35 | | | |
| K Nearest Neighbors (5) | 95,92 | 95,77 | 42 |
| Further Feature selection (Fo(Hz), Flo(Hz), Shimmer(dB), APQ3, APQ5, Shimmer(DDA)) = 0.35 | | | |
| K Nearest Neighbors (5) | 89,8 | 88,61 | 42 |

| Previously highest measurements | | | |
|---|---|---|---|
| Algorithms | Accuracy | F1 | Split Random Variable |
| K Nearest Neighbors (5) | 95,92 | 95,68 | 17 |
| Further Feature selection (Jitter(Abs), RPDE, D2) < 0.35 | | | |
| K Nearest Neighbors (5) | 95,92 | 95,68 | 17 |
| Further Feature selection (Fo(Hz), Flo(Hz), Shimmer(dB), APQ3, APQ5, Shimmer(DDA)) = 0.35 | | | |
| K Nearest Neighbors (5) | 93,88 | 93,28 | 17 |

| Further Feature elimination | Acc AVG | F1 AVG |
|---|---|---|
| None | 94,9 | 94,595 |
| Jitter(Abs)',' RPDE', 'D2' < 0.35 | 95,92 | 95,725 |
| Fo(Hz), Flo(Hz), Shimmer(dB), APQ3, APQ5, Shimmer(DDA) = 0.35 | 91,84 | 90,945 |

The retrieved results, show successful noisy data elimination on the first feature selection with outperforming accuracy of previous alterations. Furthermore, it suggests that in the second elimination, likely correlated data were removed scoring lower accuracy and hence they will be put back to the table before continuing the tuning process.

## 4.2 Train and test split variation

Based on the previously done train and test split variation, we observe a well-defined algorithmic process between the split separation for 80 / 20 and 75 / 25. For this reason, the precise test split numbers to be test are between 0.23 and 0.29. For testing and comparing with previous results, no other variables are configured.

*Table 6 Final design variation results*

| | Split Random Variable | | | | | |
|---|---|---|---|---|---|---|
| Test size | 69 | 69 | 27 | 27 | Accuracy | F1 |
| 0.23 | 86,67 | 86,67 | 91,11 | 91,38 | 88,89 | 89,025 |
| 0.24 | 87,23 | 87,23 | 87,23 | 87,23 | 87,23 | 87,23 |
| 0.26 | 88,24 | 88,24 | 88,24 | 88,24 | 88,24 | 88,24 |
| 0.27 | 88,68 | 88,68 | 88,68 | 89,01 | 88,68 | 88,845 |
| 0.28 | 89,09 | 89,09 | 89,09 | 89,42 | 89,09 | 89,255 |
| 0.29 | 87,72 | 87,9 | 89,47 | 89,8 | 88,595 | 88,85 |
| | Accuracy | F1 | Accuracy | F1 | | |

Through this testing, it is clearly shown once again how important the training and split variance is for the algorithm, as huge impact is shown in the results. Nevertheless, there is no linear correlation of the values, and the outputs seem to vary unreasonably to the split changes. Due to this observation, no testing / training split will be fixed, and further test will be optimized in both 75/25 and 80/20 split variances.

## 4.3 Choice of Neighbors

As mentioned, to further tune the complexity of the chosen algorithm, configuration based on the neighbor number will be tested, observing important information to find most optimal methods for best accuracy outputs.

*Table 7 Final design cluster testing results*

| Number of Neighbors | 80/20 Split | | 75/25 Split | | Accuracy | F1 |
|---|---|---|---|---|---|---|
| 1 | 87,18 | 86,79 | 89,9 | 89,52 | 88,54 | 88,155 |
| 2 | 89,74 | 90,22 | 91,84 | 92,19 | 90,79 | 91,205 |
| 3 | 92,31 | 92,07 | 93,88 | 93,71 | 93,095 | 92,89 |
| 4 | 97,44 | 97,5 | 97,96 | 98,01 | 97,7 | 97,755 |
| 5 | 92,31 | 92,07 | 95,92 | 95,68 | 94,115 | 93,875 |
| 6 | 92,31 | 92,07 | 95,92 | 95,68 | 94,115 | 93,555 |
| | Accuracy | F1 | Accuracy | F1 | | |

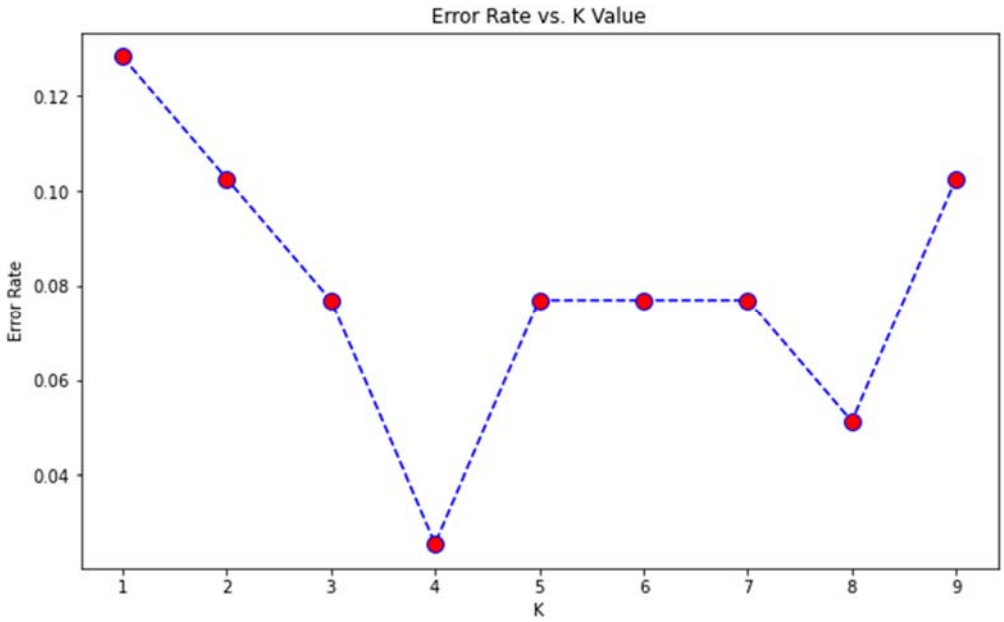



*Figure 11 Optimal K value through elbow method*

Throughout this testing, very interesting results are observed. Firstly, we can identify that 4 Neighbors output the highest results in both splitting cases with big averaging difference toward other Neighbor picking. Furthermore, results after 4 do give very similar outputs confirming the choice of '4' as the optimal number of clusters for the data to be divided into based on accuracy. These results are further validated through the analysis of error rate based on the elbow method for justification (Doumbia, 2019).

## 5. Conclusion

In this report, five different algorithmic methods have been tested on the same dataset in various stages of complexity, with aim towards an accurate classification and finite determination of individuals being diagnosed of PD. Before the algorithmic operation, data pre-processing methods take place, normalizing the data to best fit the table. Duplicate and missing values are also eliminated to prevent errors in the results. Throughout the process, the models were tested using different training and testing splits, with various randomizing variables for gathering multiple outputs and averaging them for more accurate results. Moreover, multiple feature selection stages based on the correlation matrix heatmap took place, giving valuable insights to finding which parts of the table are noisy. Identifying uncorrelated features allowed further testing of performance under noisy conditions to compare efficiency of the final chosen algorithm.

Throughout the stages, Polynomial Kernel SVM algorithm is also identified with high accuracy results nearly touching 90% performance, leading to a possible alternative model to test out in the future based on this dataset. With further hyperparameter tuning on the specified model, it is likely to expect higher results or even surpass the current report's final design choice.

K Nearest Neighbor seemed to be the most suitable model for final design, giving the highest accuracy and F1 results overall. This design went through hyperparameter tuning with over 10 variability identical runs, giving insights for most optimal number of clusters to divide the data into. Finally, as tuning process is completed, accuracy results up to 97.7% of individuals' classification to weather they have PD are observed and further validated with grid search. These results indicate the need of further investigation on the specific topic, as such classification and prediction methods can become technological tools to assist doctors and individuals, saving them time and costs with more accurate than previously used diagnosis practices.

## 6. References

Alexandropoulos, K. S. V., 2019. Data preprocessing in predictive data mining. *Knowledge Engineering Review,* Volume 34.
Alpayden, E., 2009. *Introduction to machine learning.* s.l.:s.n.
Despotovic, V., Skovranek, T. & Schommer, C., 2020. *Speech Based Estimation of Parkinson's Disease Using Gaussian Processes and Automatic Relevance Determination,* https://www.sciencedirect.com/science/article/pii/S0925231220304318: ScienceDirect.
Doumbia, M., 2019. *Medium.* [Online]
Available at: https://medium.com/@moussadoumbia_90919/elbow-method-in-supervised-learning-optimal-k-value-99d425f229e7
[Accessed 2021].
Elkouzi, D. A., 2019. *Parkinson's Foundation.* [Online]
Available at: https://www.parkinson.org/understanding-parkinsons/what-is-parkinsons
[Accessed 12 2020].
GEN Corporation, 2020. *GEN - Genetic Engineering and Biotechnology News.* [Online]
Available at: https://www.genengnews.com/news/advancing-parkinsons-disease-diagnosis-with-machine-learning/
[Accessed 12 2020].
Little, M. A., McSharry, P. E., Hunter, E. J. & Ramig, L. O., 2008. *UCI, Parkinsons Dataset, Suitability of dysphonia measurements for telemonitoring of Parkinson's disease.* [Online]
Available at: https://archive.ics.uci.edu/ml/datasets/Parkinsons
[Accessed 2020].
McDermott, A., 2019. *healthline.* [Online]
Available at: https://www.healthline.com/health/parkinsons/early-onset?c=767899079616
[Accessed 2020].
Physiopedia, 2020. *Physiopedia.* [Online]
Available at: https://www.physio-pedia.com/Parkinson%27s_Disease:_A_Case_Study
[Accessed 12 2020].
Ray, S., 2019. A PREDICTIVE DIAGNOSIS FOR PARKINSON'S DISEASE THROUGH MACHINE LEARNING. *THE CANADIAN SCIENCE FAIR JOURNAL.*
Samal, D., 2020. *Kraggle.* [Online]
Available at: https://www.kaggle.com/debasisdotcom/parkinson-disease-detection
[Accessed 2020].
Shung, K. P., 2018. *Towards Data Science.* [Online]
Available at: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9
[Accessed 1 2021].
Soni, D., 2018. *Towards Data Science.* [Online]
Available at: https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d
Triarhou, L., 2013. *Dopamine and Parkinson's Disease. In Madame curie bioscience database.,* s.l.: s.n.
UCI Corporation, 2009. *UCI Machine Learning Repository: Data Sets.* [Online]
Available at: https://archive.ics.uci.edu/ml/datasets.php
[Accessed 12 2020].