

**C O V E N T R Y**  
**U N I V E R S I T Y**

Faculty of Engineering, Environment and Computing  
School of Computing, Electronics and Mathematics



MSc Data Science  
7150CEM  
**DATA SCIENCE PROJECT**

**Detection of Parkinson's Disease**  
**using**  
**Machine Learning algorithms**

Author: Glen T John  
SID: 11980816

Supervisor: David Croft

Submitted in partial fulfilment of the requirements for the Degree of Master of Science in Data Science  
Academic Year: 2022/2023

## Declaration of Originality

I declare that this project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet etc.) has been acknowledged by citation within the main report to an item in the References or Bibliography lists. I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.

## Statement of copyright

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to Coventry University. Support, including funding, is available to commercialize products and services developed by staff and students. Any revenue that is generated is split with the inventor/s of the product or service. For further information please see [www.coventry.ac.uk/ipr](http://www.coventry.ac.uk/ipr) or contact [ipr@coventry.ac.uk](mailto:ipr@coventry.ac.uk).

## Statement of ethical engagement

I declare that a proposal for this project has been submitted to the Coventry University ethics monitoring website (<https://ethics.coventry.ac.uk/>) and that the application number is listed below (Note: Projects without an ethical application number will be rejected for marking)

Signed: GLEN T JOHN

Date:09/12/2022

Please complete all fields.

First Name:	Glen T
Last Name:	John
Student ID number	11980816
Ethics Application Number	P142574
1 <sup>st</sup> Supervisor Name	David Croft
2 <sup>nd</sup> Supervisor Name	Derrick Newton

**This form must be completed, scanned and included with your project submission to Turnitin. Failure to append these declarations may result in your project being rejected for marking.**

## **Abstract**

Parkinson's disease is a neurological disease where the brain cells are not able to produce enough dopamine in the body. The brain's dopamine-producing cells are what provide movement control, adaptability, and fluidity. Early detection of Parkinson's is very important for early management as more than 60% of the dopaminergic neurons have already been destroyed by the time clinical symptoms show up. In this paper, we use the biomedical voice measurements measured in frequency of 31 people with 23 people having Parkinson's disease (PD) and each row corresponds to one of the 195 voice recordings that these individuals produced. The main objective of the data is to identify between healthy people and those with PD, as shown by the "status" column, which again is set to 0 for healthy and 1 for PD. We will be classifying these voice recordings using Support Vector Machines (SVM), Naïve Bayes, and Random Forest. Support Vector Machines performed the best showing an accuracy of 87%, while Naïve Bayes and Random Forest showed an accuracy of 61% and 82%. We infer that with the help of voice recordings we can predict Parkinson's disease.

**Keywords – Parkinson's Disease , Machine Learning ,Support Vector Machine(SVM) ,Naïve Bayes ,Random Forest**

## Table of Contents

Abstract.....	2
Acknowledgements.....	5
1 Introduction .....	6
1.1 Voice Frequencies .....	7
1.2 Overview of the Project .....	8
2 Literature Review .....	9
2.1 Paper 1 .....	9
2.2 Paper-2.....	9
2.3 Paper-3.....	10
2.4 Paper – 4 .....	11
2.5 Paper – 5 .....	11
2.6 Paper-6.....	12
2.7 Paper – 7 .....	12
2.8 Paper – 8 .....	13
2.9 Paper – 9 .....	14
2.10 Paper -10.....	14
2.11 Paper -11.....	14
2.12 Paper -12.....	15
2.13 Paper -13.....	16
3 Methodology.....	18
3.1 Algorithms.....	18
3.1.1 Support Vector Machine (SVM) algorithm.....	18
3.1.2 Naïve Bayes Algorithm .....	20
3.1.3 Random Forest.....	21
3.2 Exploratory Data Analysis (EDA) .....	23
4 Implementation .....	26
4.1 Exploratory Data Analysis .....	28
4.1.1 Data Pre-processing .....	28
4.1.2 Exploratory Data Analysis .....	31
5 Testing.....	37
5.1 Model Analysis.....	40
5.1.1 Support Vector Machine .....	42

5.1.2	Naïve Bayes Model.....	44
5.1.3	Random Forest.....	45
6	Project Management .....	47
7	Conclusions .....	48
7.1	Future Works .....	49
	Bibliography and References .....	50
	Appendix A – Project Specification .....	60
	Appendix B– Ethics Certificate .....	63
	Appendix C– Images.....	64

## **Acknowledgements**

I want to express my gratitude to Mr. David Croft, my supervisor, for his time and work on the project. The conversations and talks were crucial in inspiring me to think critically and from a range of perspectives in order to build an in-depth and impartial criticism. I also like to thank Rochelle Sassman, the module leader, for her constant guidance. I also want to express my gratitude to Coventry University for giving me the chance to study at this institution. Also, a huge thank you to my parents, who made enormous sacrifices for me and were very supportive of me during my education. I also want to express my gratitude to my friends for their assistance with the final result.

# 1 Introduction

Parkinson's disease affects the neurological system as well as the body components that are under the control of the nervous system. With time, this illness gets worse. symptoms that appear gradually. A little tremor in one hand might serve as the early warning sign. Along with the typical tremors, the illness can also induce stiffness or sluggishness in movement.

Parkinson's illness may first make it challenging for us to communicate. It's possible that we won't swing your arms as you move. You could start to stutter or talk softly. As your health deteriorates, the symptoms of Parkinson's disease get worse over time.

Regardless of the fact that there is no known cure for Parkinson's disease, a variety of drugs can greatly reduce your symptoms. Sometimes, your doctor may suggest a procedure to address your issues and control certain brain areas.

Machine learning (ML), a collection of multivariate analytical techniques that identify patterns in data, categorize the data, and learn from the data, is extensively used in big data and artificial intelligence. There are both supervised and unsupervised machine learning analyses. Supervised machine learning uses human-labeled training data for data segmentation (which recognises statistical properties on its own and does not utilise training data)

The ability of machine learning (ML) approaches to recognize complicated data patterns, automate data processing, and draw conclusions about the data of specific patients may make its application in targeted therapy for Parkinson's disease (PD) useful. Machine learning has been more prevalent recently in Parkinson's disease diagnosis.

## Symptoms of Parkinson Disease

Parkinson's disease may show in many different ways, with several potential signs and symptoms that each person may experience differently. The most typical signs include

- 1.tremor
- 2.Rigidity (stiffness)
- 3.Slowness of movement
- 4.mild memory and thinking problems

5.Sleep problems

6.Pain

7.Mental health problems ,including anxiety and depression

Motor and non-motor symptoms are two categories of Parkinson's symptoms. Motor symptoms have an influence on your mobility. Tremor, stiffness, and cramping are a few of them.

Other people might not be able to perceive how non-motor symptoms affect you in addition to how they affect your ability to move. They include pain, sleep issues, and issues with one's mental health.

Each individuals experience symptoms uniquely, as does the order in which they emerge and the manner in which they progress.

### **Speech and Voice Disorders in Parkinson's**

According to research, 89 percent of persons with Parkinson's disease (PD) have trouble speaking and breathing, which can lead to wheezing and a low, droning voice. People with PD claim that they are less inclined to talk or feel comfortable in social situations than healthy people in their age group. Speech difficulties can gradually lower a person's quality of life if they have Parkinson's disease. The sooner a person receives speech therapy and a baseline speech evaluation, the more likely it is that they will be able to maintain their ability to speak as the illness progresses. Communication is essential for the quality of life, a positive sense of self, and confidence in people with Parkinson's disease. Voice impairment in PD is a change in the sensory processing that impacts speech. It is believed that people with PD could be unaware of how quiet and difficult to understand their speech has become. Even when listeners believe they are speaking appropriately, people in this situation typically feel like they are shouting when asked to raise their voices to normal volume.

#### **1.1 Voice Frequencies**

Although the human ear can detect frequencies as high as 20,000 Hz (20 kHz), it is most sensitive between 250 and 5,000 Hz. A typical adult male's conversational fundamental frequency ranges from 80 to 180 Hz, whereas a typical adult female's ranges from 165 to 255 Hz. Greater jitter, a lower harmonic/noise (H/N) ratio, less change in phrase frequency and



intensity, and a smaller phonological range were all seen in PD patients. Along with low voice intensity, mono-pitch, voice arrests, and struggle, they commonly reported feeling these things. These qualities don't seem to be affected by the length or severity of the condition.

## **1.2 Overview of the Project**

We will analyse the Parkinson's disease dataset in this project, which is accessible in the UCI machine learning repository. The collection has 23 numbers of characteristics and 197 occurrences. 31 people's biological voice measures are used, 23 of whom have Parkinson's disease (PD). Each row in the data corresponds to one of the 195 patient recordings, and each column in the data represents a specific voice measure. Each patient has six recordings, and the first column lists the patient's name. The local computer downloads the dataset. We read the file in the Jupyter notebook with the aid of Pandas. After that, the data undergoes data pre-processing, which involves cleaning the data, eliminating duplicate rows, and noise reduction. The pre-processed data is available for exploratory data analysis (EDA), which identifies outliers and reveals data distribution and attribute association. The data is prepared for model analysis after the exploratory data analysis. SVM (Support Vector Machine), Naive Bayes, and Random Forest are the models we used for this project. The Sklearn package is used to import these Models. In the end, the data is divided into training and testing data. The accuracy is determined once these data are put into the models. Finally, the best model is chosen once these accuracy levels have been compared.

## 2 Literature Review

### 2.1 Paper 1

In the first study, early Parkinson's disease is accurately diagnosed using machine learning-based multimodal features. In this essay, the premotor stages of Parkinson's disease—which come before the conventional motor stage—are covered. Some of the indications include olfactory loss, sleep behavioural disorder (SBD), and rapid eye movement (REM). These measurements were taken from 401 individuals with early PD and 183 healthy patients with normal PD. The techniques used were Naive Baye, Support Vector Machine (SVM), Boosted Trees, and Random Forest classifiers. SVM classification model was shown to perform best, with 96% accuracy.

**Conclusion-** We note that using SVM resulted in an almost flawless categorization. In order to increase accuracy in this study, we make use of additional significant aspects that are relevant to imaging and non-motor indicators. Additionally, when compared to similar works, our categorization performance is greater. We conclude from the study that a combination of non-motor, CSF, and dopaminergic imaging data may aid in the diagnosis of early PD by being able to distinguish it from healthy normal.

### 2.2 Paper-2

According to this study, Parkinson's disease starts many years before the motor symptoms that manifest as soon as the disease does. They included feature selection and classification processes in their suggested diagnosis. Parkinson's patients were categorized using regression trees, artificial neural networks, and support vector machines. When compared to other methods, Support vector machines and Recursive Feature Elimination fared the best. They were 93.84% accurate, whereas Artificial Neural Network performed at 91.54% accuracy and Classification and Regression Trees were 90.76% accurate.

**Conclusion -** The results demonstrate the significant advantages of integrating FS approaches with classification techniques, particularly when working with voice data, which may contain hundreds of phonetic characteristics. PD may be correctly diagnosed in its early stages with the help of the created early diagnosis technique, and the deterioration of the disease's symptoms can be stopped.

### 2.3 Paper-3

This study used acoustic speech analysis to investigate the usage of dopamine medicines in Parkinson's disease patients. They looked at the voice traits of 28 controls who fit the patients' ages and sexes and 41 patients with Parkinson's disease. The individuals with Parkinson's disease who performed laryngoscopy and fibroscopy were found to have higher jitter, a lower harmonic/noise ratio, fewer frequency and intensity variation, and a narrower phonation range.

#### Dopamine Effects

Parkinson's disease is a neurological condition that impairs a person's ability to move and their mental well-being. One neurotransmitter is dopamine. Your neurological system uses it to transmit information between nerve cells. For this reason, it is referred to as a chemical messenger. Dopamine contributes to our ability to experience pleasure and is crucial to our outstanding capacity for thought and planning. It motivates us to put in extra effort, focus, and find things interesting. Low amounts of dopamine might cause movement problems because they can change the nigrostriatal pathway and result in abnormal nerve firing patterns. Most Parkinson's disease patients appear to have lost 60 to 80% or more of the dopamine-producing cells in their substantia nigra by the time symptoms begin to appear.

**Conclusion** - In conclusion, the results of the current study indicate that compared to controls, those with PD showed greater jitter, a lower H/N ratio, less frequency and intensity variability in the sentence, and lower overall signal strength. Higher incidence of low voice, and phonation range struggle, difficulty, monopitch, and intensity. Although patients with PD reported a high frequency of

Laryngeal examination is frequently used to confirm voice tremors that aren't proven and vocal fold tremors. The length and severity of the sickness don't seem to have any impact on many of these characteristics. However, given the small sample size, lack of intersubjectivity validity, and measurement dependability, these results should be regarded with caution (27,28). (27,28). Early voice treatment has been said to help with various voice issues.

## 2.4 Paper – 4

In this study, data from national health screening programme are used to predict Parkinson's disease risk using machine learning. They analyzed data for neuroimaging and movement analysis. The dataset was taken from National Health Insurance Service-Health Screening. ML algorithms used for the prediction are Neural Network, gradient boosting machines and random forests. The best area under the curve was for Neural network 0.779. They also found out that the model performed the best was with men than in women. Body mass index, total cholesterol, glucose, hemoglobin, and blood pressure levels were the most crucial variables for predicting the incidence of Parkinson's disease. Additionally, there was a strong correlation between men's alcohol and tobacco use, socioeconomic position, and physical activity. A verified result was provided by the ML algorithms.

**Conclusion** - Body mass index, total cholesterol, glucose, hemoglobin, and blood pressure levels all had a significant role in predicting the likelihood of developing Parkinson's disease. Use of tobacco, alcohol, socioeconomic level, physical activity, diabetes in men, and mellitus in women were all strongly associated with the onset of Parkinson's disease (PD). Widely adaptable, producing findings that can be confirmed, the anticipated health-screening dataset-based PD prediction model using ML algorithms might be a good choice for PD prediction models.

## 2.5 Paper – 5

In this study, 75 participants—50 with Parkinson's disease and 25 healthy individuals—completed 14 neurocognitive functional tests, such as tests of memory, speech, executive function, and functional mobility. In this study, subjective and sensor-based neurocognitive skills are compared. Decision tree machine learning methods were employed for the analysis. The model correctly identified early and late stages of PD with 73.7% accuracy and sensor-based features with 92.6% accuracy. Early-stage Parkinson's disease was shown to have considerable reported impairments in memory and executive function, with scores that were 21.6% lower than sensor-based values. On the other hand, those in advanced stages had stronger perceptions (1.57x) of executive and behavioural functions than people in early-stage groups.

**Conclusion** - ML offers a more efficient way of examining and comprehending data patterns. The use of classification algorithms may also make additional components apparent that widen

the use of digital functional assessment versions. This knowledge may be used to assess and update standardised methodologies and scales, which can result in new digital health systems and disease monitoring. Neurodegenerative disorders include Alzheimer's disease, ALS, multiple sclerosis, Huntington's disease, and other kinds of dementia.

## 2.6 Paper-6

Parkinson's disease (PD) patients' movement analysis is used diagnostically with inertial measurement units (IMU), providing a methodical way to evaluate gait and motion parameters. Timed Up and Go (TUG) is a standardized clinical gait test that is frequently used to track patient fall risk and disease development. Home gait tests have been included in movement monitoring methods, allowing a connection to clinically supervised reference evaluations. TUG test frame detection is done using machine learning algorithms like Support Vector Machine (SVM), Random Forest (RF), and Naive Bayes Classifier (NBC), which are then used to classify the data. 300 actual TUG test samples from 32 PD patients were collected over the course of 81 clinical visits and 96 daily recordings of real-world gait data. In order to automatically classify continuous real-world gait data, RF was used, yielding prefiltering sensitivity of 98.6%, precision of 90.6%, recall of 88.5%, and F1-score of 89.6% for TUG test detection.

**Conclusion** - This is the first study to apply machine learning algorithms and only one IMU foot sensor to evaluate an algorithm for the detection of unsupervised standardized TUG tests within real-world data. As a result, they have made a substantial contribution to the automated and transparent identification of unsupervised standardized TUG testing in a real-world setting, encouraging the ease of software interaction during TUG test execution. Additionally, uTUG can aid medical professionals in movement analysis and offer clinical perceptions into neurological disorders.

## 2.7 Paper – 7

With combining an ensemble learning approach with the ability to train online from huge clinical datasets, we hope to give a combined method for detecting Parkinson's disease (PD) in this work. The approach is constructed using the Deep Belief Network (DBN) and Neuro-Fuzzy methods. The Anticipation (EM) clustering approach can handle large datasets. The Principle Component Analysis (PCA) approach is used to eliminate noise from the data. The UPDRS prediction models are designed with the intention of diagnosing PD. K-NN is the

proposed approach for handling missing data. We use incremental machine learning methods to make the provided methodology more successful. Using a real-world PD dataset, we test our method, and the outcomes are compared to those of other PD detection methods developed using machine learning methods.

**Conclusion** Additionally, EM's production of 13 segments from PD data gave the greatest criterion value, and EM's 13 clusters produced the best clustering results (275755.9052). The EM data for 13 clusters are displayed in Fig. 6. This image displays the distribution of Total-UPDRS and Motor-UPDRS among 13 clusters of EM using PC1 created by PCA. For easy viewing, PD feature findings were divided into two parts.

## 2.8 Paper – 8

This research describes a novel method for automated Parkinson's disease detection from EEG data using a flexible analytic wavelet transform (FAWT). Prior to applying the FAWT approach to separate the recorded EEG signals into five frequency sub-bands, the signals go through some preparatory processing. Numerous entropy metrics are derived from the decomposed sub-bands using analysis of variance and ranked according to their usefulness in detecting PD (ANOVA). Several classifiers are employed to locate appropriate feature sets, including support vector machines (SVM), logistics, random forests (RF), radial basis functions (RBF), and k-nearest neighbours (KNN).

**Conclusion** - The current work suggests a unique FAWT and entropy features-based method for exploiting multichannel EEG data to automatically identify Parkinson's disease patients from healthy controls. We discovered maximum mean accuracy of 99%, specificity of 99.45%, and sensitivity of 99.12% using Database-I. A further average accuracy of 95.85% was attained for dataset-II based on a consistent sample rate, along with a sensitivity of 96.14% and a specificity of 95.88%. From two distinct datasets, the diagnosis of PD patients took between 3.78 and 7.2 seconds to complete. Furthermore, the thorough examination and comparison with current studies have shown that the FAWT-based method employing a KNN classifier and a number of entropies is a promising method for detecting Parkinson's disease using EEG data.

## 2.9 Paper – 9

Prior to the development of characteristic motor impairment in persons with Parkinson's disease, hypomimia and voice changes serve as modest warning signs (PD). They are investigating if monitoring facial and acoustic expressions might aid in the early diagnosis of PD patients. A training cohort (111 controls and 112 PD patients during the "on" phase) and a validation cohort totaling 371 people were recruited (74 PD patients and 74 controls during the "off" phase). Each participant read an article while simultaneously having their voice and facial expressions recorded using a smartphone. Nine different machine learning classifiers were utilised. We discovered that the combination of face and voice data may be used to distinguish early-stage PD with an area under the receiver operating characteristic (AUROC) diagnostic value of 0.85.

**Conclusion** - Patients with PD read the article more slowly, paused more frequently, and had lower pitch and volume variation than controls in the voice analysis.

Patients with Parkinson's disease exhibited significantly fewer eye blinks than controls in the facial image study, but there was no statistically significant distinction between the groups for mouth angle or vascular motion variance.

## 2.10 Paper -10

Machine learning-based imaging methods have been employed in several neuroimaging studies to more accurately discriminate between parkinsonism and automatically diagnose PD in its early stages. Comparative studies have shown that machine-learning-based SPECT image analysis implementations in PD have outperformed traditional moderate analysis in detecting PD-associated dopamine receptor degeneration. These frameworks performed particularly well when compared to experts' visual inspection, and they have also helped improve radiologists' PD diagnostic accuracy. Combining multi-modal (clinical and imaging) data in various applications may enhance PD diagnosis and early identification. In order to integrate machine-learning-based diagnostic apps into healthcare systems, further validation and refinement of these applications are needed to make them reliable and reliable.

**Conclusion** - The utilisation of based on inter imaging and clinical data in these applications may help with PD diagnosis and early detection. Before being included into healthcare systems, more validation and improvement are needed to make these machine-learning applications reliable and accurate. Despite the challenges in integrating machine learning applications into

clinical practise, machine learning techniques have the potential to assist clinicians in better distinguishing between parkinson's disease and early PD diagnosis, which may reduce the error rate of PD diagnosis and help to determine PD at pre-motor stage so that early therapies (such as neuroprotective treatment) may be implemented to slow PD advancement, prevent the emergence of motor impairments illnesses, and reverse motor dysfunction.

### **2.11 Paper -11**

Using the most recent collection of Chinese data as a foundation, this effort aims to develop an automatic detection method. No agreement was reached about the primary indicators of vocal organ failure in language issues, in contrast to English. In order to classify the speech phonation and articulation, one of our solutions involves using a feature selection model based on machine learning. A very large sample of the vocal great way of showing from speech signals acquired in a controlled setting using three optimization techniques was used to evaluate four classifiers (Nave Bayes, K-Nearest Neighbor, Logistic Regression, and Stochastic Gradient Descent) to identify the disorder (LASSO, mRMR, Relief-F). The suggested method's ratings for precision, sensitivity, sensitivity, and accuracy are 75.76%, 82.44%, 73.15%, and 76.57%, respectively.

**Conclusion** - This study also offers helpful pointers for feature choice and classifier design. The following classifier only needs 10 characteristics from the articulation feature set, which is the most typical feature set of PWP. LASSO performs the best feature selection and LR delivers the best classification, while two combinations of LASSO & LR and LASSO & SGD all perform well. A larger dataset should be gathered to evaluate if the articulation characteristics are the most representative for Chinese-speaking PD. The optimal model suggested in this study is to filter 10 articulation characteristics using the LASSO technique and then utilize those features in an SGD or LR classifier. Further research may be done to examine the criteria for choosing between LASSO & LR or LASSO & SGD.

### **2.12 Paper -12**

This study investigated how well attribute selection strategies based on the filter method, the fuzzy system, and classification procedures performed when used to extract the features for supervised classification algorithms such as support vector machines (SVM), naive Bayes, k-nearest neighbours (K-NN), and artificial neural networks (ANN). Due to the fact that just a small number of clinical test results would be required for the diagnosis, a method like this might reduce the time and expense associated with PD screening. The suggested method was



compared to previously suggested PD diagnostic techniques and very well classifiers. The experiment findings show that SVM has an accuracy of 87.17%, naive Bayes has an accuracy of 74.11%, ANN has an accuracy of 96.7%, and KNN has an efficiency of 87.17%.

**Conclusion** - With proper feature selection, accuracy at the clinical level is achievable. The accuracy of four machine learning (ML) classifiers was investigated in this work. The classifiers were SVM (87.17%), ANN (96.7%), KNN (87.17%), and naive Bayes (74.11%). We were able to distinguish between ill and healthy people using these techniques. The sickness is determined by listening to human speech signals. The results gained demonstrate the effectiveness of feature selection techniques when combined with ML classifiers, especially when working with voice data where a large number of acoustic features may be recovered. PD may be correctly diagnosed in its early stages using the indicated early diagnostic technique, and its severe symptoms can be prevented.

### 2.13 Paper -13

The most frequent cause of resting tremors is idiopathic Parkinson's disease (PD). Neurologists assess patients with Parkinson's disease (PD) by performing tests including the pull test, finger-to-nose tests, and pacing back and forth in the hallway. The subjective Unified Parkinson's disease rating scale (UPDRS), which is based on a few short-term motor activities from daily life, is the main emphasis of this evaluation. This study skews a dataset of people with Parkinson's disease and runs a severity analysis on it. This is the cause of the noticeable decline in performance of several popular classification learning algorithms when it comes to categorizing diverse data with an imbalanced distribution of classes. In this work, we used resampling techniques such under-sampling, over-sampling, and a hybrid combination. renowned classifiers, such as XGBoost, and methods Resampling techniques are used with decision trees and K-nearest neighbours. The results showed that oversampling performed much better than hybrid and under sampling sampling techniques. The accuracy of random sampling using the oversampling techniques is 99% for the XGBoost classifier and 98% for the decision tree. It was also demonstrated that different classifiers had varying resampling responses.

**Conclusion** - To gauge the intensity of PD resting tremors, several machine learning classifiers are used with signal processing and resampling techniques. The results showed that oversampling techniques outperformed hybrid resampling and undersampling solutions. Among classifiers, the XGBoost classifier performs better than KNN and decision tree

classifiers. It is also highlighted that resampling techniques may not always work well with different classifiers. While certain measures are becoming better, others, including accuracy and precision, are extremely poor and have significantly deteriorated due to different under-sampling techniques. Our study examined the classification of certain courses, which was more significant than concentrating on total achievement. The suggested strategy requires more testing on many PwPD data sets (patients with PD). Large sample size data may be taken into consideration in the future. The most severely afflicted leg of PwPD was the source of the data evaluated in this study; if data from both upper limbs of PwPD were obtained, the results would have been different.

### 3 Methodology

In this Parkinson's disease study, we evaluated voice measures that were measured in frequency from 31 participants, 23 of whom had Parkinson's disease (PD). We utilized the classification algorithm to determine if a person had Parkinson's disease. The process of classification is understanding, identifying, and categorizing ideas and objects into specified groupings or "sub-populations." Machine learning systems use a variety of methods and pre-categorized training datasets to categorise new datasets. Machine learning classification algorithms predict the chance that new data will fall into one of the established categories based on input training data. In this study, Parkinson's illness is analysed using the methods Support Vector Machine (SVM), Naive Bayes, and Random Forest.

#### 3.1 Algorithms

##### 3.1.1 Support Vector Machine (SVM) algorithm

A family of supervised learning techniques for data categorization, regression analysis, and outlier identification is represented by support vector machines (SVMs). Statistical model-based SVMs are one of the most effective prediction techniques. Finding a hyperplane in an N-dimensional space (N is the number of features) that accurately classifies the data points is the goal of the support vector machine technique.

Support vector machines' benefits include:

1. Effective in situations with high dimensions.
2. Still helpful when the number of samples outweighs the size of the dimensions.
3. 3. Since the decision function only employs a part of the data point (also referred to as support vectors), it is also memory-efficient.
4. The decision function may be given a variety of Kernel functions, making it flexible.  
There are accessible standard kernels, but you may also design your own kernels.

Support vector machines have a number of drawbacks, including:

1. If there are many more features than samples, avoid over-fitting when choosing Kernel functions and performance and ensuring.
2. Probability estimates are derived via a time-consuming five-fold cross-validation procedure rather than being directly given by SVMs.

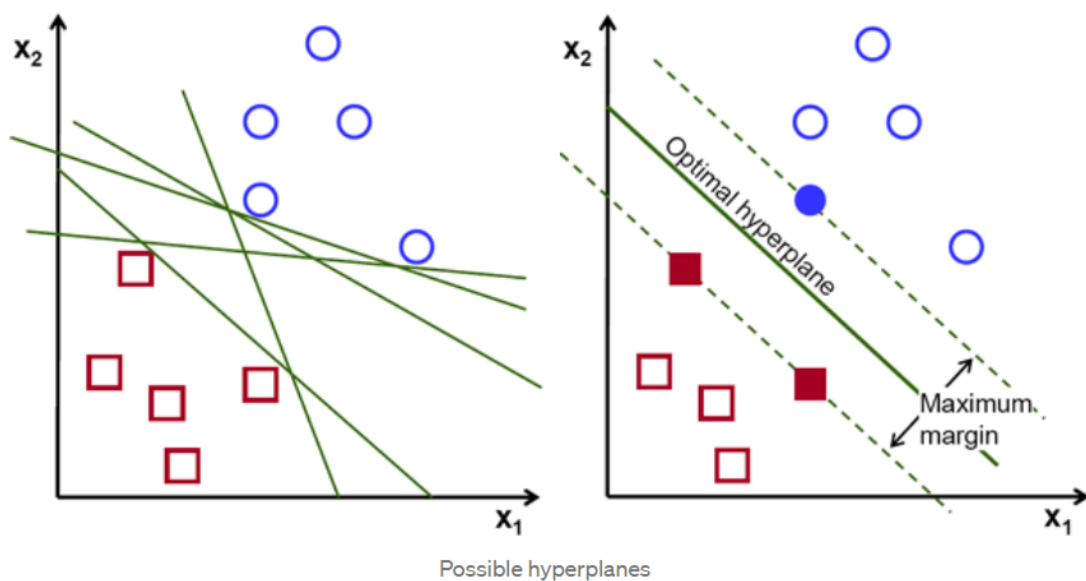


fig-1 Support Vector Machine

### Hyperplanes and Support Vectors

The classification of the data points is aided by hyperplanes, which act as decision boundaries. There are several categories that may be used to group the data points that are situated on each side of the hyperplane. The quantity of features also has an impact on the hyperplane's size. The hyperplane is essentially a line if there are just two input characteristics. The hyperplane collapses into a two-dimensional plane if there are three input features. When there are more than three characteristics, it gets more challenging to imagine.

### Large Margin Intuition

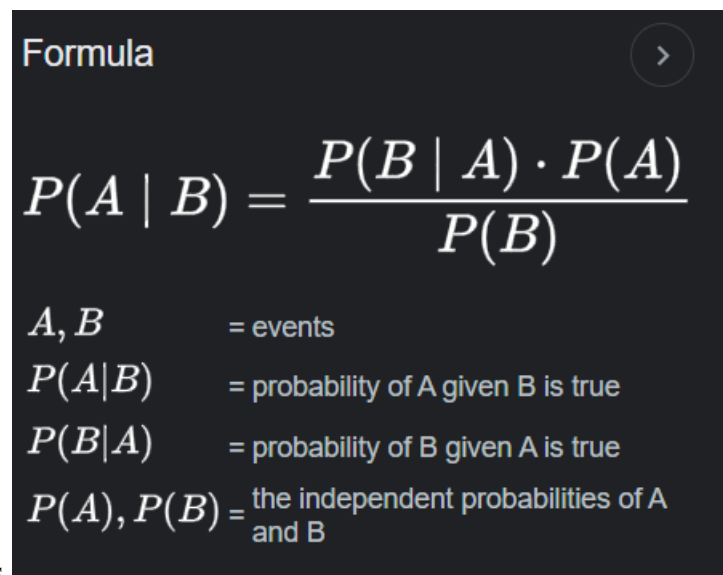
The linear function's output is taken into account in SVM. If the output is greater than 1, it belongs to one class; if it is less than -1, it belongs to another class. Since the threshold values in SVM are changed to 1 and -1, we obtain this reinforcing range of values  $[-1, 1]$  that serves as a margin..

## SVM Implementation in Python

In python Support vector machine can be implemented using the scikitlearn library . Scikit-learn is a free machine learning package for the Python programming language (formerly known as scikits.learn and also referred to as sklearn).

### 3.1.2 Naïve Bayes Algorithm

The family of straightforward "probabilistic classifiers," also known as "naive Bayes classifiers," in statistics are built on the application of Bayes' theorem with strong (naive) independence assumptions between the features (see Bayes classifier). Although they are among of the simplest Bayesian network models, they may attain extraordinarily high levels of accuracy when paired with kernel density estimation. A conditional probability model is



Formula

$$P(A | B) = \frac{P(B | A) \cdot P(A)}{P(B)}$$

$A, B$  = events  
 $P(A|B)$  = probability of A given B is true  
 $P(B|A)$  = probability of B given A is true  
 $P(A), P(B)$  = the independent probabilities of A and B

called Naive Bayes.

fig-2 Bayes Theorem Formula

### Gaussian Naïve Bayes

Naive Bayes may be expanded to real-valued features by most commonly assuming a Gaussian distribution.

This naive Bayes modification is known as Gaussian Naive Bayes. The Gaussian (or Normal) distribution is the most straightforward to use since it just needs that you estimate the mean and standard deviation from your training data, but other functions that may be used to estimate the data distribution can also be employed. Advantages

1. Reduce time and highly efficient.

2. For problems with multi-class prediction, Naive Bayes performs well..
3. If its presumption regarding the independence of attributes is accurate, it can perform better than other models and requires much less training data.
4. Naive Bayes is a better choice when the input variables are categorical as opposed to numerical.

#### Disadvantages

1. Naive Bayes bases its model on the improbable but unproven tenet that all predictors (or traits) are independent. This limits the algorithm's applicability in real-world use cases.
2. If a categorical variable's category was not included in the training dataset but was available in the test data set, this technique runs into the "zero-frequency problem," which results in a categorical variable being given with zero probability.

### Naïve Bayes Implementation in Python

In python Naïve Bayes can be implemented using the `sklearn.naive_bayes` and Importing the `GaussianNB` method .

#### 3.1.3 Random Forest

The random forest, as its name suggests, is an ensemble of many different decision trees that functions as a whole. The class with the highest votes becomes the prediction of our model.

The random forest generates a class prediction from each individual tree.

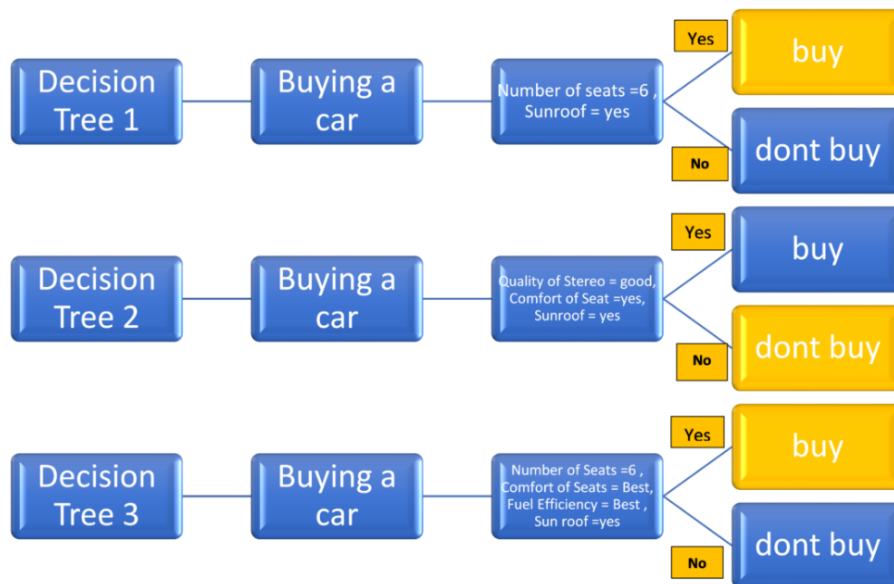
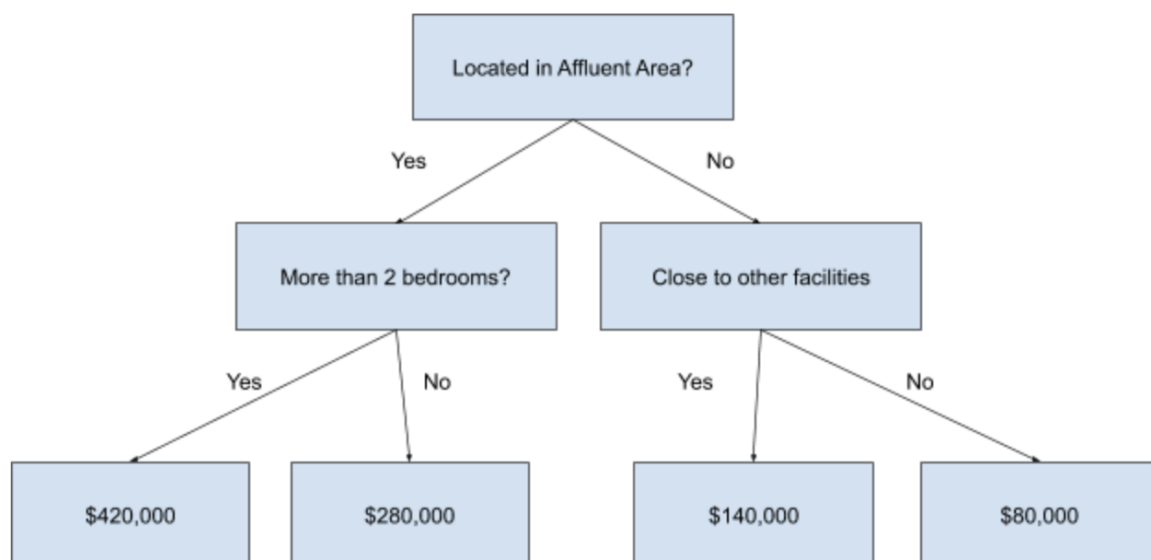


fig-3 Example of decision tree



The chart replicates a decision-making process by having a series of yes/no questions take you from the real estate description ("3 bedrooms") to its historical average price. Based on a property's attributes, the decision tree may be used to determine its expected price.

### Random Forest Benefits and Drawbacks

1. It helps to improve accuracy and prevents decision trees from overfitting. Both classification and regression problems may be accommodated by it.
2. It functions effectively with both continuous and categorical variables.
3. It automatically finds missing values in data.
4. Since it uses a rule-based methodology, data normalisation is not required.

Despite these benefits, a random forest method has certain disadvantages.

1. Since it builds several trees and then merges their results, it uses a lot of computer resources and power.
2. It also requires a lot of training time since it integrates several decision trees to determine the class.
3. The ensemble of decision trees makes it difficult to read and unable to determine the significance of each variable.

### Random Forest Implementation in Python

In python Random Forest can be implemented using the `sklearn.ensemble` and Importing the `RandomForestRegressor` method .

### 3.2 Exploratory Data Analysis (EDA)

Exploratory data analysis is a critical process for conducting initial research on data to find patterns, identify anomalies, test hypotheses, and verify presumptions using statistical results and visualizations.

#### A. Checking for Missing Data

While the data is collected, there will be missing values in the data frame .It is our responsibility to check for null values .If the dataset is huge we can take the mean of the column and replace the null value .If the dataset has too many rows ,the rows with null values can be deleted .

#### B. Brief description of the sample and its features.

Once the missing values of the dataset are resolved its important to categorize it



**Continuous:** A continuous feature can take on an endless number of values within a given range. A merchant's Gross Merchandise Value is an example of a continuous characteristic (GMV).

**Discrete:** A discrete feature has a countable number of values and is always numeric. A merchant's Sessions are an example of a distinct feature.

**Categorical:** A discrete feature can only have a limited number of values. A merchant's Shopify plan type is an example of a distinct feature.

### **Find the shape of the data**

It's important to find the shape of dataset to find the distribution, also finding the mean and variance of the features.

### **Identify Significant Correlations**

The scatter plot is the simplest visual representation of correlation. A correlation matrix is an additional option. It determines the linear relation between the features in your information and assigns a value between -1 and 1 to each feature pair. A positive value denotes a favourable association, whereas a negative value denotes an unfavourable one.

### **Spot Outliers in the dataset**

Detecting outliers in your dataset is a critical step in EDA. Outliers deviate dramatically from other samples in your dataset and might cause severe issues when executing statistical activities based on your EDA. Outliers may be easily identified using the box plot visualization. We can see in the following image that all features have a lot of outliers since we find data points that are far apart from the bulk of the data.

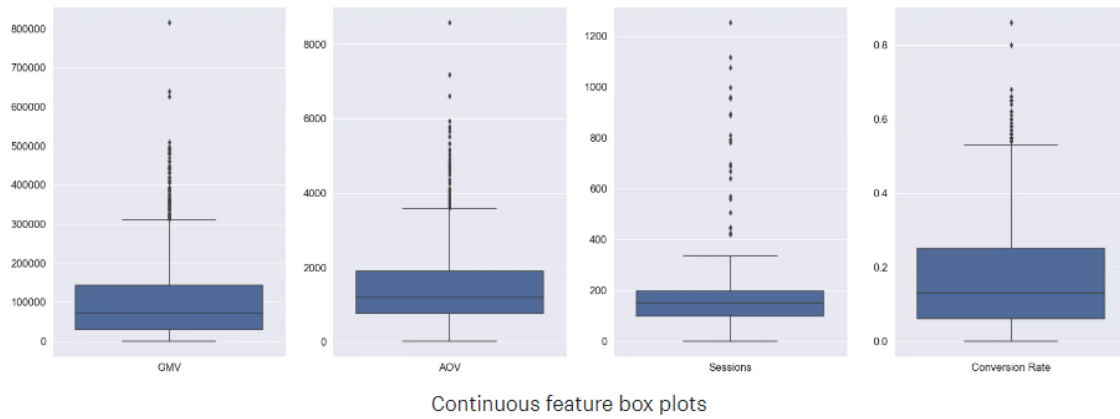


fig-4 Box plot to find outliers

## 4 Implementation

To predict whether the patient has Parkinson's disease, the data has been extracted from the UCI machine learning repository. Various biological voice measurements from 31 individuals, 23 of whom have Parkinson's disease, are included in this collection (PD). Each row in the table corresponds to one of the 195 voice recordings of these persons, and each column denotes a certain vocal measure ("name" column). The "status" column, which is set to 0 for healthy and 1 for PD, is used to identify healthy and PD individuals using the data.

The data is in CSV ASCII format. Each voice recording has its own instance in the CSV file's rows. Each patient has about six recordings, with the patient's name appearing in the first column.

### Attribute Information:

Matrix column entries (attributes):

name - ASCII subject name and recording number

MDVP:Fo(Hz) - Average vocal fundamental frequency

MDVP:Fhi(Hz) - Maximum vocal fundamental frequency

MDVP:Flo(Hz) - Minimum vocal fundamental frequency

MDVP: Jitter(%), MDVP:Jitter(Abs), MDVP: RAP,MDVP: PPQ, Jitter: DDP - Several measures of variation in fundamental frequency

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

NHR, HNR - Two measures of the ratio of noise to tonal components in the voice

status – The health status of the subject (one) - Parkinson's, (zero) – healthy

RPDE,D2 - Two nonlinear dynamical complexity measures

DFA - Signal fractal scaling exponent

spread1,spread2, PPE - Three nonlinear measures of fundamental frequency variation.

Below shows the table of the Parkinson's Disease having 195 rows and 24 columns

Out[4]:

	name	MDVP:F0(Hz)	MDVP:F1(Hz)	MDVP:F2(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	...	Sh
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109	0.04374	...	
1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	...	
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633	0.05233	...	
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	...	
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	...	
...	...	...	...	...	...	...	...	...	...	...	...	
190	phon_R01_S50_2	174.188	230.978	94.261	0.00459	0.00003	0.00263	0.00259	0.00790	0.04087	...	
191	phon_R01_S50_3	209.516	253.017	89.488	0.00564	0.00003	0.00331	0.00292	0.00994	0.02751	...	
192	phon_R01_S50_4	174.688	240.005	74.287	0.01360	0.00008	0.00624	0.00564	0.01873	0.02308	...	
193	phon_R01_S50_5	198.764	396.961	74.904	0.00740	0.00004	0.00370	0.00390	0.01109	0.02296	...	
194	phon_R01_S50_6	214.289	260.277	77.973	0.00567	0.00003	0.00295	0.00317	0.00885	0.01884	...	

195 rows x 24 columns

Fig-5

Most of the code is written in Pandas and NumPy

## Pandas

Pandas is a python library that is used to analyse the data. The pandas package is the most significant tool available to Python Data Scientists and Analysts today. Although advanced machine learning and glitzy visualisation technologies get all the attention, pandas remains the foundation of most data initiatives.

## NumPy

A Python library for processing arrays is called NumPy. Additionally, it offers tools for working with matrices, the Fourier transform, and linear algebra.

## Jupyter Notebook

The Jupyter Notebook is a powerful tool for creating and presenting interactive data science projects. A notebook is a piece of writing that combines code, its output, visuals, narrative prose, mathematical formulas, and other rich media. In order to make your work more understandable, reproducible, and shareable, you may run code, evaluate the results, and add explanations, formulas, and charts on a single page.

## Anaconda

Anaconda is an open-source distribution of Python and R for data analysis that aims to simplify package management and deployment. Package versions are managed by Conda, the Anaconda package management system. Before starting an installation, Conda checks the current environment to rule out conflicts with other frameworks and packages. The Anaconda distribution automatically sets up more than 250 packages. More than 7500 more open-source packages may be installed through PyPI in addition to the virtual environment manager and the conda package manager. Also available as a graphical substitute for the command line interface is Anaconda Navigator, a GUI (graphical user interface)

### 4.1 Exploratory Data Analysis

#### 4.1.1 Data Pre-processing

The data has been extracted from the UCI machine learning repository. Using Pandas the data has been extracted. The first and most important part is to import the necessary libraries. Here Pandas and NumPy libraries are imported first.

```
In [1]: import numpy as np  
import pandas as pd
```

Once the libraries are imported, we write the code for reading the csv file which is stored in the local folder.

```
In [2]: df = pd.read_csv("parkinsons.csv")
```

```
In [3]: df
```

```
Out[3]:
```

	name	MDVP:F0(Hz)	MDVP:F1(Hz)	MDVP:F2(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	...	Sh
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109	0.04374	...	...
1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	...	...
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633	0.05233	...	...
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	...	...
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...
190	phon_R01_S50_2	174.188	230.978	94.261	0.00459	0.00003	0.00263	0.00259	0.00790	0.04087	...	...
191	phon_R01_S50_3	209.516	253.017	89.488	0.00564	0.00003	0.00331	0.00292	0.00994	0.02751	...	...
192	phon_R01_S50_4	174.688	240.005	74.287	0.01360	0.00008	0.00624	0.00564	0.01873	0.02308	...	...
193	phon_R01_S50_5	198.764	396.961	74.904	0.00740	0.00004	0.00370	0.00390	0.01109	0.02296	...	...
194	phon_R01_S50_6	214.289	260.277	77.973	0.00567	0.00003	0.00295	0.00317	0.00885	0.01884	...	...

195 rows x 24 columns

Fig-6

Fig -6 shows the CSV (comma separated value) being read using the pandas library where `pd` is the pandas object and `read_csv` is the function, `df` stands for data frame.

The next step is to find the shape of the dataset. Here the shape function returns a tuple of array. Once the shape is found it is important for finding missing values in the data frame .If the missing values are not resolved the data frame is not set for further analysis, for example ,the dataset cannot fit into the model and an error would be thrown.

```
In [4]: df.shape
```

```
Out[4]: (195, 24)
```

Fig-7

```
In [6]: #Finding null values in the dataset
df.isnull().sum()
```

```
Out[6]: name                0
MDVP:F0(Hz)                0
MDVP:F1(Hz)                0
MDVP:F1o(Hz)               0
MDVP:Jitter(%)             0
MDVP:Jitter(Abs)           0
MDVP:RAP                    0
MDVP:PPQ                    0
Jitter:DDP                  0
MDVP:Shimmer                0
MDVP:Shimmer(dB)            0
Shimmer:APQ3                0
Shimmer:APQ5                0
MDVP:APQ                    0
Shimmer:DDA                 0
NHR                          0
HNR                          0
status                      0
RPDE                         0
DFA                          0
spread1                      0
spread2                      0
D2                           0
PPE                          0
dtype: int64
```

Fig-8

In fig-8 we use the isnull function to check whether the columns has any missing values ,we also the sum function to give the total missing values in the columns. As shown in the figure no missing values were found, all the columns with zero value, which is a good sign, and also we can say the dataset is a good dataset.

Next thing would be finding the datatypes of the attributes .Here we going to use the dtypes function to check the datatype.

```
In [7]: df.dtypes
```

```
Out[7]: name                object
MDVP:F0(Hz)               float64
MDVP:F1(Hz)               float64
MDVP:F1o(Hz)              float64
MDVP:Jitter(%)            float64
MDVP:Jitter(Abs)          float64
MDVP:RAP                   float64
MDVP:PPQ                   float64
Jitter:DDP                 float64
MDVP:Shimmer               float64
MDVP:Shimmer(dB)          float64
Shimmer:APQ3               float64
Shimmer:APQ5               float64
MDVP:APQ                   float64
Shimmer:DDA                float64
NHR                        float64
HNR                        float64
status                     int64
RPDE                       float64
DFA                        float64
spread1                    float64
spread2                    float64
D2                         float64
PPE                        float64
dtype: object
```

Fig-9

The above figure shows that the name attribute is an object ,the rest of the attributes are float64 data type.

#### 4.1.2 Exploratory Data Analysis

The Matplotlib library and seaborn library will be used for the exploratory data analysis on our dataset. A Python charting toolkit called Matplotlib and its NumPy extension for numerical mathematics are also available. Plots may be included into applications that utilise all-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK using its object-oriented API. There is also a procedural "pylab" interface that is meant to be comparable to MATLAB and is built on a state machine (similar to OpenGL), albeit its use is discouraged. A Python data visualisation toolkit called Seaborn uses the matplotlib library. It provides a high-level interface for producing attractive and practical statistical graphics.

In our dataset, our main concentration would be the status column. The column is either one or zero, one representing the person who has Parkinson disease. We plotted a bar plot of the status attribute which is in ones and zeros using the seaborn library.

```
In [8]: #Checking Label imbalance
import matplotlib.pyplot as plt
import seaborn as sns
temp = df['status'].value_counts()
temp_df=pd.DataFrame({'status':temp.index,'values':temp.values})
print((sns.barplot(x='status',y='values',data=temp_df)))

AxesSubplot(0.125,0.125;0.775x0.755)
```

Fig-10



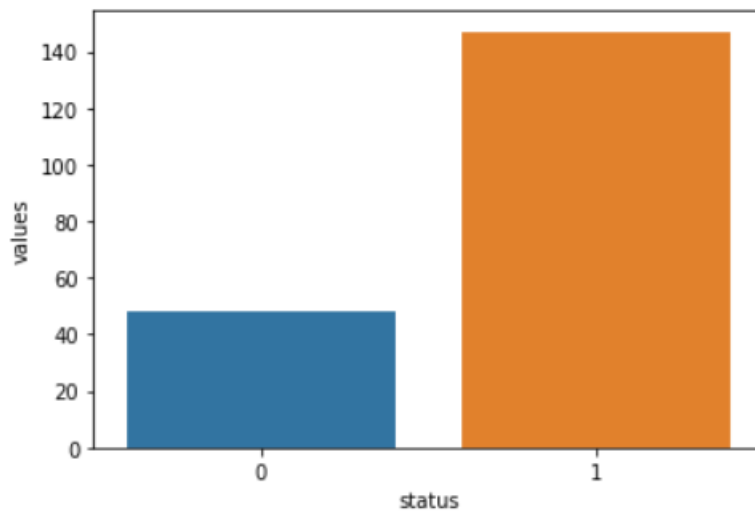


Fig-11

Fig-11 shows the bar plot of the status column .The orange bar plot shows number people who have Parkinson's disease and blue bar represents the number of patients who do not have any Parkinson disease.

### Plotting the Pair plot

A pairs plot displays the connections between two variables as well as the distribution of a single variable. Pair plots are a great tool for spotting patterns that need further investigation and are comparatively easy to make in Python. The scatter plot and the histogram serve as the foundation for the pairs plot. The top and bottom triangular scatter plots demonstrate the relationship (or lack thereof) between the two variables, whereas the diagonal histogram displays the distribution of a single variable.

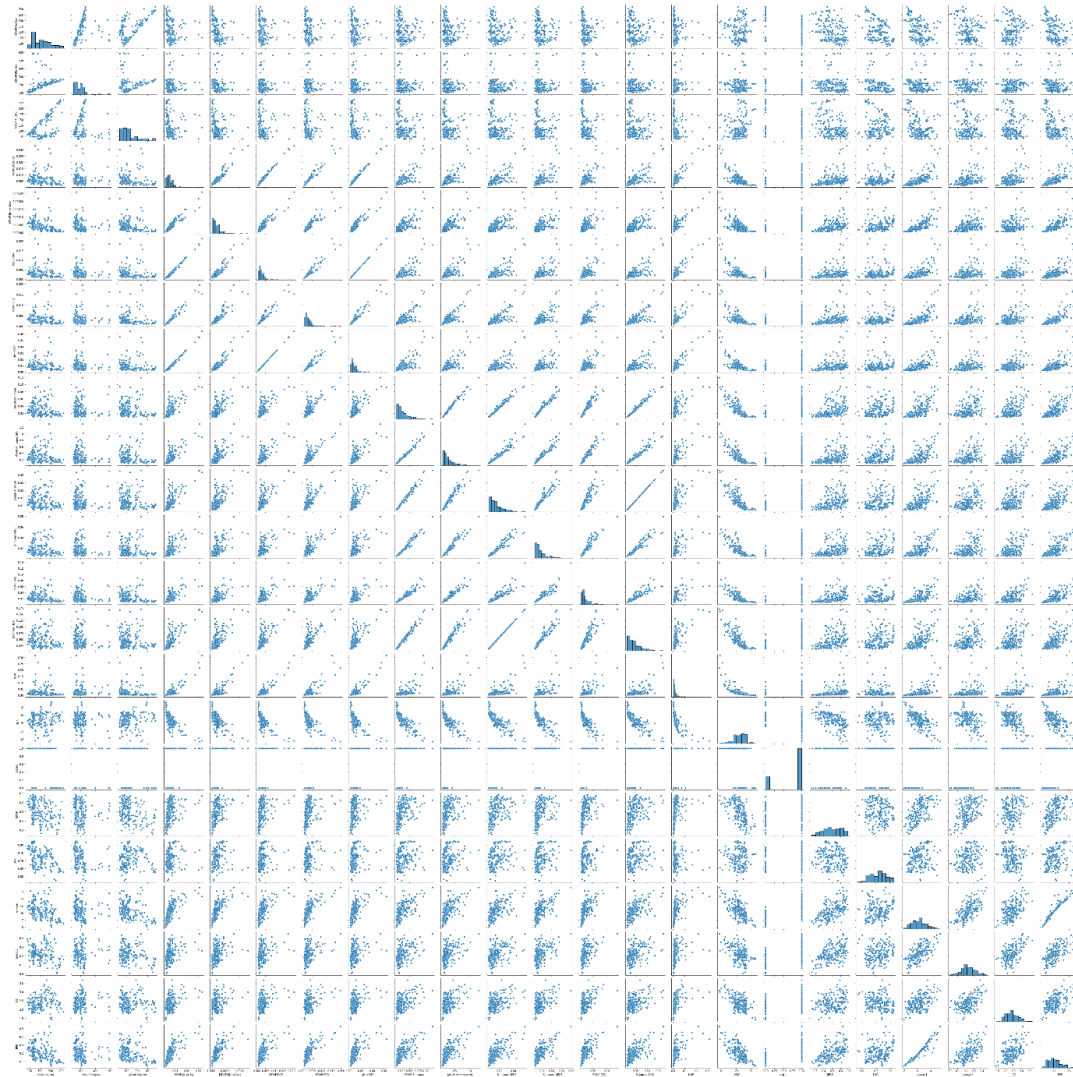


Fig-12

The above fig shows the relationship between each other using the scatter plot , and the diagonal shows the histogram of the data frame.

### Finding the distribution

All of the possible values for Data are listed by the function called Data Distribution. A continuous or discrete data distribution is conceivable. The possibility of several likely outcomes in an experiment is estimated using a number of well-known classical probability distribution functions. The distribution code in Python may be written as follows.

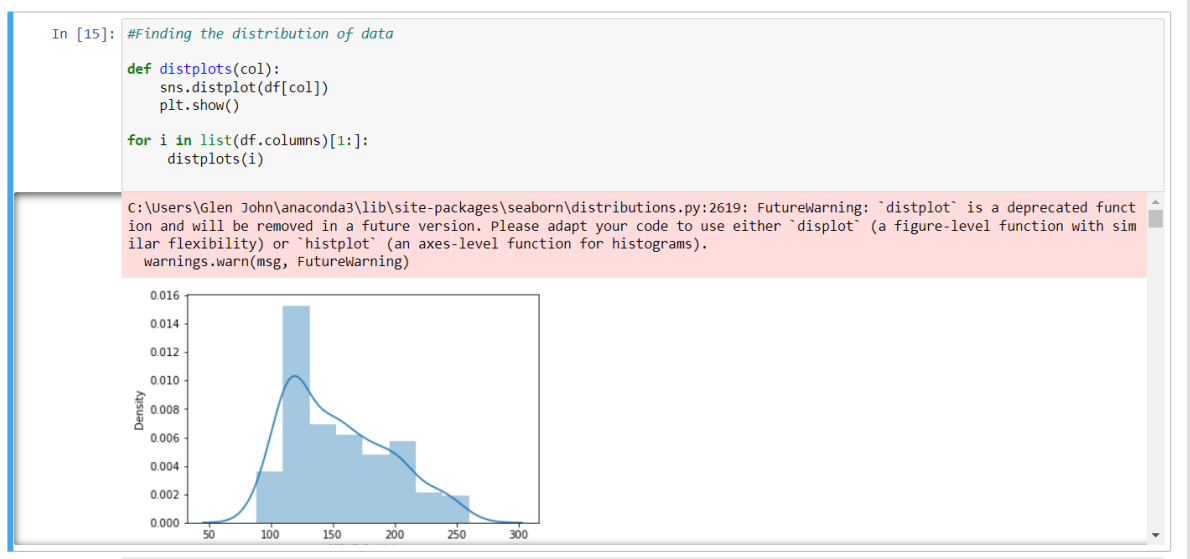


Fig-13

The above code gives the distribution of all the columns, they give a histogram.

### Finding the Outliers

A value in a random sample from a population that differs considerably from the other values is called an outlier. In some aspects, this approach leaves the determination of what is abnormal up to the analyst (or a consensus process). Before anomalous observations may be found, normal observations must first be described. For explaining the behaviour of data in the middle and at the ends of distributions, the box plot is a useful graphical tool. The lower and upper quartiles, as well as the median, are used in the box plot (defined as the 25th and 75th percentiles). The difference ( $Q3 - Q1$ ) is known as the interquartile range if the lower quartile is  $Q1$  and the higher quartile is  $Q3$  (IQ). The boxplot in Python can be by the following code.

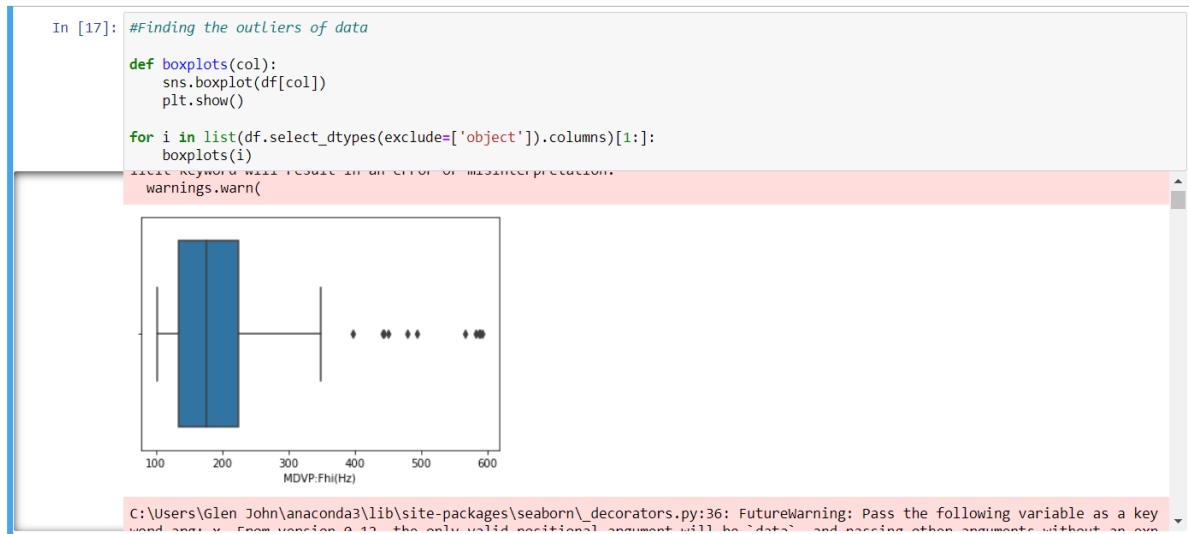


Fig-14

So the majority of the columns have outliers, the small dots indicate the outliers present in the data. Box plots are added in the appendix.

### Finding the correlation

The statistical link between two variables is measured via correlation analysis. The outcome will indicate how a change in one parameter affects the other. Correlation analysis is a crucial topic that is often used in predictive analytics. Also, before developing the model and reaching a conclusion regarding variable associations, the correlation analysis must be completed. Although correlation analysis can help us comprehend the relationship between two variables in a dataset, it cannot explain or quantify the reason.

Correlation values will vary from -1 to +1, with a positive number indicating a positive link and a negative value indicating a negative relationship. In contrast to the negative relationship, which suggests that increasing the value of one variable would reduce the value of the other, the positive association suggests that increasing the value of one variable will raise the value of the other variable. The parameters in the example above have a strong link with one another, as indicated by the correlation value of 0.8934.

In python correlation can be found using the corr function, also using seaborn the heatmap is plotted.

```
In [18]: #Finding correlation
plt.figure(figsize=(20,20))
corr=df.corr()
sns.heatmap(corr,annot=True)

Out[18]: <AxesSubplot:>
```

Fig-15

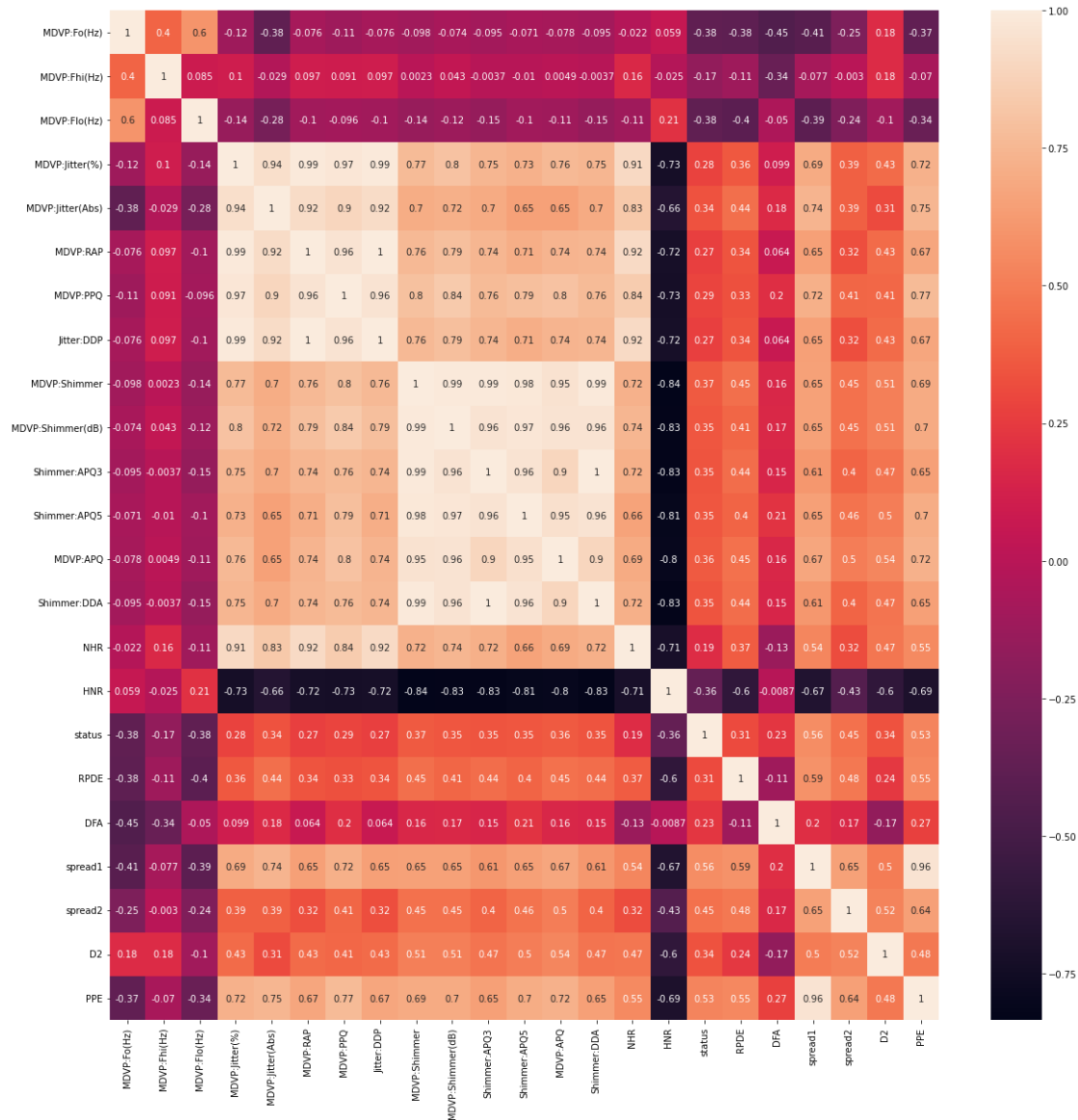


Fig-16

In the above heatmap, the independent variable which is the status column has a very good correlation with the dependent variable, which is what we need for the further analysis of the model,5.

## 5 Testing

Once the data is preprocessed, its time to train and test the model .In this study 3 algorithms have been used which are SVM(Support vector machine) , Naïve bayes and random forest .All the algorithms have been imported from scikit learn .All the codes have been written in jupyter notebook .

The first step is importing the necessary libraries ,here NumPy ,Pandas ,svm(support vector machine), GaussianNB ,RandomForestClassifier ,accuracy\_score ,StandardScaler,train\_test\_split.

### Accuracy\_score

The accuracy score function in the Python sklearn.metrics module determines the degree of agreement between a set of predicted labels and actual labels.

### StandardScaler

The scaling of features is an important stage in modelling algorithms using datasets. The data that is typically utilized for modelling purposes comes from surveys and Questionnaire,research or scraping. As a result, the acquired data comprises characteristics of various dimensions and sizes. The modelling of datasets is negatively impacted by the different sizes of data features. As a result, the accuracy and misclassification error rates of the prediction are skewed. As a result, the data has to be scaled before modelling.

### Train\_test\_split

It is sufficient to randomly partition your dataset into the following three subsets:

1. Using the training set, your model is fitted or trained. The training set can be used to find the ideal weights or coefficients for logistic regression, neural networks, or linear regression.
2. The validation set is used for unbiased model evaluation during hyperparameter tuning.

To ascertain the appropriate kernel for a support vector machine or the perfect number of neurons for a neural network, for instance, experiment with various values. For each considered value of the hyperparameters, fit the model with the training set and assess its performance using the validation set.

3. The test set is necessary for an objective evaluation of the final model. Not for fitting or validation purposes.

Following is the code for importing libraries

```
In [2]: import numpy as np
import pandas as pd
from sklearn import svm
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

Fig-17

The next step is to import the downloaded dataset from the local computer

```
In [3]: df = pd.read_csv("parkinsons.csv")
```

```
In [4]: df
```

```
Out[4]:
```

	name	MDVP:F0(Hz)	MDVP:F1(Hz)	MDVP:F2(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	...	Sh
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109	0.04374	...	...
1	phon_R01_S01_2	122.400	148.850	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	...	...
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633	0.05233	...	...
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	...	...
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...
190	phon_R01_S50_2	174.188	230.978	94.261	0.00459	0.00003	0.00263	0.00259	0.00790	0.04087	...	...
191	phon_R01_S50_3	209.516	253.017	89.488	0.00564	0.00003	0.00331	0.00292	0.00994	0.02751	...	...
192	phon_R01_S50_4	174.688	240.005	74.287	0.01360	0.00008	0.00624	0.00564	0.01873	0.02308	...	...
193	phon_R01_S50_5	198.764	396.961	74.904	0.00740	0.00004	0.00370	0.00390	0.01109	0.02296	...	...
194	phon_R01_S50_6	214.289	260.277	77.973	0.00567	0.00003	0.00295	0.00317	0.00885	0.01884	...	...

195 rows x 24 columns

Fig-18

When we use the info () function in the dataframe ,it gives the information of the dataset like the name of the columns , number of null values, datatype of column

```
In [5]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 24 columns):
#   Column              Non-Null Count  Dtype
---  -
0   name                 195 non-null   object
1   MDVP:Fo(Hz)         195 non-null   float64
2   MDVP:Fhi(Hz)        195 non-null   float64
3   MDVP:Flo(Hz)        195 non-null   float64
4   MDVP:Jitter(%)      195 non-null   float64
5   MDVP:Jitter(Abs)    195 non-null   float64
6   MDVP:RAP            195 non-null   float64
7   MDVP:PPQ            195 non-null   float64
8   Jitter:DDP          195 non-null   float64
9   MDVP:Shimmer        195 non-null   float64
10  MDVP:Shimmer(dB)    195 non-null   float64
11  Shimmer:APQ3        195 non-null   float64
12  Shimmer:APQ5        195 non-null   float64
13  MDVP:APQ            195 non-null   float64
14  Shimmer:DDA        195 non-null   float64
15  HNR                 195 non-null   float64
16  HNR                 195 non-null   float64
17  status              195 non-null   int64
18  RPDE               195 non-null   float64
19  DFA                195 non-null   float64
20  spread1            195 non-null   float64
21  spread2            195 non-null   float64
22  D2                 195 non-null   float64
23  PPE                195 non-null   float64
dtypes: float64(22), int64(1), object(1)
memory usage: 36.7+ KB
```

Fig-18

Next function used is the describe function ,which shows the statistical values of the attributes .The describe function will show count value ,mean value , standard deviation ,minimum value , maximum value , lower quartile(25%) , upper quartile (75%) ,median value (50%) .

```
In [6]: df.describe()

Out[6]:
```

	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	MDVP:Shimmer(dB)
count	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000
mean	154.228641	197.104918	116.324631	0.006220	0.000044	0.003306	0.003446	0.009920	0.029709	0.282251
std	41.390065	91.491548	43.521413	0.004848	0.000035	0.002968	0.002759	0.008903	0.018857	0.194877
min	88.333000	102.145000	65.476000	0.001680	0.000007	0.000680	0.000920	0.002040	0.009540	0.085000
25%	117.572000	134.862500	84.291000	0.003460	0.000020	0.001660	0.001860	0.004985	0.016505	0.148500
50%	148.790000	175.829000	104.315000	0.004940	0.000030	0.002500	0.002690	0.007490	0.022970	0.221000
75%	182.769000	224.205500	140.018500	0.007365	0.000060	0.003835	0.003955	0.011505	0.037885	0.350000
max	260.105000	592.030000	239.170000	0.033160	0.000260	0.021440	0.019580	0.064330	0.119080	1.302000

8 rows x 23 columns

Fig-19

For our analysis status column is the dependent variable and the independent variable will be every column with name and status dropped from the dataset.

```
In [11]: X = df.drop(columns=['name', 'status'], axis=1)
         Y = df['status']
```

Fig-20



```
In [12]: print(X)
```

	MDVP:F0(Hz)	MDVP:F1(Hz)	MDVP:F2(Hz)	MDVP:Jitter(%)	\
0	119.992	157.302	74.997	0.00784	
1	122.400	148.650	113.819	0.00968	
2	116.682	131.111	111.555	0.01050	
3	116.676	137.871	111.366	0.00997	
4	116.014	141.781	110.655	0.01284	
..	...	...	...	...	
190	174.188	230.978	94.261	0.00459	
191	209.516	253.017	89.488	0.00564	
192	174.688	240.005	74.287	0.01360	
193	198.764	396.961	74.904	0.00740	
194	214.289	260.277	77.973	0.00567	

	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	\
0	0.00007	0.00370	0.00554	0.01109	0.04374	
1	0.00008	0.00465	0.00696	0.01394	0.06134	
2	0.00009	0.00544	0.00781	0.01633	0.05233	
3	0.00009	0.00502	0.00698	0.01505	0.05492	
4	0.00011	0.00655	0.00908	0.01966	0.06425	
..	...	...	...	...	...	
190	0.00003	0.00263	0.00259	0.00790	0.04087	
191	0.00003	0.00331	0.00292	0.00994	0.02751	
192	0.00008	0.00624	0.00564	0.01873	0.02308	
193	0.00004	0.00370	0.00390	0.01109	0.02296	
194	0.00003	0.00295	0.00317	0.00885	0.01884	

Fig-21

```
In [13]: print(Y)
```

0	1
1	1
2	1
3	1
4	1
..	
190	0
191	0
192	0
193	0
194	0

Name: status, Length: 195, dtype: int64

Fig-22

Fig-21 and fig-22 shows the print statements of the X and the Y variables

## 5.1 5.1 Model Analysis

The 3 algorithms that have been used for this project are Support vector machine , naïve bayes and random forest .All the algorithms have been imported from scikit learn library.As shown in the figure -17.For the next step we have to split the dataset into training data and testing data

```
In [14]: X_train ,X_test ,Y_train ,Y_test =train_test_split(X, Y ,test_size=0.2,random_state =2)
```

Fig-23

In fig-23 X,Y variables are split ,the test\_size is the size of the testing data , here the test\_size is give as 0.2 which is the common value taken for most of the time for splitting the data. The random state is given as 2, which means Every time we run our code, a new random value is created, and the train and test datasets have distinct values. However, if a fixed number is

assigned, such as random state = 0 or 1 or 42 or any other integer, No matter how many times our code gets performed, the result is constant. in the train and test datasets, same values.

```
In [15]: print(X_train.shape,X_test.shape)
(156, 22) (39, 22)
```

Fig-24

The above figure shows the shape of the training data and shape of the testing data .Here the training data is set to 156 rows with 22 columns and testing data is split into 39 rows and 22 columns.

```
In [16]: ss = StandardScaler()
In [17]: ss.fit(X_train)
Out[17]: StandardScaler()
```

Fig-25

In the above figure the trained dataset is scaled down using the standard scaler library.

```
In [18]: X_train = ss.transform(X_train)
X_test = ss.transform(X_test)
In [19]: print(X_train)
[[ 0.63239631 -0.02731081 -0.87985049 ... -0.97586547 -0.55160318
  0.07769494]
 [-1.05512719 -0.83337041 -0.9284778 ... 0.3981808 -0.61014073
  0.39291782]
 [ 0.02996187 -0.29531068 -1.12211107 ... -0.43937044 -0.62849605
 -0.50948408]
 ...
 [-0.9096785 -0.6637302 -0.160638 ... 1.22001022 -0.47404629
 -0.2159482 ]
 [-0.35977689 0.19731822 -0.79063679 ... -0.17896029 -0.47272835
 0.28181221]
 [ 1.01957066 0.19922317 -0.61914972 ... -0.716232 1.23632066
 -0.05829386]]
```

Fig-26

Transform() is a method of class `sklearn.preprocessing.StandardScaler`. We may alter our test data using the same mean and variance that we calculated from our training set utilising the transform approach. Therefore, the parameters that our model learned from the training set will help us change the test set of data. An array of data points in the range of 0 to 1 will be the result of the transform technique.

### 5.1.1 Support Vector Machine

```
In [21]: model = svm.SVC(kernel = 'linear')  
In [22]: model.fit(X_train,Y_train)  
Out[22]: SVC(kernel='linear')
```

Fig-27

Now the first model is created, here the SVM analysis will be done using the SVC classifier by using the kernel argument as linear. A kernel machine is what the Support Vector Machine is. As a consequence, by utilizing a different kernel function, you may modify its behavior.

The following are the most often-used kernel functions:

1. Linear kernel
2. Polynomial kernel
3. RBF (Gaussian) kernel
4. String kernel

Finally the trained data is fitted into the model.

```
In [23]: X_train_pre = model.predict(X_train)  
         train_pre_accu = accuracy_score(Y_train,X_train_pre)  
In [24]: print('accuracy of training data:',train_pre_accu)  
accuracy of training data: 0.8846153846153846
```

Fig-28

The predict method () is used to predict the trained data. The data to be tested is often the sole input that the predict() function allows. Based on the model's learned or trained data, it returns the labels for the input data. To map and predict the labels for the data to be tested using the learned label, the predict() function acts on top of the trained model.

The accuracy score is predicted using the accuracy\_score function. The input arguments are Y\_train and X\_train\_pre

One parameter for assessing classification models is accuracy. Informally, accuracy is the percentage of correct predictions made by our model. Formally, accuracy is defined as follows:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Fig-29

Since for our SVM model we got an accuracy of 88.4% ,which means 88 predictions were made correct out of 100 .So finally we can say our model can successfully predict the Parkinson's disease.

```
In [25]: X_test_pre = model.predict(X_test)
         test_pre_accu = accuracy_score(Y_test,X_test_pre)

In [26]: print('accuracy of testing data:',test_pre_accu)
         accuracy of testing data: 0.8717948717948718
```

Fig-30

The above code predicts the accuracy for test data. When the test data prediction was done the total accuracy was found to be 87.1% which means 87 predictions were correct out of 100

```
In [27]: input_data = (202.26600,211.60400,197.07900,0.00180,0.000009,0.00093,0.00107,0.00278,0.00954,0.08500,0.00469,0.00606,0.00719,0.0
         input_data_np = np.asarray(input_data)
         input_data_np_re = input_data_np.reshape(1,-1)
         s_data = ss.transform(input_data_np_re)

         predict = model.predict(s_data)
         print(predict)

         if(predict[0]==0):
             print('Negative,No parkinson Found')
         else:
             print('Positive, parkinson Found')

[0]
Negative,No parkinson Found
```

Fig-31

When the above model was tested with raw data in which the patient had no Parkinson disease. First the raw input data was converted into an array .Once the inputted array is formed the it is reshaped into (1,-1) .The reshaped data is scaled down using the transform method from the standardscaler.The scaled down model is added into the model. And finally using a simple if – else statement to print down the Parkinson's statement whether the patient has Parkinson disease or not.

In the final output the model predicted that the person has no Parkinson disease.

### 5.1.2 Naïve Bayes Model

#### NAIVE bayes

```
In [28]: clf = GaussianNB()  
         clf.fit(X_train , Y_train)  
  
Out[28]: GaussianNB()
```

Fig-32

The GaussianNB() method is imported from the sklearn library .Following the same steps of the previous model ,the X\_train and Y\_train are fitted .

```
In [29]: X_train2_pre = clf.predict(X_train)  
         train2_pre_accu = accuracy_score(Y_train,X_train2_pre)  
  
In [30]: print('accuracy of training data:',train2_pre_accu)  
         accuracy of training data: 0.7243589743589743
```

Fig-33

The training data prediction values are created using the predict method ,and the accuracy score of the model is found out to be 72.4% which is less than the SVM model prediction .

```
In [31]: X_test2_pre = clf.predict(X_test)  
         test2_pre_accu = accuracy_score(Y_test,X_test2_pre)  
  
In [32]: print('accuracy of testing data:',test2_pre_accu)  
         accuracy of testing data: 0.6153846153846154
```

Fig-34

The accuracy of testing data is found out to be 61.4% which is low as compared to training data.

The model was tested with raw data .

```

In [33]: input_data = (119.99200,157.30200,74.99700,0.00784,0.00007,0.00370,0.00554,0.01109,0.04374,0.42600,0.02182,0.03130,0.02971,0.0654
input_data_np = np.asarray(input_data)
input_data_np_re = input_data_np.reshape(1,-1)
s_data = ss.transform(input_data_np_re)

predict = clf.predict(s_data)
print(predict)

if(predict[0]==0):
    print('Negative, No parkinson Found')
else:
    print('Positive, parkinson Found')

[1]
Positive, parkinson Found

```

Fig-35

When raw data was inputted in the model .The model successfully predicted the patient has Parkinson disease.

### 5.1.3 Random Forest

For the third model the RandomForestClassifier is imported from sklearn

```

In [34]: model3 = RandomForestClassifier(criterion='gini')

```

Fig-36

The gini impurity quantifies the likelihood that any piece of the dataset will be mislabeled when randomly classified.

The minimal value of the Gini Index is 0. This happens when the node is pure, which means that every component of the node belongs to the same class. This node will never again be split as a result. The traits with the lowest Gini Index thereby determine the optimal division. Additionally, it is most valuable when the probabilities for the two classes are identical.

```

In [35]: model3.fit(X_train,V_train)
Out[35]: RandomForestClassifier()

In [36]: X_train3_pre = model3.predict(X_train)
train3_pre_accu = accuracy_score(Y_train,X_train3_pre)

In [37]: print('accuracy of training data:',train3_pre_accu)

accuracy of training data: 1.0

```

The prediction for training data is done using the predict method ,the accuracy for training data was found out to be 1 ,which is not quite accurate.

```
In [38]: X_test3_pre = model3.predict(X_test)
         test3_pre_accu = accuracy_score(Y_test,X_test3_pre)

In [39]: print('accuracy of testing data:',test3_pre_accu)
         accuracy of testing data: 0.8205128205128205
```

Fig-37

The accuracy of the testing data was found to be 82% which is better than naïve bayes model.

```
In [40]: input_data = (119.99200,157.30200,74.99700,0.00784,0.00007,0.00370,0.00554,0.01109,0.04374,0.42600,0.02182,0.03130,0.02971,0.0654)
         input_data_np = np.asarray(input_data)
         input_data_np_re = input_data_np.reshape(1,-1)
         s_data = ss.transform(input_data_np_re)

         predict = model3.predict(s_data)
         print(predict)

         if(predict[0]==0):
             print('Negative,No parkinson Found')
         else:
             print('Positive, parkinson Found')

[1]
Positive, parkinson Found
```

Fig-28

Testing the model ,the model predicted the patients Parkinson disease correctly.

## 6 Project Management

The Project began on September 12 and ran until December 9. My ethics were accepted on October 28, 2022. The project got underway on October 30. Learning about machine learning algorithms and how to code them was the first step. The Literature Review for then began on November 10th. I began writing code for my project at the same time. My code included several issues that were resolved. I started by performing an exploratory analysis on the data, which took some time. The model analysis was then launched. We often met with Mr. David Croft, our supervisor. We would meet with him every two weeks on a Wednesday to discuss the project.

Task	Start Date	End Date	Duration
Literature Review	10/11/2022	16/11/2022	3
Running the data in the model (SVM)	17/11/2022	21/11/2022	4
Running the data in the model (Naïve Bayes)	22/11/2022	22/11/2022	4
Running the data in the model (Random Forest)	23/11/2022	23/12/2022	5
Comparison of Models	24/12/2022	24/12/2022	4
Abstract	25/12/2022	26/12/2022	2
Introduction	27/12/2022	27/12/2022	4
Methodology	28/1/2022	30/11/2022	3
Result	30/11/2022	30/11/2022	2
Conclusion	1/12/2022	2/12/2022	4
References	3/12/2022	3/12/2022	3
Final Overview of Research of Project	03/12/2022	08/12/2022	7



## 7 Conclusions

Parkinson disease is a disease that affects the human brain where over the year it progressively damages. It is caused by a loss of nerve cells in part of the brain called substantial nigra. It's a neurological disease where dopamine reduces over time. Dopamine is like a neurotransmitter and hormone that sends signals to other neurons, it is also pleasure giving hormone. Symptoms of Parkinson Disease are involuntary shaking of body, slow movement, stiff and inflexible muscles. A patient with PD will have anxiety and depression, loss of memory, patient will have sleeping problems. Most people suffer with Parkinson disease with people over the age 50. They experience symptoms when they are in their 40s. There is no cure for the treatment of Parkinson Disease, but there are treatments that are available to help reduce symptoms. Some of the treatments include physiotherapy, supportive treatments, some times brain surgery, medication. More than 10 million people worldwide are living with Parkinson Disease. So it's important to predict Parkinson disease as soon as possible.

This paper successfully predicts Parkinson Disease with patients voice measurements from 31 people. The dataset is taken from UCI machine learning Repository. The dataset was created by Max Little of the University of Oxford, in collaboration with the National Centre for Voice and speech. Dataset Contains 195 voice recordings from the patients.

Once the data is collected from the repository, the data was pre-processing where missing data was resolved. After that Exploratory Data Analysis was done, where the distribution of the data was found, the correlation was found using the heatmap. Outliers were found using boxplots. After the completion of EDA analysis, the data was ready for model analysis. Support Vector Machine, Naïve Bayes and random Forest are the algorithms to predict if the patient has Parkinson Disease. It was found that Support Vector Machine performed the best with an accuracy of 87%, followed by Random Forest with accuracy score of 82%, and naïve bayes got the score for 61%. So Support Vector Machine is the best algorithm to predict Parkinson Disease.

## **7.1 7.1 Future Works**

Since 10 millions people are affected with Parkinson disease .The number of Pd patients are expected to rise to 1.2 million by 2030 according to Parkinson's foundation statistics . Men are more likely to get diagnosed before the age of 50.The prediction model that this project has successfully created can be used a web app ,or a mobile application. So that more people can use this application, and early treatments can be started as soon as possible. This application can be used in research organization or hospitals to discover PD as soon as possible.

## Bibliography and References

*An efficient machine learning approach for diagnosing Parkinson's disease by utilizing voice features.*

(2022, November 17). MDPI. <https://www.mdpi.com/2079-9292/11/22/3782>

*Parkinson's disease resting tremor severity classification using machine learning with resampling techniques.* (n.d.). Frontiers.

<https://www.frontiersin.org/articles/10.3389/fnins.2022.955464/full>

*Vocal feature guided detection of Parkinson's disease using machine learning algorithms.* (n.d.). IEEE

Xplore. <https://ieeexplore.ieee.org/document/9965732>

[https://www.researchgate.net/publication/235421872\\_Predication\\_of\\_Parkinson's\\_disease\\_using\\_data\\_mining\\_methods\\_A\\_comparative\\_analysis\\_of\\_tree\\_statistical\\_and\\_support\\_vector\\_machine\\_classifiers](https://www.researchgate.net/publication/235421872_Predication_of_Parkinson's_disease_using_data_mining_methods_A_comparative_analysis_of_tree_statistical_and_support_vector_machine_classifiers)

Prashanth, R., Dutta Roy, S., Mandal, P. K., & Ghosh, S. (2016). High-Accuracy Detection of Early Parkinson's Disease through Multimodal Features and Machine Learning. *International Journal of Medical Informatics*, 90, 13-21. <https://doi.org/10.1016/j.ijmedinf.2016.03.001>

<https://www.sciencedirect.com/science/article/pii/S1386505616300326?via%3Dihub>

<https://pubmed.ncbi.nlm.nih.gov/27103193/>

Gamboa, J., Jiménez-Jiménez, F. J., Nieto, A., Montojo, J., Ortí-Pareja, M., Molina, J. A., García-Albea, E., & Cobeta, I. (1997). Acoustic voice analysis in patients with Parkinson's disease treated with

dopaminergic drugs. *Journal of Voice*, 11(3), 314-320. [https://doi.org/10.1016/S0892-1997\(97\)80010-0](https://doi.org/10.1016/S0892-1997(97)80010-0)

[https://www.jvoice.org/article/S0892-1997\(97\)80010-0/pdf](https://www.jvoice.org/article/S0892-1997(97)80010-0/pdf)

<https://ieeexplore.ieee.org/document/7167480>

Park, Y.H., Suh, J.H., Kim, Y.W. *et al.* Machine learning based risk prediction for Parkinson's disease with nationwide health screening data. *Sci Rep* **12**, 19499 (2022).

<https://doi.org/10.1038/s41598-022-24105-9>

Templeton, J.M., Poellabauer, C. & Schneider, S. Classification of Parkinson's disease and its stages using machine learning. *Sci Rep* **12**, 14036 (2022). <https://doi.org/10.1038/s41598-022-18015-z>

<https://doi.org/10.1038/s41598-022-18015-z>

<https://www.nature.com/articles/s41598-022-18015-z>

<https://www.proquest.com/docview/2661072826>

da Rosa Tavares, J. E., Ullrich, M., Roth, N., Kluge, F., Eskofier, B. M., Gaßner, H., Klucken, J., Gladow, T., Marxreiter, F., da Costa, C. A., da Rosa Righi, R., & Victória Barbosa, J. L. (2023). uTUG: An unsupervised Timed Up and Go test for Parkinson's disease. *Biomedical Signal Processing and Control*, 81, 104394. <https://doi.org/10.1016/j.bspc.2022.104394>

<https://www.sciencedirect.com/science/article/pii/S1746809422008485?via%3Dihub>

Nilashi, M., Abumalloh, R. A., Yusuf, S. Y. M., Thi, H. H., Alsulami, M., Abosag, H., Alyami, S., & Alghamdi, A. (2023). Early diagnosis of Parkinson's disease: A combined method using deep learning and neuro-fuzzy techniques. *Computational Biology and Chemistry*, 102, 107788.

<https://doi.org/10.1016/j.compbiolchem.2022.107788>

<https://www.sciencedirect.com/science/article/pii/S0306987719314148?via%3Dihub>

<https://www.sciencedirect.com/science/article/pii/S1476927122001682?via%3Dihub>

Chawla, P., Rana, S. B., Kaur, H., Singh, K., Yuvaraj, R., & Murugappan, M. (2023). A decision support system for automated diagnosis of Parkinson's disease from EEG using FAWT and entropy features.

*Biomedical Signal Processing and Control*, 79, 104116. <https://doi.org/10.1016/j.bspc.2022.104116>

<https://www.sciencedirect.com/science/article/pii/S1746809422005730?via%3Dihub>

Prashanth, R., Dutta Roy, S., Mandal, P. K., & Ghosh, S. (2016). High-Accuracy Detection of Early Parkinson's Disease through Multimodal Features and Machine Learning. *International Journal of Medical Informatics*, 90, 13-21. <https://doi.org/10.1016/j.ijmedinf.2016.03.001>

Lim, W.S., Chiu, S.I., Wu, M.C. *et al.* An integrated biometric voice and facial features for early detection of Parkinson's disease. *npj Parkinsons Dis.* **8**, 145 (2022).

<https://doi.org/10.1038/s41531-022-00414-8>

Zhang, J. Mining imaging and clinical data with machine learning approaches for the diagnosis and early detection of Parkinson's disease. *npj Parkinsons Dis.* **8**, 13 (2022).

<https://doi.org/10.1038/s41531-021-00266-8>

Wang, Q., Fu, Y., Shao, B., Chang, L., Ren, K., Chen, Z., & Ling, Y. (2022). Early detection of Parkinson's disease from multiple signal speech: Based on Mandarin language dataset. *Frontiers in Aging Neuroscience*, *14*. <https://doi.org/10.3389/fnagi.2022.1036588>

<https://www.frontiersin.org/articles/10.3389/fnagi.2022.1036588/full>

Rana, A., Dumka, A., Singh, R., Rashid, M., Ahmad, N., & Panda, M. K. (2022). An Efficient Machine Learning Approach for Diagnosing Parkinson's Disease by Utilizing Voice Features. *Electronics*, *11*(22), 3782.

<https://doi.org/10.3390/electronics11223782>

Channa, A., Cramariuc, O., Memon, M., Popescu, N., Mammone, N., & Ruggeri, G. (2022).

Parkinson's disease resting tremor severity classification using machine learning with resampling techniques. *Frontiers in Neuroscience*, *16*. <https://doi.org/10.3389/fnins.2022.955464>

*Parkinson's disease - Symptoms and causes.* (2022, July 8). Mayo

Clinic. <https://www.mayoclinic.org/diseases-conditions/parkinsons-disease/symptoms-causes/syc-20376055>

Zhang, J. Mining imaging and clinical data with machine learning approaches for the diagnosis and early detection of Parkinson's disease. *npj Parkinsons Dis.* **8**, 13 (2022).

<https://doi.org/10.1038/s41531-021-00266-8><https://pubmed.ncbi.nlm.nih.gov/35064123/>

M. Mamun, M. I. Mahmud, M. I. Hossain, A. M. Islam, M. S. Ahammed and M. M. Uddin, "Vocal Feature Guided Detection of Parkinson's Disease Using Machine Learning Algorithms," *2022 IEEE*

*13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2022, pp. 0566-0572, doi: 10.1109/UEMCON54665.2022.9965732.

Park, Y.H., Suh, J.H., Kim, Y.W. *et al.* Machine learning based risk prediction for Parkinson's disease with nationwide health screening data. *Sci Rep* **12**, 19499 (2022).

<https://doi.org/10.1038/s41598-022-24105-9>

Gupta, D., Julka, A., Jain, S., Aggarwal, T., Khanna, A., Arunkumar, N., & de Albuquerque, V. H. C. (2018). Optimized cuttlefish algorithm for diagnosis of Parkinson's disease. *Cognitive Systems Research*, 52, 36-48. <https://doi.org/10.1016/j.cogsys.2018.06.006>

1.4. *Support vector machines*. (n.d.). scikit-learn. Retrieved December 9, 2022, from <https://scikit-learn.org/stable/modules/svm.html>

*5 types of classification algorithms in machine learning*. (2020, August 26). MonkeyLearn Blog. <https://monkeylearn.com/blog/classification-algorithms/>

*Python predict() function with examples*. (2022, January 3). Python Programs. <https://python-programs.com/python-predict-function-with-examples/>

7.1.6. *What are outliers in the data?* (n.d.).

<https://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm#:~:text=An%20outlier%20is%20an%20observation,what%20will%20be%20considered%20abnormal>

Admin. (2020, October 2). *Who is most at risk for Parkinson's disease?* Evolution Research Group. <https://joinaresearchstudy.com/2020/10/02/who-is-most-at-risk-for-parkinsons-disease/>

Aznar, P. (2020, December 13). *Decision trees: Gini vs entropy* ★ Quantdare. Quantdare.

<https://quantdare.com/decision-trees-gini-vs-entropy/>

Cristol, H. (2019, June 19). *What is dopamine?* WebMD. [https://www.webmd.com/mental-](https://www.webmd.com/mental-health/what-is-dopamine)

[health/what-is-dopamine](https://www.webmd.com/mental-health/what-is-dopamine)

*Just a moment...* (n.d.). Just a moment... [https://machinelearningmastery.com/machine-learning-in-](https://machinelearningmastery.com/machine-learning-in-python-step-by-step/)

[python-step-by-step/](https://machinelearningmastery.com/machine-learning-in-python-step-by-step/)

*Speech therapy and PD.* (n.d.). Parkinson's Foundation. [https://www.parkinson.org/library/fact-](https://www.parkinson.org/library/fact-sheets/speech-therapy?gclid=&utm_campaign=&utm_medium=adgrant&utm_source=google&utm_term=)

[sheets/speech-](https://www.parkinson.org/library/fact-sheets/speech-therapy?gclid=&utm_campaign=&utm_medium=adgrant&utm_source=google&utm_term=)

[therapy?gclid=&utm\\_campaign=&utm\\_medium=adgrant&utm\\_source=google&utm\\_term=](https://www.parkinson.org/library/fact-sheets/speech-therapy?gclid=&utm_campaign=&utm_medium=adgrant&utm_source=google&utm_term=)

*A five-step guide for conducting exploratory data analysis.* (2021, April 28). Shopify.

<https://shopify.engineering/conducting-exploratory-data-analysis>

Gandhi, R. (2018, July 5). *Support vector machine — Introduction to machine learning algorithms.*

Medium. [https://towardsdatascience.com/support-vector-machine-introduction-to-](https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47)

[machine-learning-algorithms-934a444fca47](https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47)

Great Learning Team. (2022, October 25). *Random forest algorithm in machine learning: An overview.* Great Learning Blog: Free Resources what Matters to shape your Career!.

<https://www.mygreatlearning.com/blog/random-forest-algorithm/>

*Introduction to NumPy.* (n.d.). W3Schools Online Web Tutorials.

[https://www.w3schools.com/python/numpy/numpy\\_intro.asp](https://www.w3schools.com/python/numpy/numpy_intro.asp)

*Just a moment...* (n.d.). Just a moment... [https://machinelearningmastery.com/naive-bayes-for-](https://machinelearningmastery.com/naive-bayes-for-machine-learning/)

[machine-learning/](https://machinelearningmastery.com/naive-bayes-for-machine-learning/)



Khanna, C. (2020, December 25). *What and why behind fit\_transform() vs transform() in scikit-learn !*

Medium. <https://towardsdatascience.com/what-and-why-behind-fit-transform-vs-transform-in-scikit-learn-78f915cf96fe>

Koehrsen, W. (2018, April 6). *Visualizing data with pairs plots in Python*. Medium.

<https://towardsdatascience.com/visualizing-data-with-pair-plots-in-python-f228cf529166#:~:text=A%20pairs%20plot%20allows%20us,are%20easily%20implemented%20in%20Python!>

KOWALCZYK, A. (2014, October 19). *Linear kernel: Why is it recommended for text classification ?*

SVM Tutorial. <https://www.svm-tutorial.com/2014/10/svm-linear-kernel-good-text-classification/>

*Matplotlib*. (2022, October 6). Wikipedia, the free encyclopedia. Retrieved December 9, 2022, from

<https://en.wikipedia.org/wiki/Matplotlib>

*Naive Bayes classifier*. (2022, October 29). Wikipedia, the free encyclopedia. Retrieved December 9,

2022, from [https://en.wikipedia.org/wiki/Naive\\_Bayes\\_classifier](https://en.wikipedia.org/wiki/Naive_Bayes_classifier)

*Naive Bayes explained: Function, advantages & disadvantages, applications in 2023*. (2022,

November 22). upGrad blog. <https://www.upgrad.com/blog/naive-bayes-explained/>

*Parkinson's symptoms*. (2022, August 26). Parkinson's UK.

<https://www.parkinsons.org.uk/information-and-support/parkinsons-symptoms>

Patil, P. (2022, May 30). *What is exploratory data analysis?* Medium.

<https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15>

*Python pandas tutorial: A complete introduction for beginners*. (n.d.). Learn Data Science - Tutorials,

Books, Courses, and More – LearnDataSci. <https://www.learndatasci.com/tutorials/python-pandas-tutorial-complete-introduction-for-beginners/>

*Python predict() function - All you need to know! - AskPython.* (2020, October 13). AskPython.

<https://www.askpython.com/python/examples/python-predict-function>

*Scikit-learn.* (2011, October 22). Wikipedia, the free encyclopedia. Retrieved December 9, 2022, from

<https://en.wikipedia.org/wiki/Scikit-learn>

*Seaborn: Statistical data visualization — Seaborn 0.12.1 documentation.* (n.d.).

<https://seaborn.pydata.org/>

*Speech therapy and PD.* (n.d.). Parkinson's Foundation. [https://www.parkinson.org/library/fact-](https://www.parkinson.org/library/fact-sheets/speech-therapy#:~:text=Research%20shows%20that%2089%20percent,hoarse%20voice%20and%20uncertain%20articulation)

[sheets/speech-](https://www.parkinson.org/library/fact-sheets/speech-therapy#:~:text=Research%20shows%20that%2089%20percent,hoarse%20voice%20and%20uncertain%20articulation)

[therapy#:~:text=Research%20shows%20that%2089%20percent,hoarse%20voice%20and%20uncertain%20articulation](https://www.parkinson.org/library/fact-sheets/speech-therapy#:~:text=Research%20shows%20that%2089%20percent,hoarse%20voice%20and%20uncertain%20articulation)

*UCI machine learning repository: Parkinsons data set.* (n.d.).

<https://archive.ics.uci.edu/ml/datasets/parkinsons>

*The ultimate guide to random forest regression.* (2022, November 15). Keboola - Connect any data source in less than 20 minutes. <https://www.keboola.com/blog/random-forest-regression>

*Using StandardScaler() function to standardize Python data.* (2022, August 3). DigitalOcean | The Cloud for Builders. <https://www.digitalocean.com/community/tutorials/standardscaler-function-in-python>

*What is anaconda? | Domino data science dictionary.* (n.d.). Domino Data Lab | Unleash Data Science at Scale. <https://www.dominodatalab.com/data-science-dictionary/anaconda>

*What is "random-state" in sklearn.model\_selection.train\_test\_split example?* (n.d.). Stack Overflow. <https://stackoverflow.com/questions/49147774/what-is-random-state-in-sklearn-model-selection-train-test-split-example>

*What is the accuracy\_score function in Sklearn?* (n.d.). Educative: Interactive Courses for Software

Developers. <https://www.educative.io/answers/what-is-the-accuracy-score-function-in-sklearn>

Yiu, T. (2021, September 29). *Understanding random forest*. Medium.

<https://towardsdatascience.com/understanding-random-forest-58381e0602d2>

*UCI machine learning repository: Parkinsons data set*. (n.d.).

<https://archive.ics.uci.edu/ml/datasets/Parkinsons/>

*Intelligent computing and optimization: Proceedings of the 3rd International Conference on*

*intelligent computing and optimization 2020 (ICO 2020) 303068153X, 9783030681531 -*

*Ebin.pub*. (2021, March 29). ebin.pub. [https://ebin.pub/intelligent-computing-and-](https://ebin.pub/intelligent-computing-and-optimization-proceedings-of-the-3rd-international-conference-on-intelligent-computing-and-optimization-2020-ico-2020-303068153x-9783030681531.html)

[optimization-proceedings-of-the-3rd-international-conference-on-intelligent-computing-](https://ebin.pub/intelligent-computing-and-optimization-proceedings-of-the-3rd-international-conference-on-intelligent-computing-and-optimization-2020-ico-2020-303068153x-9783030681531.html)

[and-optimization-2020-ico-2020-303068153x-9783030681531.html](https://ebin.pub/intelligent-computing-and-optimization-proceedings-of-the-3rd-international-conference-on-intelligent-computing-and-optimization-2020-ico-2020-303068153x-9783030681531.html)

Mungoli, A. (2022, October 11). *Identify your data's distribution*. Medium.

<https://towardsdatascience.com/identify-your-datas-distribution-d76062fc0802>

*What is "random-state" in sklearn.model\_selection.train\_test\_split example?* (n.d.). Stack Overflow.

[https://stackoverflow.com/questions/49147774/what-is-random-state-in-sklearn-model-](https://stackoverflow.com/questions/49147774/what-is-random-state-in-sklearn-model-selection-train-test-split-example)  
[selection-train-test-split-example](https://stackoverflow.com/questions/49147774/what-is-random-state-in-sklearn-model-selection-train-test-split-example)

*Python predict() function with examples*. (2022, January 3). Python Programs. [https://python-](https://python-programs.com/python-predict-function-with-examples/)

[programs.com/python-predict-function-with-examples/](https://python-programs.com/python-predict-function-with-examples/)

<http://www.jgenng.com/wp-content/uploads/2020/04/volume10-issue3-09.pdf>

<https://www.theseus.fi/bitstream/handle/10024/155257/ThesisNurbekFlokart.pdf?isAllowed=y&sequence=1>

<https://www.nature.com/articles/s41598-022-18015-z>

[https://www.mukpublications.com/resources/19.%20Asif%20Raza1\\_pagenumber.pdf](https://www.mukpublications.com/resources/19.%20Asif%20Raza1_pagenumber.pdf)

<https://machinelearningmastery.com/machine-learning-in-python-step-by-step/>

[https://jitm.ut.ac.ir/article\\_80024\\_91234a9e9483e5fd178186cffb3b7683.pdf](https://jitm.ut.ac.ir/article_80024_91234a9e9483e5fd178186cffb3b7683.pdf)

<https://pubmed.ncbi.nlm.nih.gov/36410240/>

<https://digitalcommons.wustl.edu/cgi/viewcontent.cgi?article=12391>

[https://www.researchgate.net/publication/334822271\\_A\\_review\\_of\\_feature\\_selection\\_methods\\_in\\_medical\\_applications](https://www.researchgate.net/publication/334822271_A_review_of_feature_selection_methods_in_medical_applications)

## Appendix A – Project Specification

- Project Code

```
import numpy as np
import pandas as pd
df = pd.read_csv("parkinsons.csv")
df.shape
#Finding null values in the dataset
df.isnull().sum()
df.dtypes
#Checking LAbel imbalance
import matplotlib.pyplot as plt
import seaborn as sns
temp =df['status'].value_counts()
temp_df=pd.DataFrame({'status':temp.index,'values':temp.values})
print((sns.barplot(x='status',y='values',data=temp_df)))
sns.pairplot(df)
#Finding the distribution of data

def distplots(col):
    sns.distplot(df[col])
    plt.show()

for i in list(df.columns)[1:]:
    distplots(i)
#Finding the outliers of data

def boxplots(col):
    sns.boxplot(df[col])
    plt.show()
```

```

for i in list(df.select_dtypes(exclude=['object']).columns)[1:]:
    boxplots(i)
#Finding correlation

plt.figure(figsize=(20,20))
corr=df.corr()
sns.heatmap(corr,annot=True)

```

### Model Analysis

```

from sklearn import svm
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

df = pd.read_csv("parkinsons.csv")
df.info()
df.describe()
df.shape
df.isnull().sum()
df['status'].value_counts()
df.groupby('status').mean()
X = df.drop(columns=['name', 'status'],axis=1)
Y = df['status']
print(X)
print(Y)
X_train,X_test,Y_train,Y_test=train_test_split(X, Y
,test_size=0.2,random_state=2)
print(X_train.shape,X_test.shape)
ss = StandardScaler()
ss.fit(X_train)
X_train = ss.transform(X_train)
X_test = ss.transform(X_test)
model = svm.SVC(kernel = 'linear')

model.fit(X_train,Y_train)

X_train_pre = model.predict(X_train)
train_pre_accu = accuracy_score(Y_train,X_train_pre)
print('accuracy of training data:',train_pre_accu)
X_test_pre = model.predict(X_test)
test_pre_accu = accuracy_score(Y_test,X_test_pre)
print('accuracy of testing data:',test_pre_accu)
input_data =
(202.26600,211.60400,197.07900,0.00180,0.000009,0.00093,0.00107,0.00278,0.0
0954,0.08500,0.00469,0.00606,0.00719,0.01407,0.00072,32.68400,0.368535,0.74
2133,-7.695734,0.178540,1.544609,0.056141)
input_data_np =np.asarray(input_data)
input_data_np_re =input_data_np.reshape(1,-1)
s_data = ss.transform(input_data_np_re)

```

```

predict = model.predict(s_data)
print(predict)

if(predict[0]==0):
    print('Negative, No parkinson Found')
else:
    print('Positive, parkinson Found')

```

### Naïve Bayes

```

clf = GaussianNB()
clf.fit(X_train , Y_train)
X_train2_pre = clf.predict(X_train)
train2_pre_accu = accuracy_score(Y_train,X_train2_pre)
print('accuracy of training data:',train2_pre_accu)
X_test2_pre = clf.predict(X_test)
test2_pre_accu = accuracy_score(Y_test,X_test2_pre)
print('accuracy of testing data:',test2_pre_accu)
input_data =
(119.99200,157.30200,74.99700,0.00784,0.00007,0.00370,0.00554,0.01109,0.043
74,0.42600,0.02182,0.03130,0.02971,0.06545,0.02211,21.03300,0.414783,0.8152
85,-4.813031,0.266482,2.301442,0.284654)
input_data_np =np.asarray(input_data)
input_data_np_re =input_data_np.reshape(1,-1)
s_data = ss.transform(input_data_np_re)

predict = clf.predict(s_data)
print(predict)

```

### Random Forest

```

model3 = RandomForestClassifier(criterion='gini')
model3.fit(X_train,Y_train)
X_train3_pre = model3.predict(X_train)
train3_pre_accu = accuracy_score(Y_train,X_train3_pre)
print('accuracy of training data:',train3_pre_accu)
X_test3_pre = model3.predict(X_test)
test3_pre_accu = accuracy_score(Y_test,X_test3_pre)
print('accuracy of testing data:',test3_pre_accu)
input_data =
(119.99200,157.30200,74.99700,0.00784,0.00007,0.00370,0.00554,0.01109,0.043
74,0.42600,0.02182,0.03130,0.02971,0.06545,0.02211,21.03300,0.414783,0.8152
85,-4.813031,0.266482,2.301442,0.284654)
input_data_np =np.asarray(input_data)
input_data_np_re =input_data_np.reshape(1,-1)
s_data = ss.transform(input_data_np_re)

predict = model3.predict(s_data)
print(predict)

if(predict[0]==0):
    print('Negative, No parkinson Found')
else:
    print('Positive, parkinson Found')

```

## **Appendix B– Ethics Certificate**





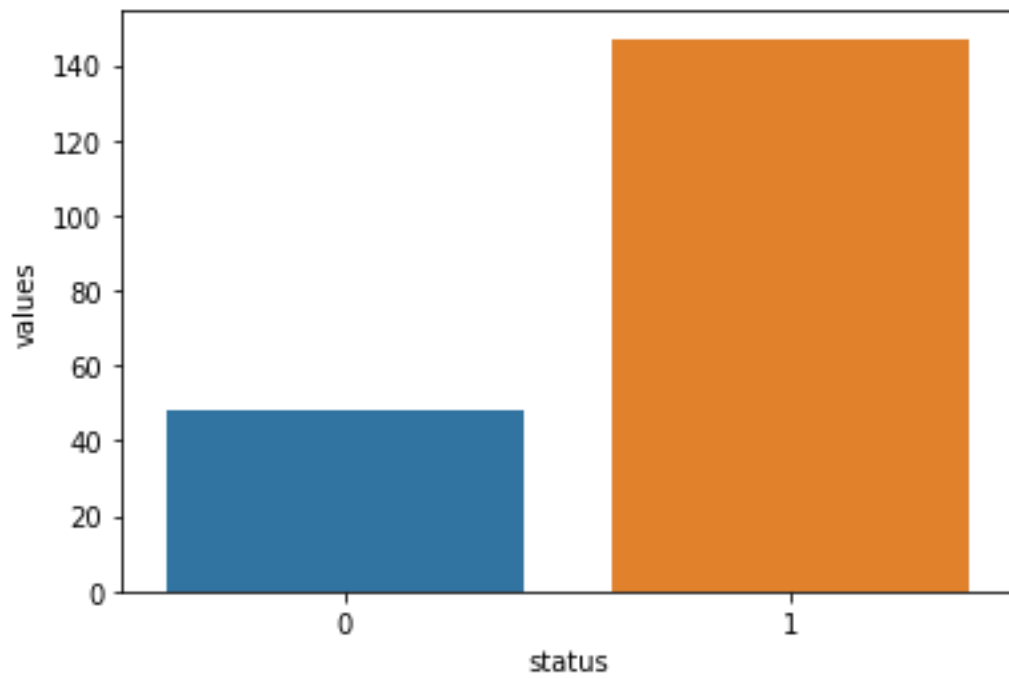
## **Certificate of Ethical Approval**

Applicant: Glen John  
Project Title: Detection of Parkinson's Disease using Machine Learning algorithms.

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

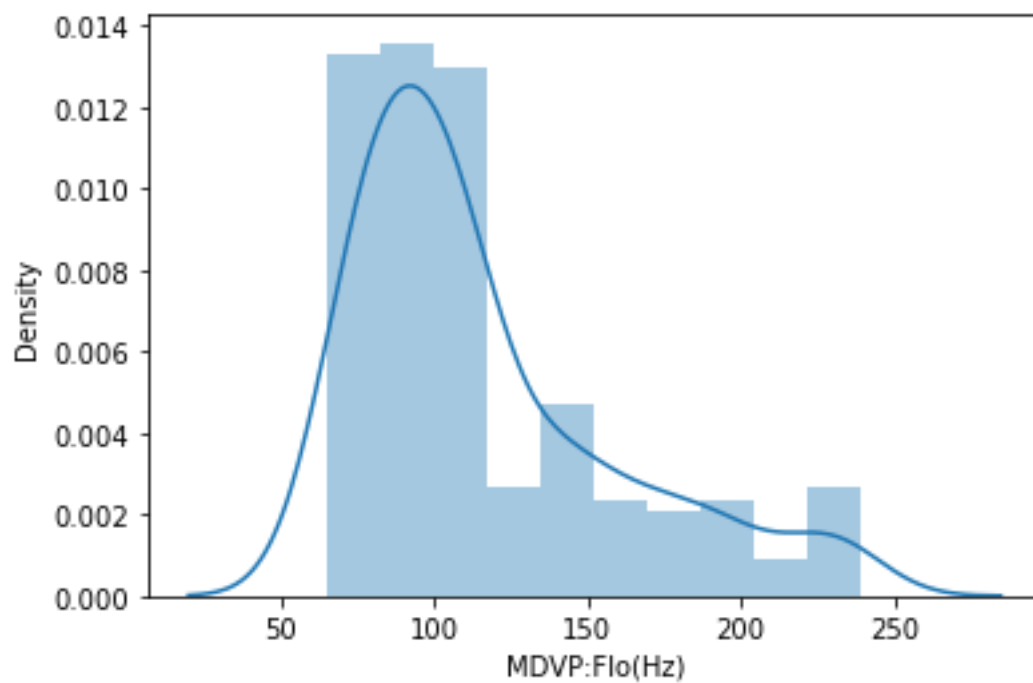
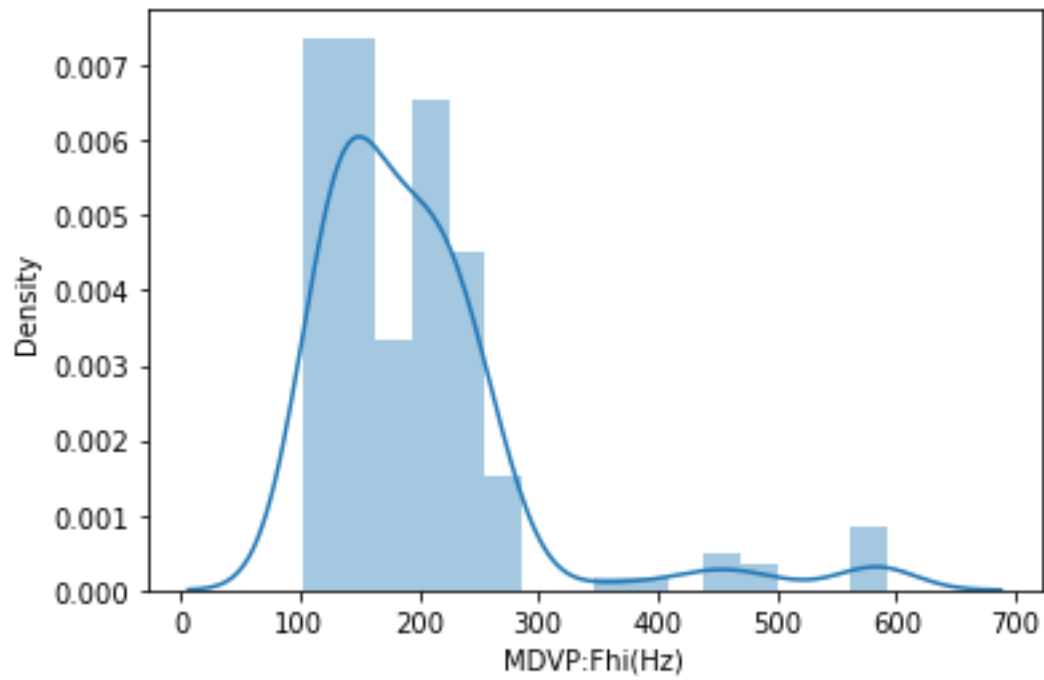
Date of approval: 28 Oct 2022  
Project Reference Number: P142574

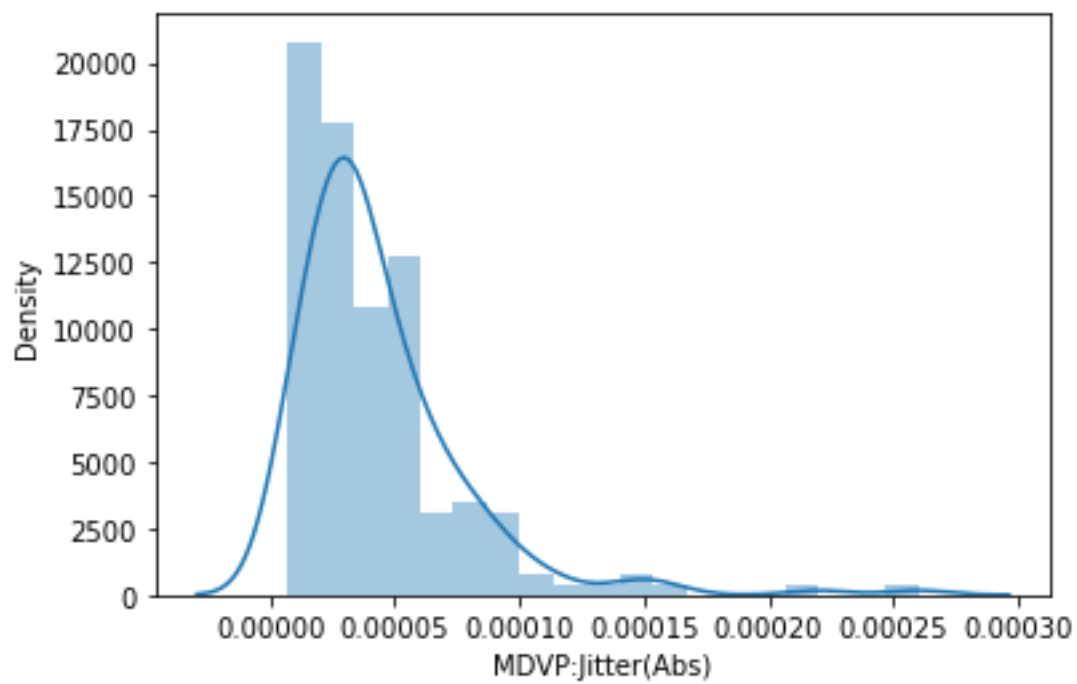
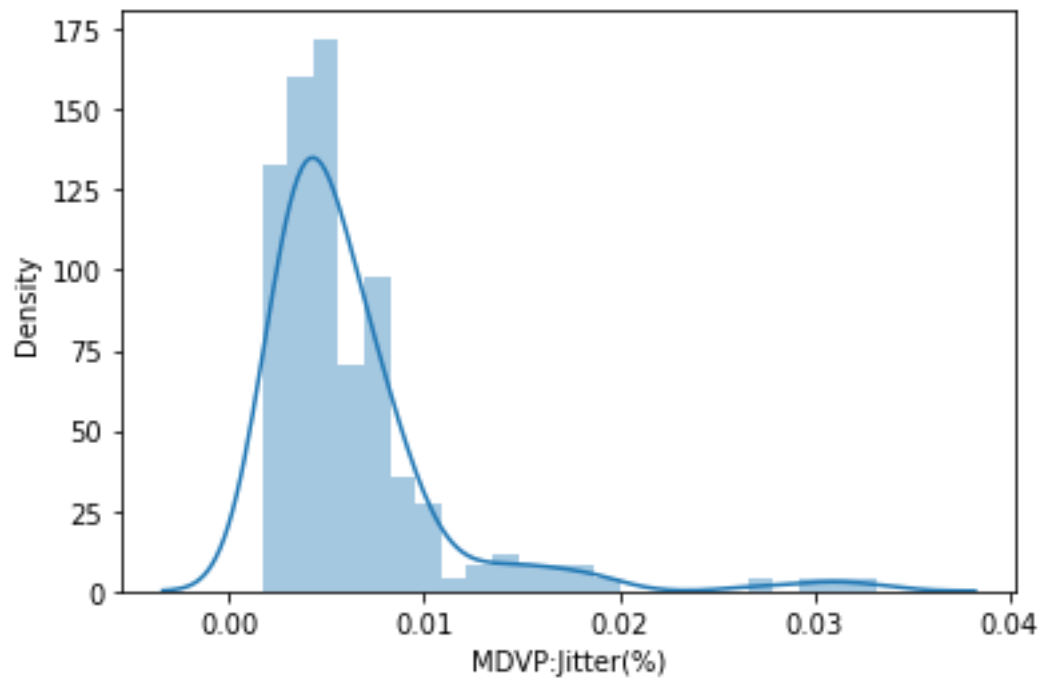
## **Appendix C– Images**

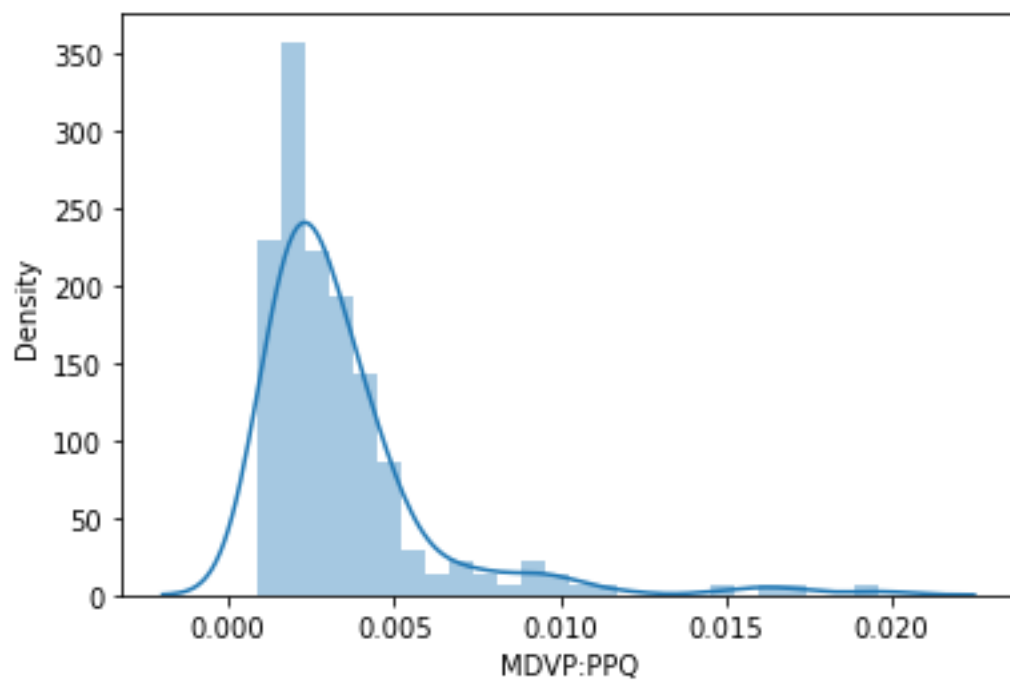
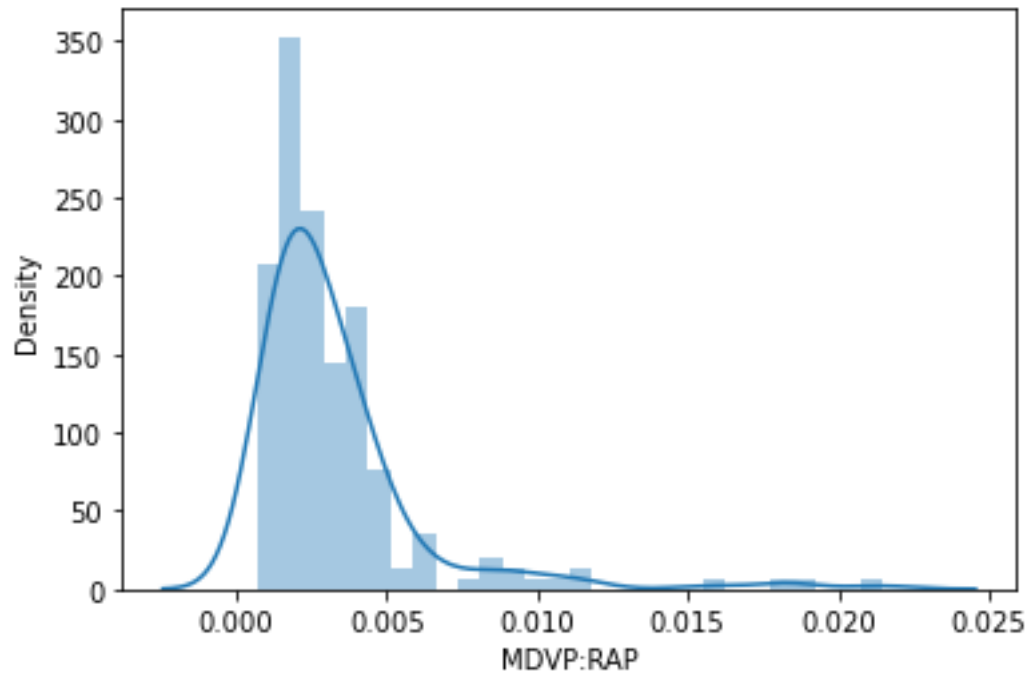


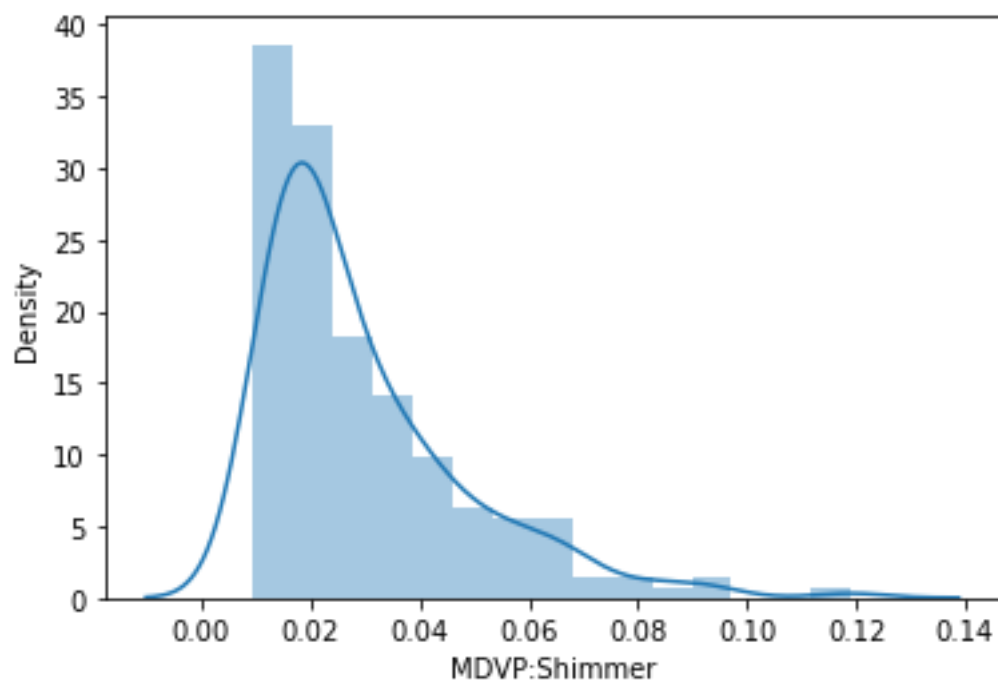
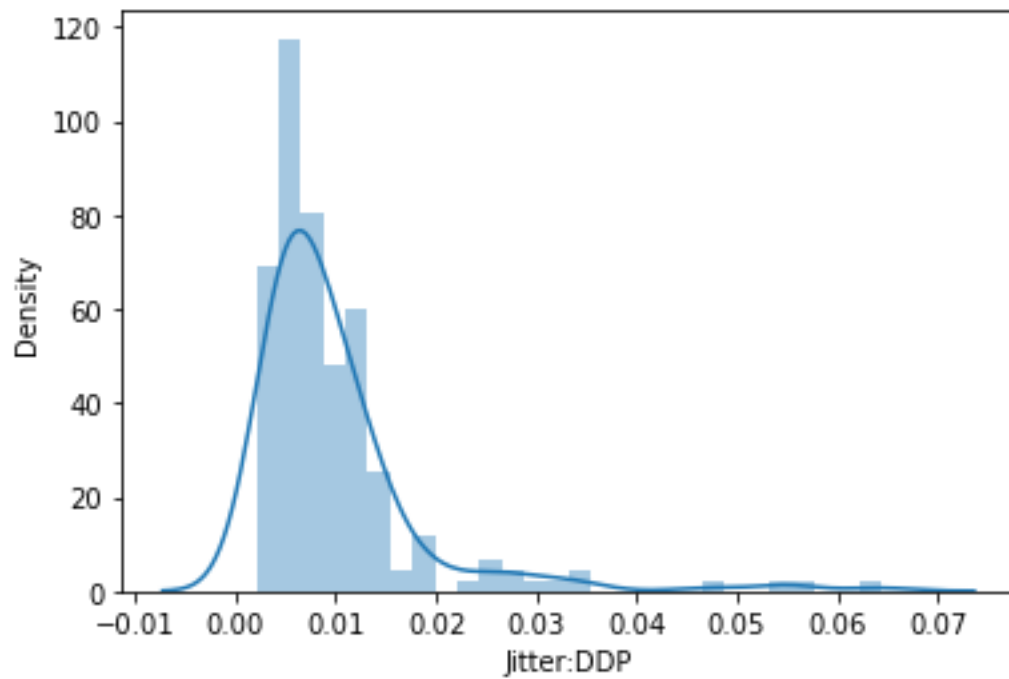


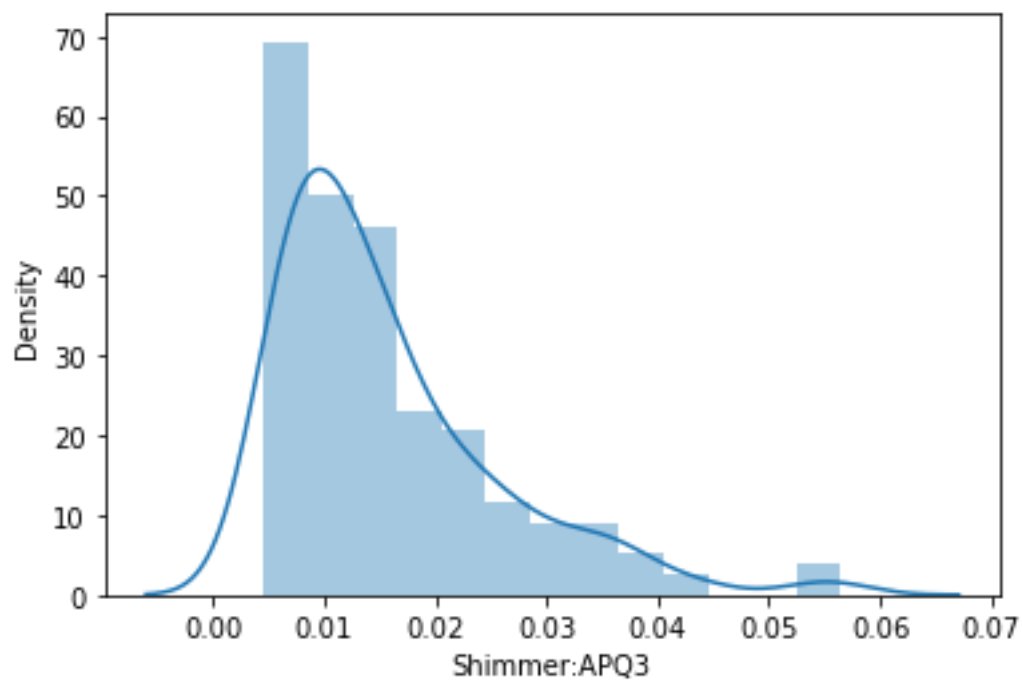
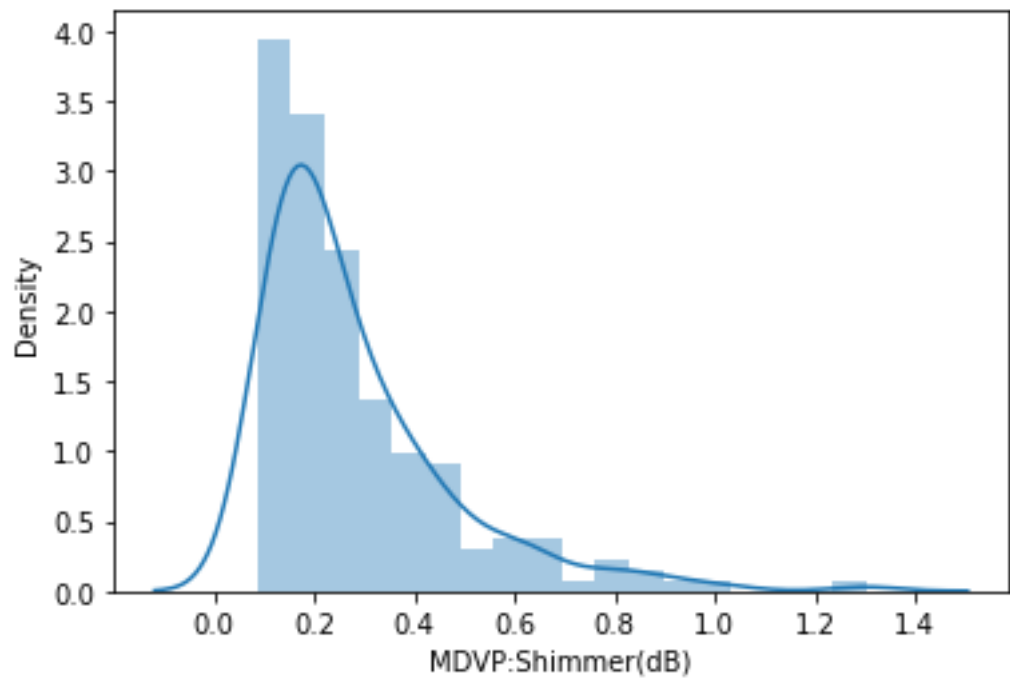
## Distribution of data



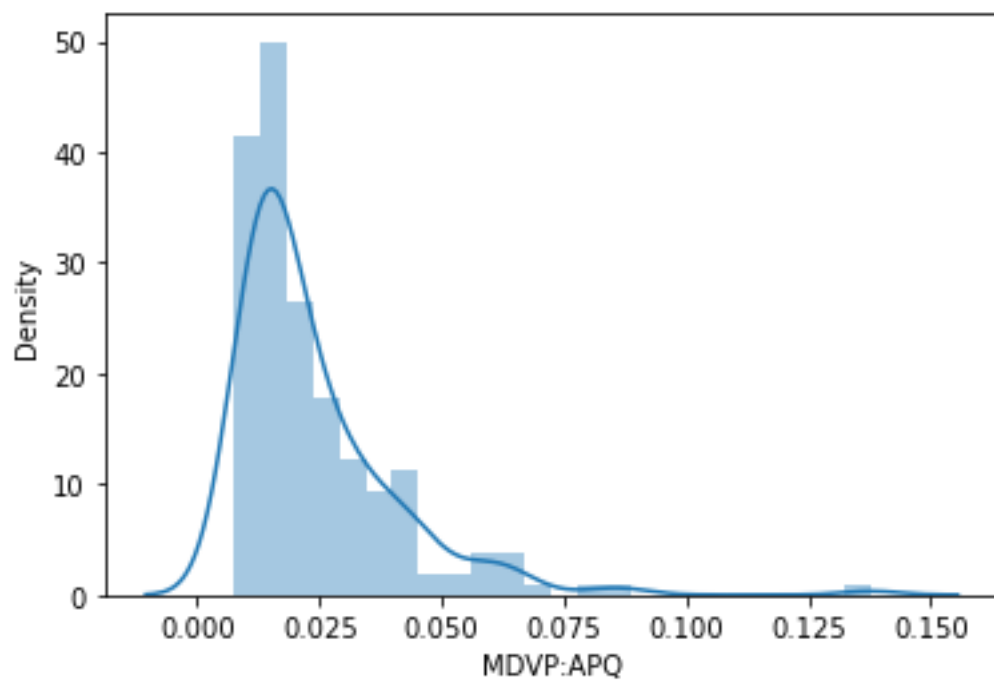
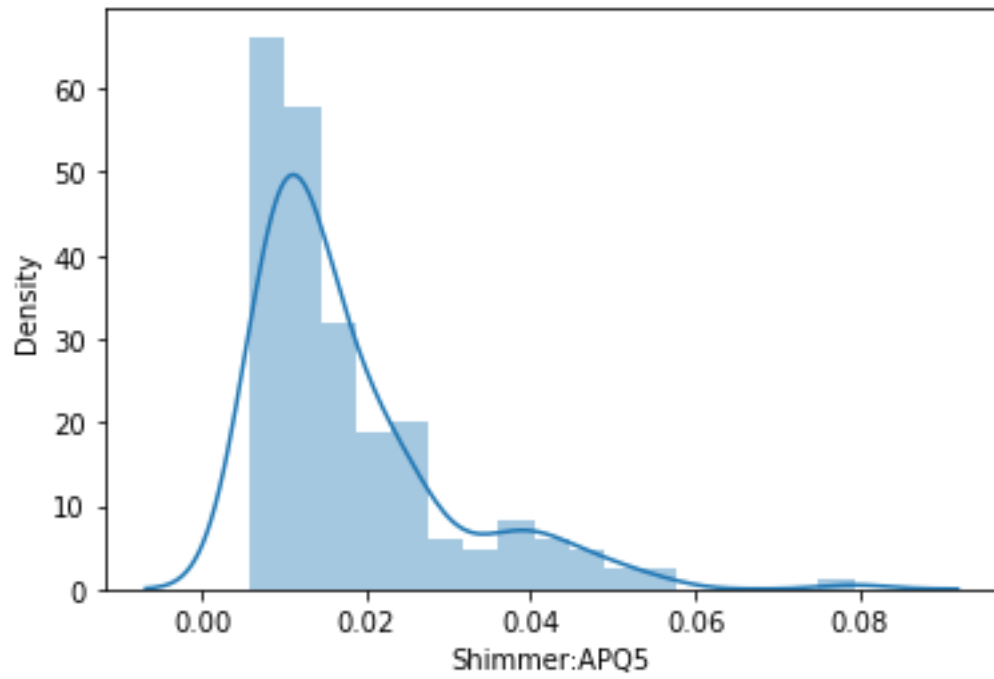


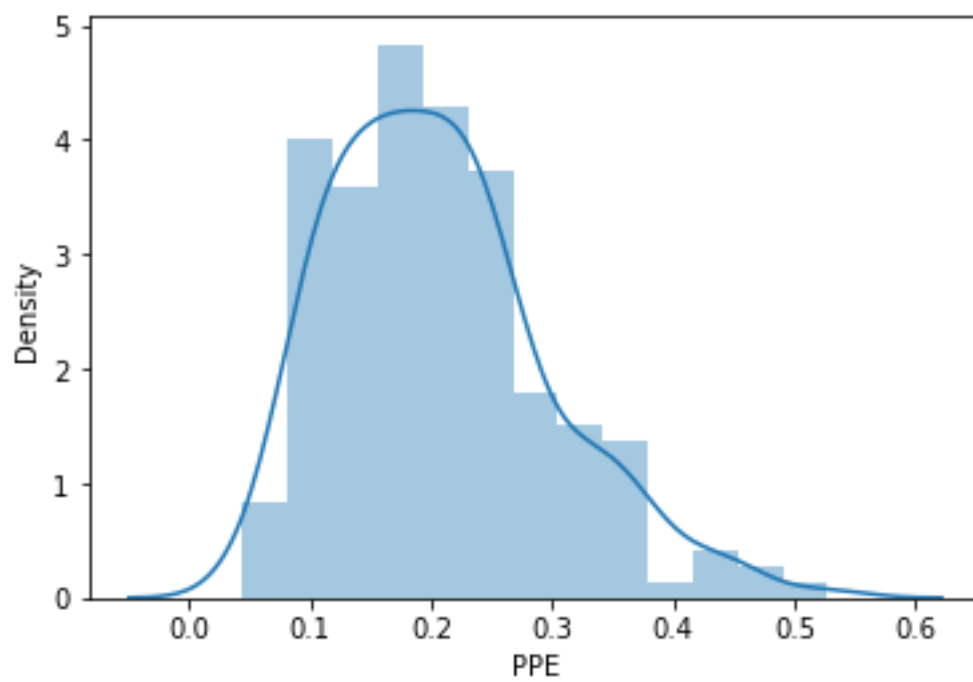
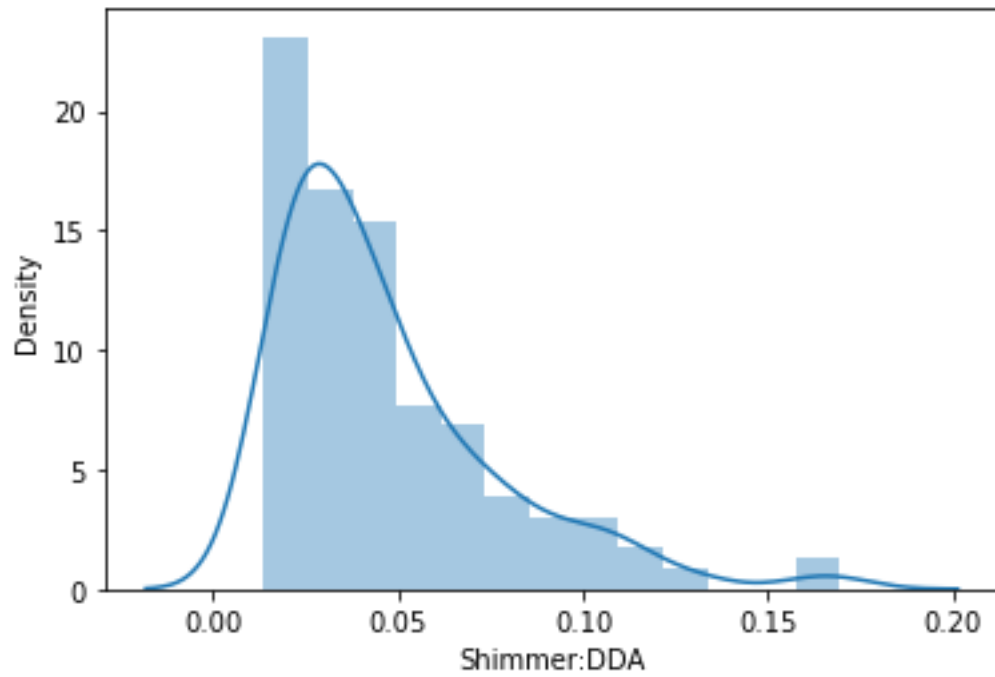












## Boxplots of the Data

