

**INTRODUCTION TO DATA SCIENCE**

**BITI2513**

**CAPSTONE PROJECT:**

**(Parkinson Disease Detection)**

|  |  |
| --- | --- |
| **NAME:** | **MATRIC NUM:** |
| Shamhinidevi Jayabalan | B031820098 |
| Ain Shazryna Binti Abdul Rahim | B031810157 |
| Azeera Fazreenaa Binti Aminuddin | B031810411 |
| Muhammad Farid bin Ed Noor | B031810455 |

**INTRODUCTION**

Parkinson’s Disease (PD) is a progressive neurodegenerative movement disease affecting approximately 2% of people at the age of 65 and is the most second most commonly occurring neurodegenerative disease in the elderly (after Alzheimer’s Disease), with more than 6.3 million people worldwide with PD. In PD sufferers, loss of dopamine-producing neurons results in a range of both motor and non-motor symptoms and currently there is no cure, no means of slowing the disease progression, and no means of prevention. From the perspective of patient quality of life, PD is one of the most severe of all chronic diseases.

Therefore, in this project we will implement algorithm in python language to detect the presence of PD.

**OBJECTIVE**

The objective of this project is:

1. To build a model to accurately detect the presence of Parkinson’s disease in an individual.
2. To prevent the disease from being worsen.
3. To analyze the Parkinson disease that affect individual.

**GOALS**

It has 5 stages to it and affects more than 1 million individuals every year. Generally, by the time of diagnosis of Parkinson Disease (PD), the disease is already well advanced, significant neuron loss and damage has already occurred, and any possibility of delaying further disease progression or providing neuroprotection is unlikely. The goal must be to diagnose and treat PD well before the irreversible destructive changes have taken place, ideally at least 5 years earlier than is currently the case. In addition, because the most severe symptoms occur in the advanced stages of the disease, strategies aimed at early detection and treatment will have the most benefit.

The detection presence of Parkinson’s disease in individuals using various factors can provide faster result without waiting for a longer time.

**QUESTION**

1. What is the scope of the disease and how many of them in total?
2. Which trusted sources are willing to cooperate in providing the quality dataset?
3. How many experts needed in order to make an early classification of the disease?
4. How long this system will be relevance due to the rapid growth of technology?

**SUCCESS**

The system can be finalized as success when it performs a very good diagnosis in term of detection Parkinson’s disease as well as generating a proper reporting.

**MEASUREMENT**

The main measurement of this system is the Accuracy. The higher the accuracy we get the better the result will been conclude. The accuracy is important because it will affect our analyzing of Parkinson disease.

**DATA SOURCES**

Our first data will be has 24 columns and 195 records and is only 39.7 KB. Usually our data can get from the kaggle or we can get the data from the hospital but need the approval from the staff before we provide the data. Second data has 755 columns and 756 records. We downloaded this data from UCI machine learning repository.

**DATA MANAGEMENT PLAN**

**Purposed**

This Data Management Plan will be used to describe the types of data, metadata, how the our project team generate the data from various sources, experimental results, types of software that we use to analyse the existing data and other materials that need to be produced in the course of this project.

**Types of data**

Data on this project will be focused on is Detection Parkinson Disease. All data will be gathered from existing two dataset. Raw data or unprocessed data can be used as testing data later. In Data Description will brief about how each data be collected.

**Data Description**

* Data sources are from various source like Kaggle website and UCI Machine learning Repository.
* The way we integrate the data is by data cleaning for existing data. The data that suite to be reused will be assigned to make comparison our new data.
* There will be no restriction on the reused of third party data as this project for learning

**Discuss how this plan will facilitate the needed analysis**

Recent discoveries support the principle that palliative care may improve the quality of life of patients with Parkinson’s disease and those who care for them. Advance care planning, a component of palliative care, provides a vehicle through which patients, families, and clinicians can collaborate to identify values, goals, and preferences early, as well as throughout the disease trajectory, to facilitate care concordant with patient wishes. While research on this topic is abundant in other life-limiting disorders, particularly in oncology, there is a paucity of data in Parkinson’s disease and related neurological disorders. We review and critically evaluate current practices on advance care planning through the analyses of three bioethical challenges pertinent to Parkinson’s disease and propose recommendations for each.

Self – management support (SMS) is considered one of the most important activities of outpatient nurses because they have the medical knowledge as well as the close contact with those affected by disease to know how it impacts on their daily life . Self-monitoring is considered an important skill to manage daily life. Self-monitoring includes observations, assessment and monitoring of physical symptoms and activities of daily living, and cognitive processes leading to self-awareness. This information can form the basis for self-care activities and for consultation with health care professionals.

Self-monitoring is one component in the broader concept of self-management defined as the ability of the person, in union with family, community and healthcare professionals to manage symptoms, treatments, lifestyle changes and psychosocial, cultural and spiritual consequences of health conditions. The concept of self- management is frequently used in the literature concerning people living with chronic diseases. Self-management is considered a complex and multidimensional phenomenon, linked intimately in almost all patients to family management as it occurs in the context of family. There are facilitators and barriers to self-management as well as the processes leading to self-management behaviours. A person’s ability in self-management will have outcomes both for the person as well as for the relatives and health care. Self-management interventions are designed to help persons with chronic diseases deal with different aspects of the disease, such as symptom control, medication and emotional reactions of disease in daily life.

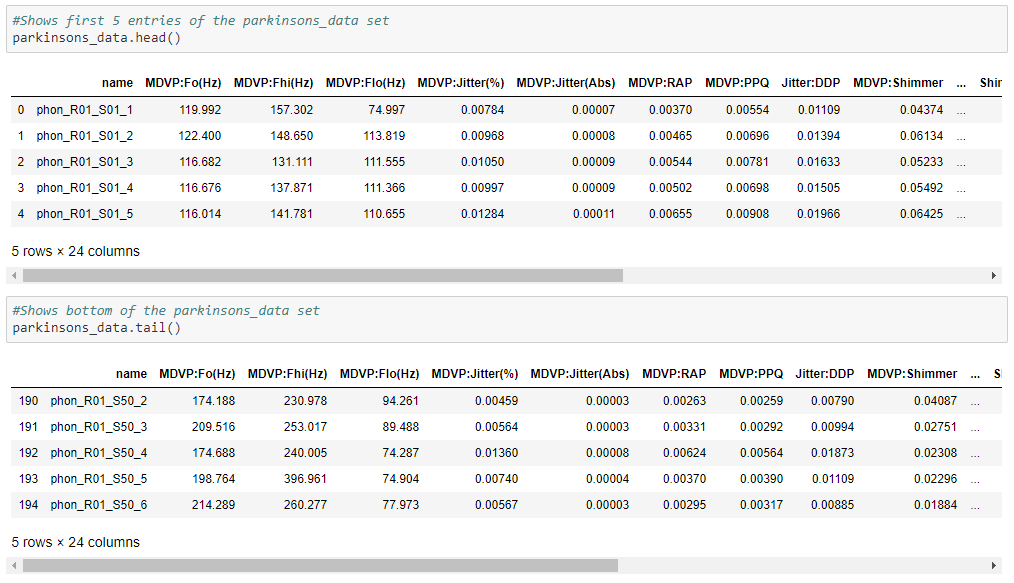
**Exploratory Data Analysis (EDA) and Data Visualization**

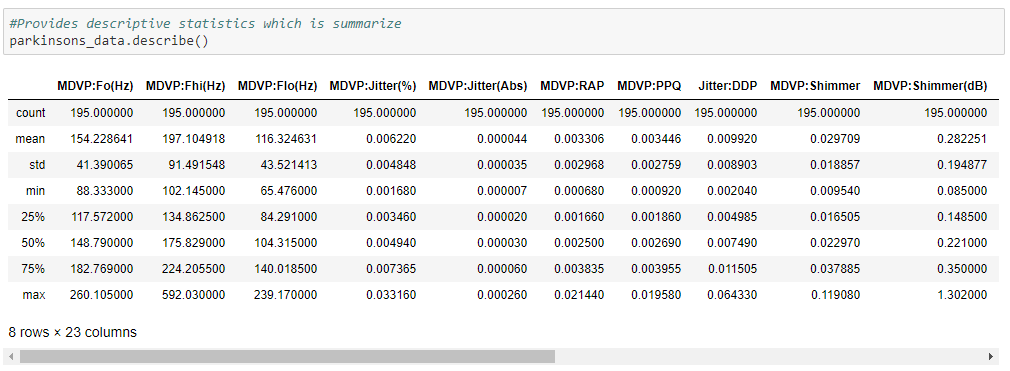
**Dataset 1: From Kaggle  
  
1) Import dataset**

To read data and performing EDA operations, primarily use the numpy and pandas Python packages, which offer simple API’s that allow us to plug our data sources and perform our desired operation.

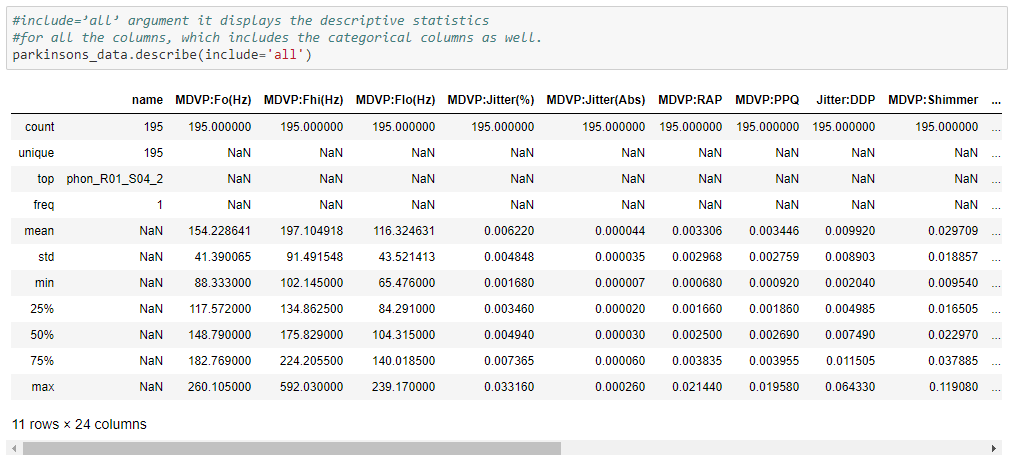
****For the output, we’ll be using the Seaborn package which is a Python-based data visualization library built on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Data visualization is an important part of analysis since it allows even non-programmers to be able to decipher trends and patterns. Followed by, load parkinsons\_data.

**2) Describing Descriptive Statistics**

After the data is loaded, have to check what's in the content of data before analysing data. The first function is the head function. This function lists the first five lines in data. Another function is the tail function. This function is a function that helps us get the last five lines in data.

Describe function returns a pandas series type that provides descriptive statistics which summarize the central tendency, dispersion, and shape of a dataset’s distribution, excluding NaN values. The three main numerical measures for the centre of a distribution are the mode, mean(µ), and the median (M). The mode is the most frequently occurring value. The mean is the average value, while the median is the middle value.

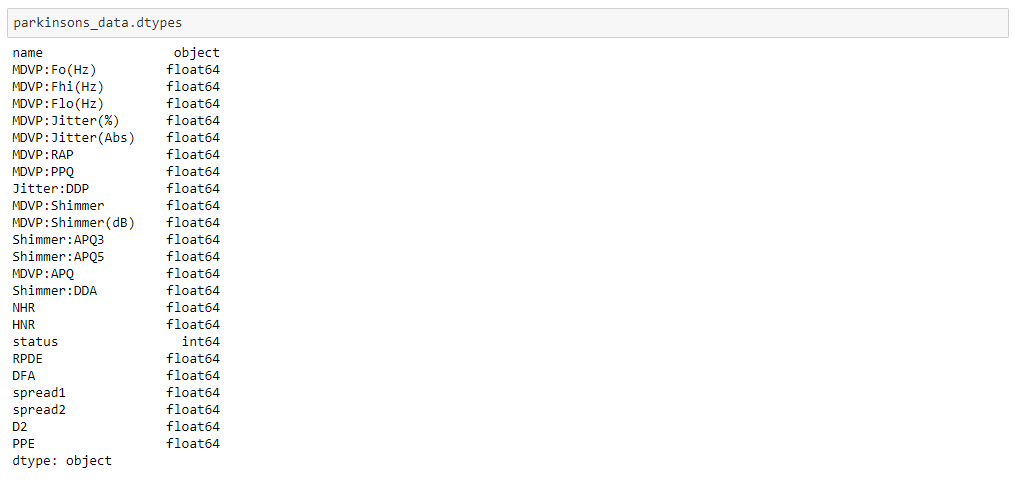
When call the describe function with include=’all’ argument it displays the descriptive statistics for all the columns, which includes the categorical columns.



This coding will print number of entries in the dataset, which are rows of Parkinson data is 195 and number of columns are 24. nunique() function will return the number of unique elements in each column.



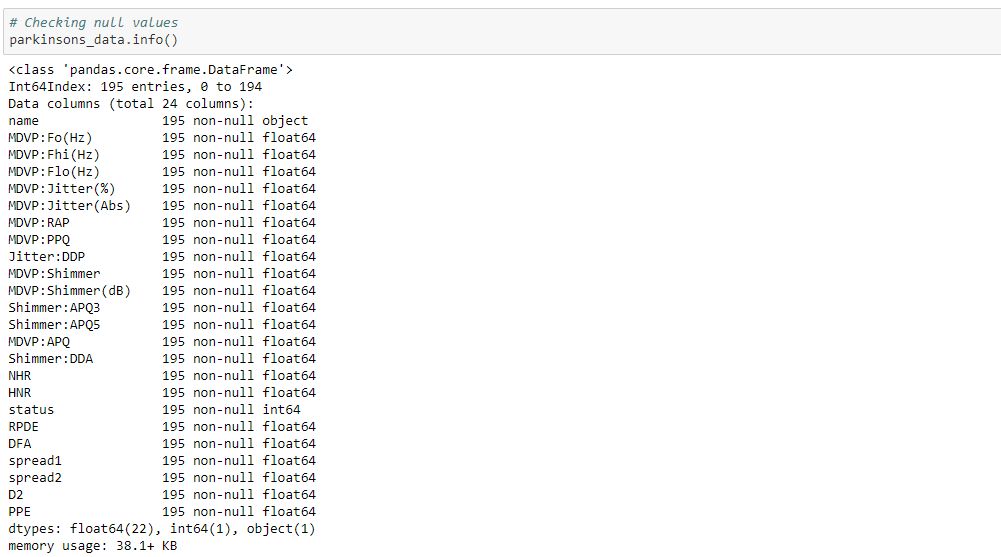
Dtypes function describe the data type of each attribute.

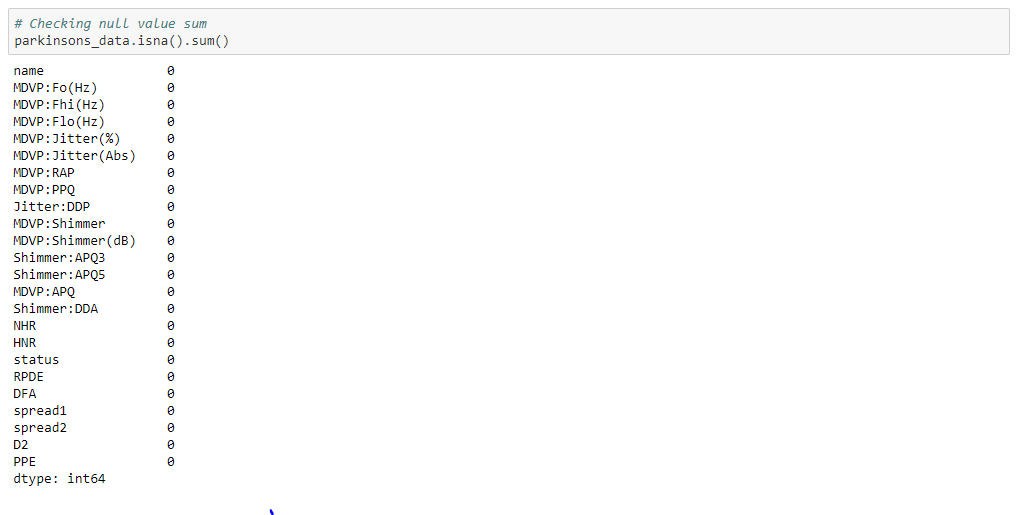


**3) Checking Null or Empty Values (Data Cleaning)**

Drop\_duplicates function checks duplicate data in datasets. In this dataset we found out that we don’t have any null value.

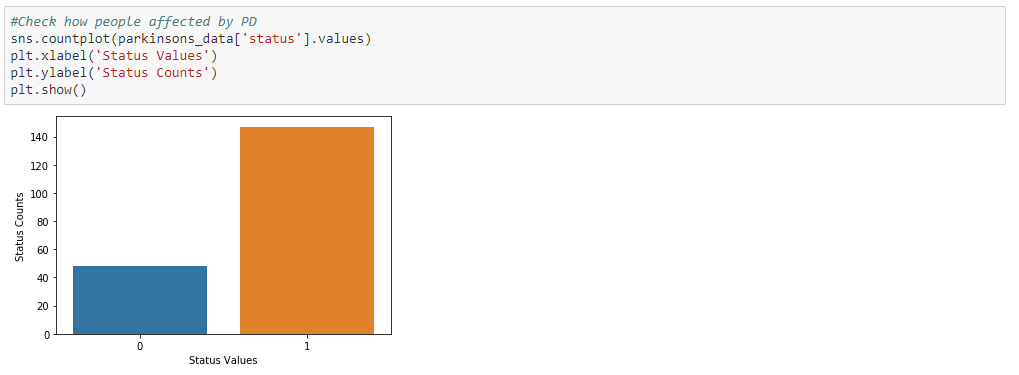






**4) Bar chart**

Histogram graph shows how many people are affected by Parkinson’s disease. 0 indicates not affected by the disease and 1 indicates affected by disease.

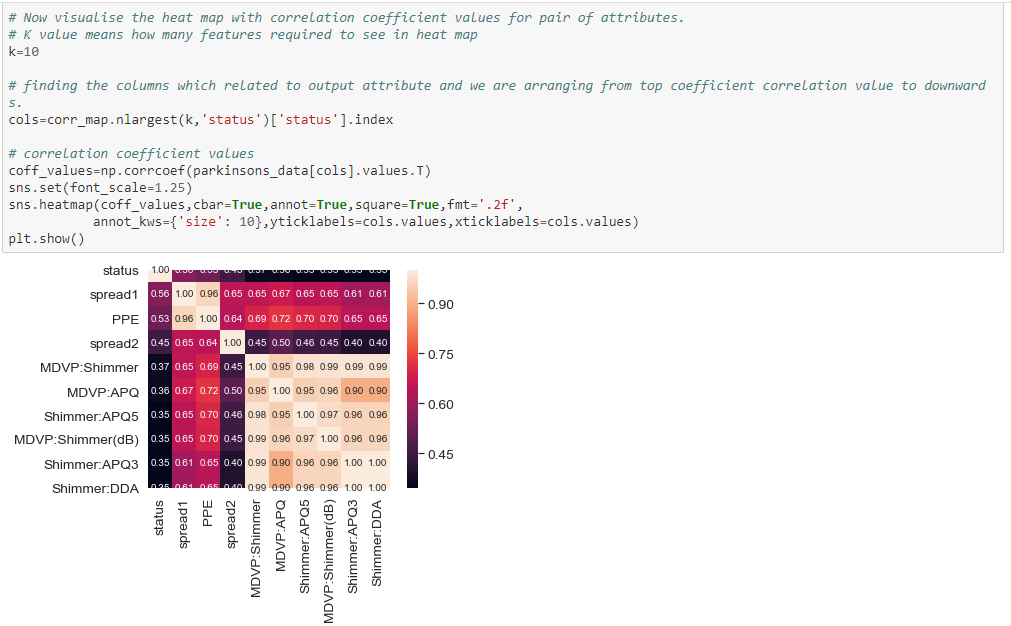


5) Visualising Descriptive Statistics

Histogram plot visualisation for each attribute will be so difficult because we have high dimensional column 23.Better, can use heat map to find the correlations coefficient values. We will remove the less correlation coefficient columns. We can remove the irrelevant features it will minimize the accuracy of an algorithm.

It will be better if we take relevant features columns then we can achieve to get good accuracy. We got correlation coefficient values for each pair of values. But we just visualized top 10 coefficient values.

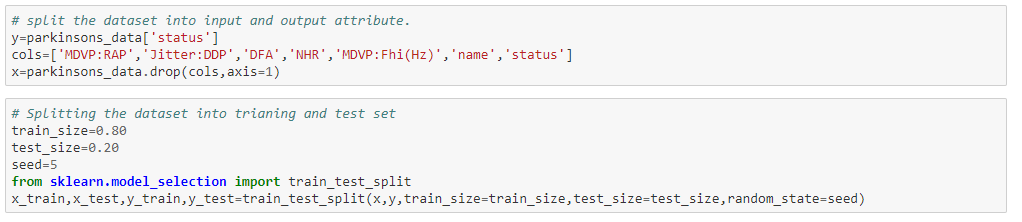






Above is the correlation values in descending order, we have correlation values in each attribute so we drop from MDVP:RAP column to MDVP:Fhi(Hz) because it have less correlation with other columns.

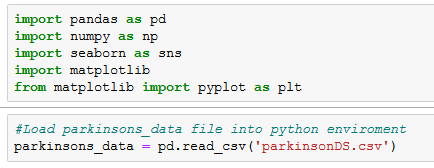
**6) Feature splitting dataset into training and test set.**



**Dataset 2: From UCI Machine Learning Repository**

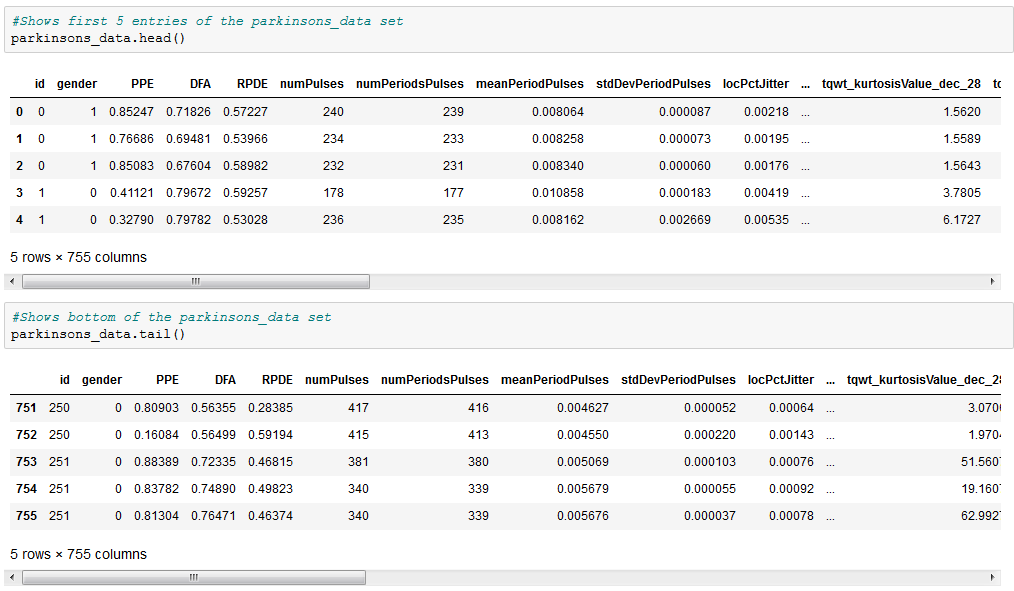
**1) Import Dataset**

The analysis have been conduct same like the previous dataset analysis and the data that been taken from UCI machine learning repository. The data have about 755 attribute and 756 rows.

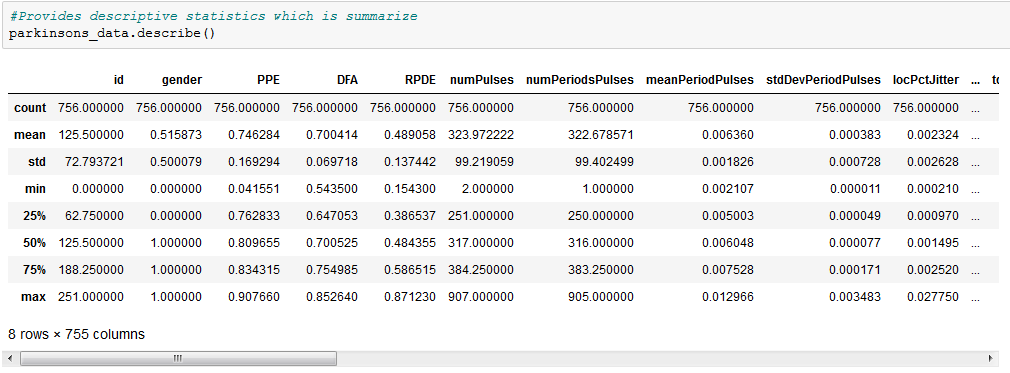


**2) Describing Descriptive Statistic**

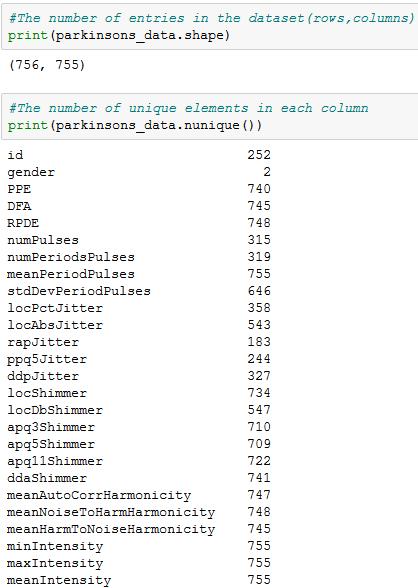
After successfully load the “parkinsonDS.csv”, we checked the content of the data using head and tails function to see how many data that we need to handle in our analysis.



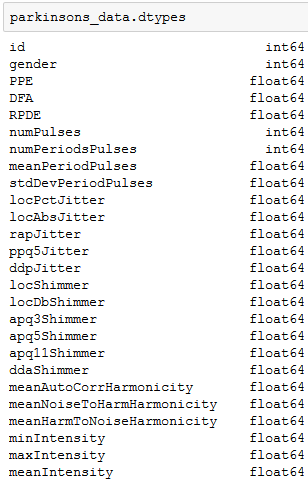
After that, we use describe function to see descriptive statistic for the dataset that been summarize to understand more about the dataset.



The shape function show how many rows and columns this dataset have and nununique() function return the number of element that unique in each of the column.



And dtypes function use to see the data type of each attribute.



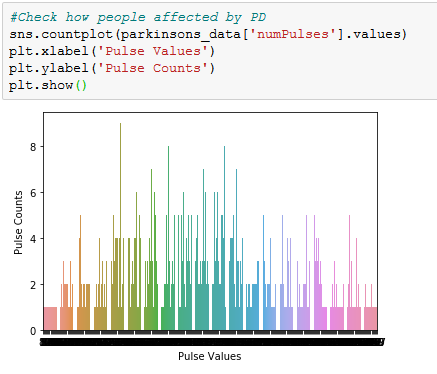
**3) Checking Null or Empty Values (Data Cleaning)**

drop\_duplicates function use to check any redundance data in the dataset.



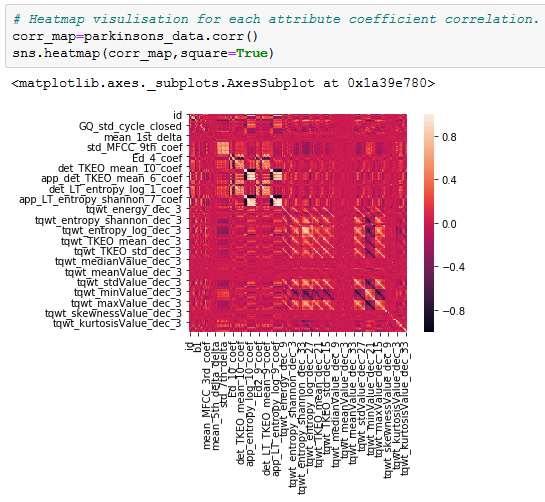
**4) Bar chart**

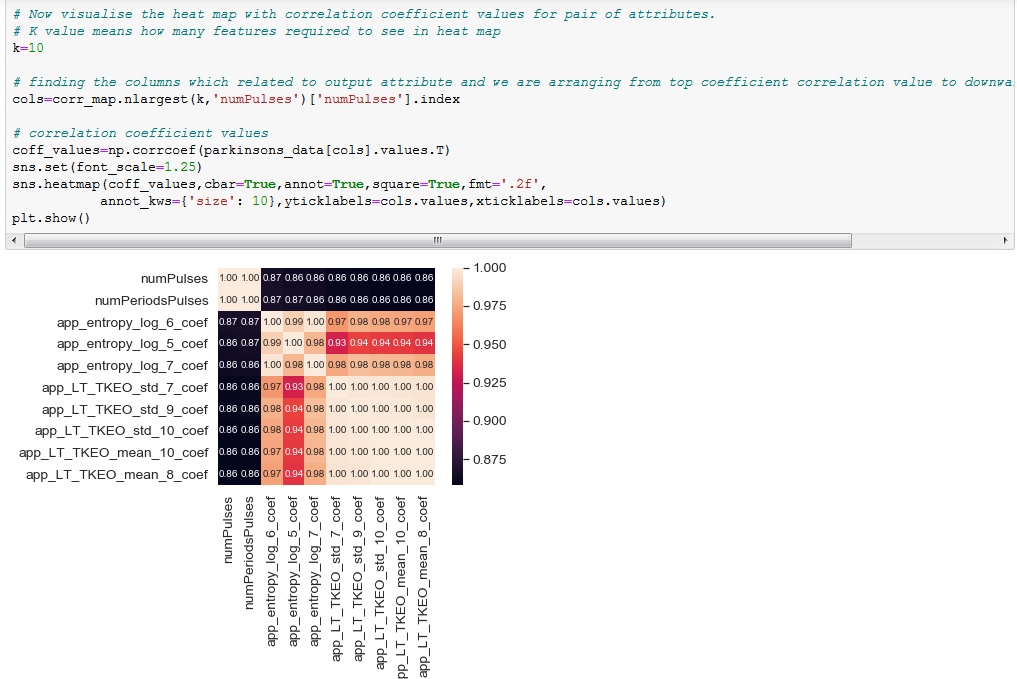
The histogram graph show the number of pulses that affect the patient that have Parkinson diseases.

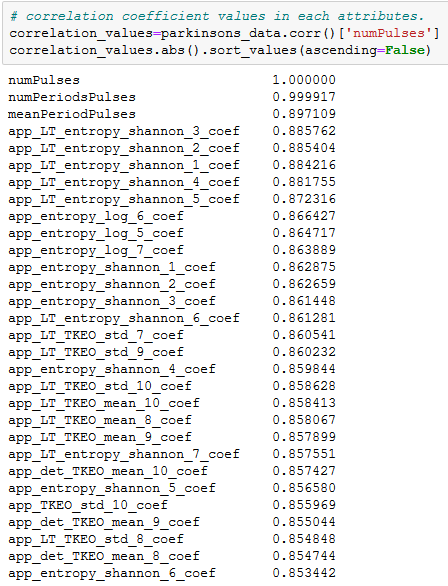


**5) Visualising Descriptive Statistic**

We use heat map to find the correlation coefficients values that will use less correlation coefficient and remove unwanted features that can minimize the accuracy.

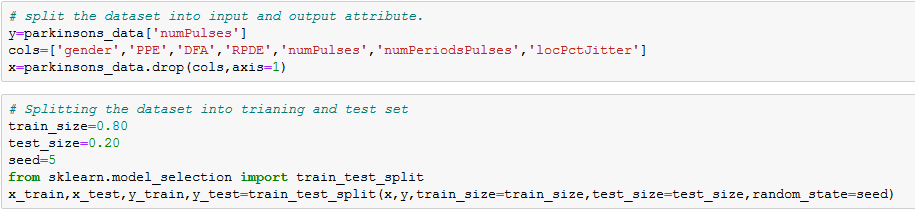






**6) Feature splitting dataset into training and test set.**

We split the dataset to do the training and testing the data. We train 80% of the data and test 20% of the data.

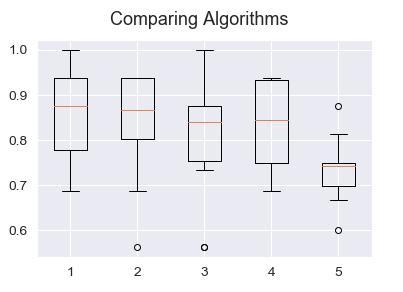


**MODELLING AND EVALUATION**

1. **Classification Models**

Prediction of models without applying feature scaling:

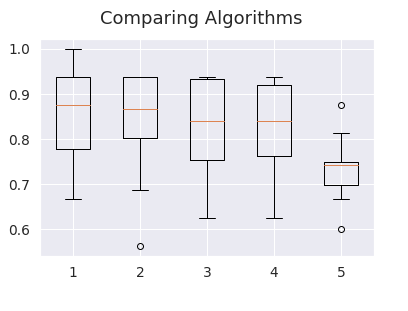
1. Logistic Regression Classification Algorithm : 0.859583 (0.114429)
2. K-Nearest Neighbour’s classification Algorithm : 0.834167 (0.118714)
3. Support Vector Machine classification Algorithm : 0.821667 (0.117951)
4. Decision Tree Classification Algorithm : 0.840000 (0.106771)
5. Naive bayes Classification Algorithm : 0.735833 (0.071715)



1. **Classification Models With Feature Scale.**

Prediction we got with applying feature scaling:

1. Logistic Regression Classification Algorithm : 0.859583 (0.114429)
2. K-Nearest neighbour’s classification Algorithm : 0.834167 (0.118714)
3. Support Vector Machine classification Algorithm : 0.821667 (0.117951)
4. Decision Tree Classification Algorithm : 0.865833 (0.076508)
5. Naive bayes Classification Algorithm : 0.735833 (0.071715)



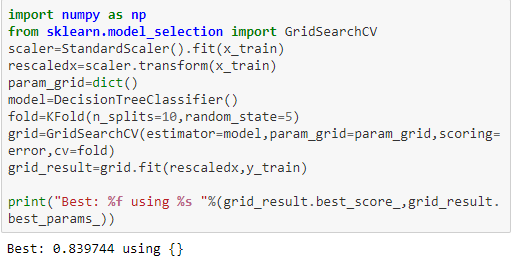
1. **Regularisation Tuning For Top 2 Classification Algorithms.**

As per above accuracy we chose top 2 best performance algorithms:

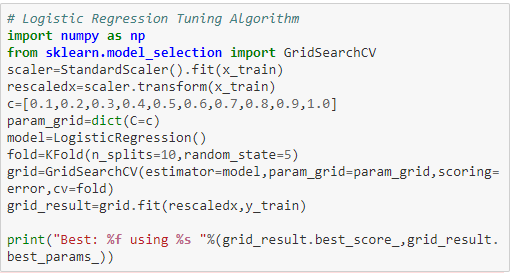
1. Decision Tree Classification Algorithm
2. Logistic Regression Classification Algorithm

After Applying Tuning to top 2 algorithms:

1. Decision Tree Classification Algorithm Best: 0.852500 using {}

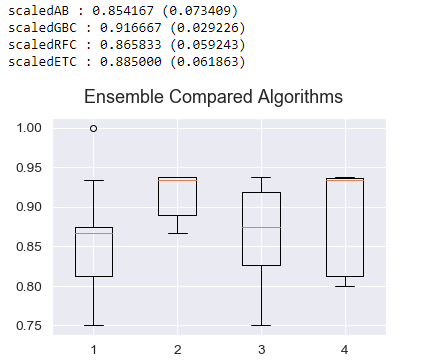


1. Logistic Regression Classification Algorithm Best: 0.853333 using {'C': 0.1}





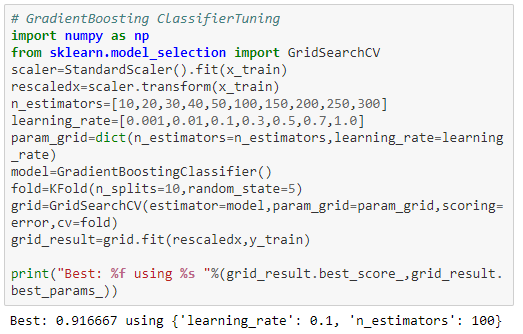
1. **Ensemble and Boosting Classification Algorithms with Feature Scale.**
2. Ada Boost Classification Algorithm
3. Gradient Boosting Classification Algorithm
4. Random Forest Classification Algorithm
5. Extra Trees Classification Algorithm



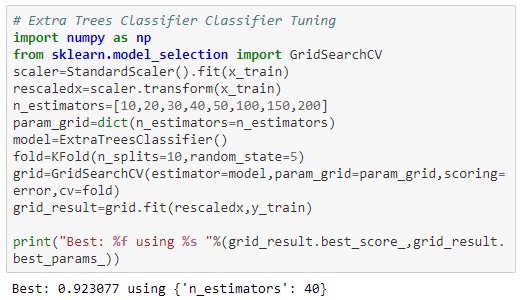
1. **Regularisation Tuning For Top 2 Ensemble and Boosting Classification Algorithms.**

Top 2 ensemble algorithms:

1. Gradient Boosting Classification Algorithm

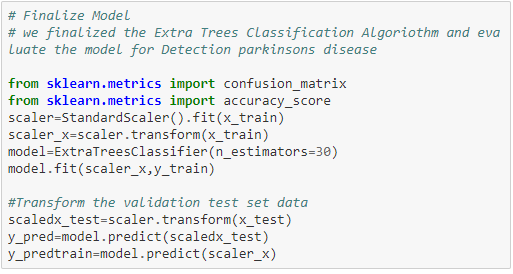


1. Extra Trees Classification Algorithm



1. **Compare All 4 Tuned Algorithms and Selecting the Best Algorithm comparing all 4 algorithms top 2 algorithm and top 2 ensemble algorithms.**
2. Decision Tree Classification Algorithm Best: 0.852500 using {}
3. Logistic Regression Classification Algorithm Best: 0.853333 using {'C': 0.1}
4. Gradient Boosting Classification Algorithm 0.916667 using {'learning\_rate': 0.1, 'n\_estimators': 100}
5. Extra Trees Classification Algoriothm 0.917083 using {'n\_estimators': 30}

Extra Trees Classification Algoriothm 0.917083 using {'n\_estimators': 30} gave the best accuracy performance so used this ensemble algorithm to fit and predict our dataset.

1. **Fit and Predict The Best Algorithm.**

1. **Accuracy of an Algorithm.**

