

# **PARKINSON'S DISEASE PREDICATION USING MACHINE LEARNING TECHNIQUE**

Submitted in partial fulfillment of the requirements for the award of  
Bachelor of Engineering degree in Computer Science and Engineering

by

**VIVEK KUMAR (Reg. No - 40111447)**  
**ABHISHEK KUMAR (Reg. No - 40110015)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**SCHOOL OF COMPUTING**

## **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)**

**Category-1 University by UGC |**

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,  
CHENNAI - 600119**

**APRIL - 2024**



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

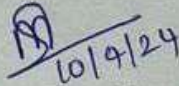
Accredited "A++" by NAAC | Approved by AICTE

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **VIVEK KUMAR** (Reg. No - 40111447) and **ABHISHEK KUMAR** (Reg. NO - 40110015) who carried out the Project entitled "**PARKINSON'S DISEASE PREDICTION USING MACHINE LEARNING TECHNIQUE**" under my supervision from November 2023 to April 2024.

  
10/9/24

Internal Guide

Dr. SANKARI M, M.E., Ph.D.

HEAD OF THE DEPARTMENT  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
**SATHYABAMA**  
Head of the Department  
INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)  
Dr. L. LAKSHMANAN, M.E., Ph.D.  
Jeppiaar Nagar, Rajiv Gandhi Salil,  
Chennai - 600 119.

Submitted for Viva voce Examination held on

13/04/2024

Internal Examiner

External Examiner

## DECLARATION

I, **ABHISHEK KUMAR** (Reg. No - 40110015) hereby declare that the Project Report entitled "**PARKINSON'S DISEASE PREDICTION USING MACHINE LEARNING TECHNIQUE**" done by me under the guidance of **Dr. SANKARI M, M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 13/04/2024

PLACE: Chennai

*Abhishek Kumar*

SIGNATURE OF CANDIDATE

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management** of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. L. Lakshmanan, M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. SANKARI M, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

## **ABSTRACT**

In various data repositories, there are large medical datasets available that are used to identify the diseases. Parkinson's is considered one of the deadliest and progressive nervous system diseases that affect movement. It is the second most common neurological disorder that causes disability, reduces the life span, and still has no cure. Nearly 90% of affected people with this disease have speech disorders. In real-world applications, the information is generated by using various Machine Learning techniques. Machine learning algorithms help to generate useful content from it. To increase the lifespan of elderly people the machine learning algorithms are used to detect diseases in the early stages. Speech features are the main concept while taking into consideration the term 'Parkinson's'. In this model, we are using various Machine Learning techniques like KNN, Linear Regression, SVM, Decision Tree, Random Forest XGBoost and Logistic Regression and how these algorithms are used to predict Parkinson's based on the input taken from the user and the input for algorithms is the dataset. Based on these features the author predicts the algorithm that gives more accuracy. The accuracies obtained for the seven algorithms are KNN with 96%, highest accuracy and it is used in the frontend to predict whether the patient has Parkinson's is present or not. To recover the patients from early stages, prediction is important. This process can be done with the help of Machine Learning.

## TABLE OF CONTENTS

| CHAPTER<br>NO. | TITLE                                     | PAGE<br>NO. |
|----------------|-------------------------------------------|-------------|
|                | <b>ABSTRACT</b>                           | v           |
|                | <b>LIST OF FIGURES</b>                    | vii         |
|                | <b>LIST OF ABBREVIATIONS</b>              | ix          |
| <b>1</b>       | <b>INTRODUCTION</b>                       | 1           |
|                | 1.1 Introduction to Parkinson's Disease   | 1           |
|                | 1.2 Parkinson's Disease Symptoms          | 4           |
|                | 1.3 Treatment of Parkinson's Disease      | 6           |
|                | 1.4 Motivation of the work                | 7           |
|                | 1.5 Problem Statement                     | 7           |
| <b>2</b>       | <b>LITERATURE SURVEY</b>                  | 8           |
|                | 2.1 Related Work                          | 8           |
| <b>3</b>       | <b>REQUIREMENT ANALYSIS</b>               | 12          |
|                | 3.1 Feasibility Studies of the Project    | 12          |
|                | 3.1.1 Economic Feasibility                | 12          |
|                | 3.1.2 Technical Feasibility               | 13          |
|                | 3.1.3 Operational Feasibility             | 14          |
|                | 3.2 Requirements Specification            | 14          |
|                | 3.2.1 System Requirement                  | 14          |
|                | 3.2.2 Functional Requirement              | 15          |
|                | 3.2.3 Non - Functional Requirement        | 15          |
|                | 3.2.4 Software Requirement                | 16          |
|                | 3.2.5 Hardware Requirement                | 16          |
|                | 3.2.6 Python                              | 16          |
|                | 3.2.7 Flask Framework                     | 17          |
|                | 3.2.8 NumPy                               | 18          |
|                | 3.2.9 Pandas                              | 18          |
|                | 3.2.10 Pickle                             | 18          |
|                | 3.2.11 Heatmap                            | 19          |
| <b>4</b>       | <b>DESCRIPTION OF THE PROPOSED SYSTEM</b> | 20          |
|                | 4.1 System Architecture                   | 20          |
|                | 4.2 Module Divisions                      | 22          |

|          |                                      |           |
|----------|--------------------------------------|-----------|
|          | 4.2.1 Speech Dataset                 | 22        |
|          | 4.2.2 Data Pre- Processing           | 24        |
|          | 4.2.3 Feature Extraction             | 25        |
|          | 4.2.3 Feature Selection              | 26        |
|          | 4.2.5 Training Data.                 | 27        |
| <b>5</b> | <b>IMPLEMENTATION DETAILS</b>        | <b>29</b> |
|          | 5.1 Development and Deployment Setup | 29        |
|          | 5.2 Algorithms                       | 30        |
|          | 5.2.1 KNN                            | 30        |
|          | 5.2.2 Linear Regression              | 30        |
|          | 5.2.3 Logistic Regression            | 31        |
|          | 5.2.4 Decision Tree                  | 31        |
|          | 5.2.5 Support Vector Machine         | 32        |
|          | 5.2.6 Random Forest                  | 32        |
|          | 5.2.7 XGBoost                        | 33        |
|          | 5.3 Testing                          | 33        |
| <b>6</b> | <b>RESULTS AND DISCUSSIONS</b>       | <b>35</b> |
|          | 6.1 Analysis of the Results          | 36        |
| <b>7</b> | <b>CONCLUSION</b>                    | <b>40</b> |
|          | <b>REFERENCES</b>                    | <b>41</b> |
|          | <b>APPENDIX</b>                      | <b>43</b> |
|          | <b>A. SOURCE CODE</b>                | <b>46</b> |
|          | <b>B. SCREENSHOTS</b>                | <b>56</b> |
|          | <b>C. RESEARCH PAPER</b>             | <b>58</b> |

## LIST OF FIGURES

| FIGURE NO. | FIGURE NAME                                   | PAGE NO. |
|------------|-----------------------------------------------|----------|
| 1.1        | Structure of Neuron                           | 2        |
| 1.2        | Parkinson's Disease Symptoms                  | 5        |
| 4.1        | System Architecture                           | 21       |
| 4.2        | Speech Dataset from Kaggle                    | 22       |
| 4.3        | Reading the Dataset from CSV file             | 23       |
| 4.4        | Gender Count                                  | 24       |
| 4.5        | Correlation Matrix                            | 25       |
| 4.6        | Dropping Unnecessary feature from data frame  | 26       |
| 4.7        | Process of feature Selection and Sample data  | 27       |
| 4.8        | Splitting Dataset into training and test data | 27       |
| 6.1        | Relationship between data values on heat map  | 35       |
| 6.2        | Accuracy Table                                | 36       |
| 6.3        | Accuracy graph of KNN                         | 37       |
| 6.4        | Accuracy graph of SVM                         | 37       |
| 6.5        | User have to give input                       | 38       |
| 6.6        | Parkinson's Disease Detected                  | 38       |
| 6.7        | Parkinson's Disease Not Detected              | 39       |
| 6.8        | Attribute Information                         | 39       |

## LIST OF ABBREVIATIONS

### ABBREVIATION

### EXPANSION

|      |                                       |
|------|---------------------------------------|
| AUC  | Area under curve                      |
| CSS  | Cascading Style Sheets                |
| DNA  | Deoxyribonucleic acid                 |
| HTML | Hyper Text Markup Language            |
| IDE  | International Development Association |
| KNN  | K – nearest neighbor                  |
| LR   | Linear Regression                     |
| ML   | Machine Learning                      |
| PD   | Parkinson's Disease                   |
| SVM  | Support Vector Machine                |
| URL  | Uniform Resource Locator              |

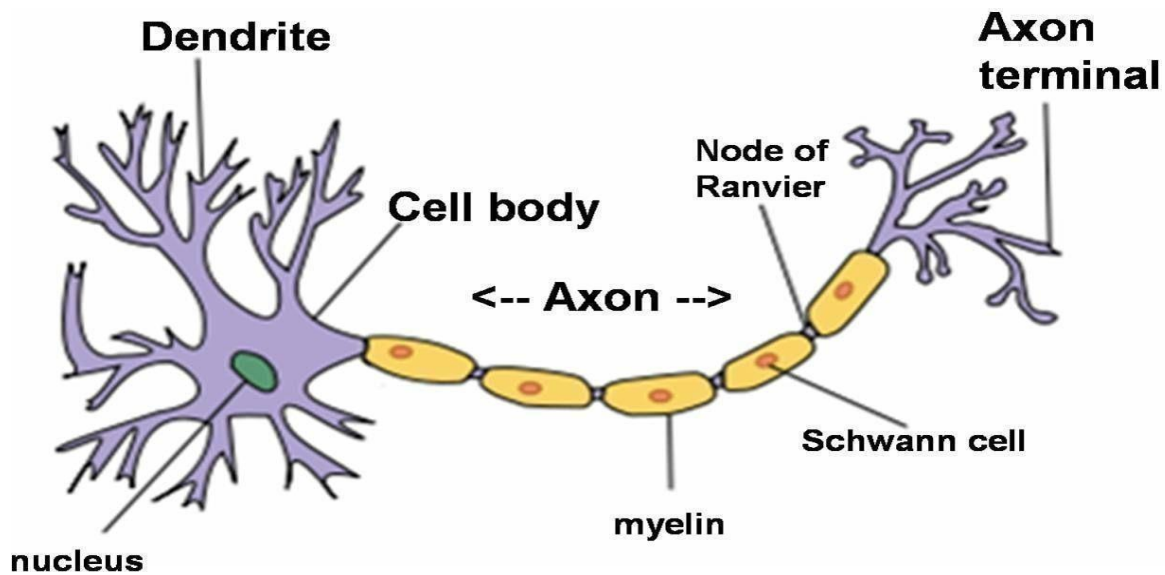
# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 INTRODUCTION TO PARKINSON'S DISEASE**

The recent report of the World Health Organization shows a visible increase in the number and health burden of Parkinson's disease patients increases rapidly. In China, this disease is spreading so fast and estimated that it reaches half of the population in the next 10 years. Classification algorithms are mainly used in the medical field for classifying data into different categories according to the number of characteristics. Parkinson's disease is the second most dangerous neurological disorder that can lead to shaking, shivering, stiffness, and difficulty walking and balance. It caused mainly due by the breaking down of cells in the nervous system. Parkinson's can have both motor and non-motor symptoms. The motor symptoms include slowness of movement, rigidity, balance problems, and tremors. If this disease continues, the patients may have difficulty walking and talking. The non-motor symptoms include anxiety, breathing problems, depression, loss of smell, and change in speech. If the above-mentioned symptoms are present in the person, then the details are stored in the records. In this paper, the author considers the speech features of the patient, and this data is used for predicting whether the patient has Parkinson's disease or not.

Neurodegenerative disorders are the results of progressive tearing and neuron loss in different areas of the nervous system. Neurons are functional units of the brain. They are contiguous rather than continuous. A good healthy-looking neuron as shown in fig 1 has extensions called dendrites or axons, a cell body, and a nucleus that contains our DNA. DNA is our genome and a hundred billion neurons contain our entire genome which is packaged into it. When a neuron gets sick, it loses its extension and hence its ability to communicate which is not good for it and its metabolism becomes low so it starts to accumulate junk and it tries to contain the junk in the little packages in little pockets. When things become worse and if the neuron is a cell culture it completely loses its extension, becomes round and full of vacuoles.



**Fig 1.1 Structure of Neuron**

This work deals with the prediction of Parkinson's disorder which is now a day is tremendously increasing incurable disease. Parkinson's disease is a most spreading disease which gets its name from James Parkinson who earlier described it as a paralysis agitans and later gave his surname was known as PD. It generally affects the neurons which are responsible for overall body movements. The main chemicals are dopamine and acetylcholine which affect the human brain. There is a various environmental factor which has been implicated in PD below are the listed factor which caused Parkinson's disease in an individual.

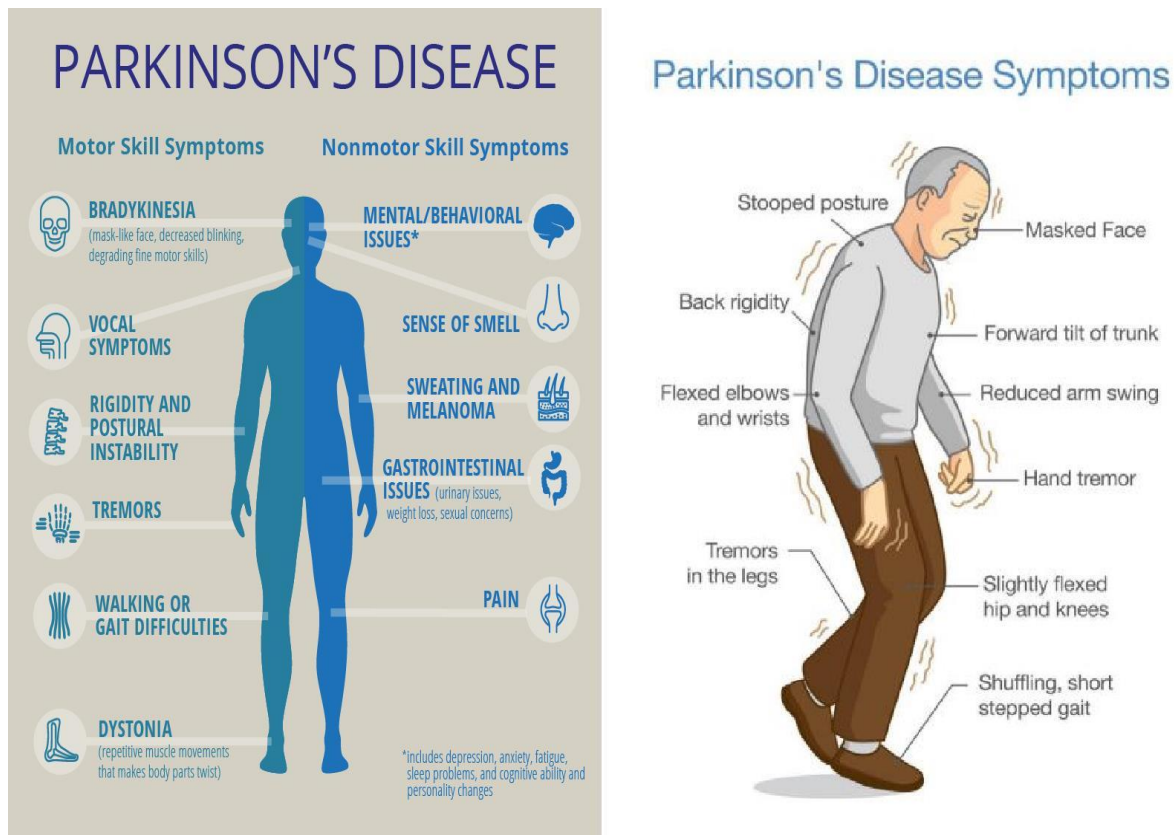
- **Environmental factors:** Environment is defined as the surroundings or the place in which an individual lives. So the environment is the major factor that will not only affects the human's brain but also affects all the living organism who lives in the vicinity of it. Many types of research and evidence have proved that the environment has a big hand in the development of neurodegenerative disorders mainly Alzheimer's and Parkinson's. There are certain environmental factors that are influencing neurodegenerative disorder with high pace are: -
  - Exposure to heavy metals (like lead and aluminum) and pesticides.
  - Air Quality: Pollution results in respiratory diseases.

- Water quality: Biotic and Abiotic contaminants present in water lead to water pollution.
- Unhealthy lifestyle: It leads to obesity and a sedentary lifestyle.
- Psychological stress: It increases the level of stress hormone that depletes the functions of neurons.
- **Brain injuries or Biochemical Factors:** The brain is the control center of our complete body. Due to certain trauma, people have brain injuries which leads some biochemical enzymes to come into the picture which provides neurons stability and provides support to some chromosomes and genes in maintenance.
- **Aging Factor:** Aging is one of the reasons for the development of Parkinson's disease. According to the author in India, 11,747,102 people out of 1, 065, 070, 6072 are affected by Parkinson's disease.
- **Genetic factors:** Genetic factor is considered as the main molecular physiological cause which leads to neurodegenerative disorders. The size, depth, and effect of actions of different genes define the status or level of neurodegenerative disease which increases itself gradually over time. Mainly the genetic factors which lead to Neurodegenerative disorders are categorized into pharmacodynamics and pharmacokinetics.
- **Speech Articulation factors:** Due to the condition associated with Parkinson's disease (rigidity and bradykinesia), some speech-language pathology such as voice, articulation and swallowing alterations are found. There are various ways in which Parkinson's disease (PD) might affect the individual.
  - The voice get breathy and softer.
  - Speech may be smeared.
  - The person finds difficulty in finding the right words due to which speech becomes slower.

## 1.2 PARKINSON'S DISEASE SYMPTOMS

The symptoms of Parkinson's disease broadly divided into two categories.

- **Motor symptoms:** This is a symptom where any voluntary action involved. It indicates the movement-related disorders like tremors, rigidity, freezing, Bradykinesia or any voluntary muscle movement.
- **Non-Motor symptoms:** Non motor symptoms include disorders of mood and affect with apathy, cognitive dysfunction as well as complex behavioral disorders. There are two other categories of PD which are divided by doctors: Primary symptom and Secondary symptom.
- **Primary symptoms:** It is the most important symptom. Primary symptoms are rigidity, tremor and slowness of movement.
- **Secondary symptoms:** It is a symptom that directly impacts the life of an individual. These can be either motor or non-motor. Its effect depends on person to person. A very wide range of symptoms is associated with Parkinson's. Besides these symptoms, there are some other symptoms found that lead to Parkinson's disease. These symptoms are micrographic, decreased olfaction & postural instability, slowing of the digestive system, constipation, fatigue, weakness, and Hypotension. Speech difficulties i.e. dysphonia (impaired speech production) and dysarthria (speech articulation difficulties) are found in patients with Parkinson's.



**Fig 1.2 Parkinson's Disease Symptoms**

Before the cardinal motor aspects of PD show, it is estimated that up to 80% of dopaminergic cells in the nigro-striatal pathway are destroyed. (Sveinbjornsdottir, 2016) The earliest motor signs are usually used to diagnose the condition. The diagnosis is made using criteria established by the UK PD Brain Bank. (Sveinbjornsdottir, 2016) Slow voluntary movement start with increasing reduction in speed and amplitude of repeating acts (bradykinesia) plus one additional symptom, such as muscle rigidity, resting tremor, or postural instability, is required for diagnosis. (Sveinbjornsdottir, 2016) Step 2 in the diagnosis is to rule out symptoms that could indicate other aetiologies, such as parkinsonian syndromes, which have their own neuropathological changes, and Step 3 is to establish at least three supportive criteria for Parkinson's disease, such as unilateral onset of symptoms, persistent asymmetry of clinical symptoms, good response to levodopa treatment, and induction of dyskinesias by dopaminergic treatment. (Sveinbjornsdottir, 2016) In most cases, symptoms begin on one side of the body and progress to the opposite side within a

few years. (Sveinbjornsdottir, 2016) When walking, the body posture becomes stooped, there is axial and limb rigidity with or without cogwheel phenomena, a shuffling gait, and a lack of arm movement. (Sveinbjornsdottir, 2016) The bradykinesia can cause an expressionless face (hypomimia) and reduced handwriting amplitudes (micrographia). (Sveinbjornsdottir, 2016) Around 80% of people have limb tremor, which is most typically described as a resting pill-rolling tremor of the hands. (Sveinbjornsdottir, 2016) The tendency of the thumb and index finger to come into touch and move in a circular motion is known as pill rolling. (Sveinbjornsdottir, 2016) The tremor can sometimes affect the legs, and other tremor kinds can develop. (Sveinbjornsdottir, 2016) Other gait disorders besides shuffling include blocking, hesitating, and gait festination, in which steps get smaller and faster over time, potentially leading to loss of balance and falls. (Sveinbjornsdottir, 2016) After several years of beginning, a quarter to 60 percent of individuals report freezing of movements. (Sveinbjornsdottir, 2016)

### **1.3 TREATMENT OF PARKINSON DISEASE**

The cause of Parkinson's disease is likely complex, and there is currently no treatment that can slow or stop the disease's course. (Sveinbjornsdottir, 2016) The use of dopaminergic medications is symptomatic, with the goal of resolving the motor abnormalities. (Sveinbjornsdottir, 2016) Standard and most usual initial medication for patients is levodopa, a prodrug to dopamine. (Sveinbjornsdottir, 2016) The initial response is usually positive. The majority of patients experience shorter duration of response to individual doses (wearing-off symptoms), alternate phases of good and poor response to medication (on-off symptoms), involuntary movements of the head, trunk, or limbs (dyskinesias), and other motor complications as the disease progresses and the system's capacity to store dopamine decreases. To control these swings, other dopaminergic drugs are employed. (Sveinbjornsdottir, 2016) Monoamine oxidase type B inhibitors, catechol-O-methyltransferase inhibitors, amantadine, an NMDA receptor antagonist, and dopamine receptor agonists are among them. (Sveinbjornsdottir, 2016) When medicinal therapy fails to manage the motor symptoms, surgical therapy,

usually with deep brain electrical stimulation, is possible for a small percentage of patients. (Sveinbjornsdottir, 2016)

## **1.4 MOTIVATION OF THE WORK**

Many of the people aged 65 or more do have a neurodegenerative disease, which has no cure. If we detect the disease in the early stages, then we can control it. Almost 30% of the patients are facing this incurable disease. Current treatment is available for patients who have minor symptoms. If these symptoms cannot be found at the early stages, it leads to death. The main cause for Parkinson's disease is the accumulation of protein molecules in the neuron which gets misfolded and hence causing Parkinson's disease. So till now, researchers got the symptoms and the root causes i.e. from where this disease had evolved. But very few symptoms have come to their cure and there are many symptoms that have no solution. So in this era where Parkinson's disease is increasing, it is very important to find the solution which can predict it in its early stages.

## **1.5 PROBLEM STATEMENT**

The main aim is to predict the prediction efficiency that would be beneficial for the patients who are suffering from Parkinson and the percentage of the disease will be reduced. Generally, in the first stage, Parkinson's can be cured by the proper treatment. So it's important to identify the PD at the early stage for the betterment of the patients. The main purpose of this research work is to find the best prediction model i.e. the best machine learning technique which will distinguish the Parkinson's patient from the healthy person. The techniques used in this problem are KNN, Naïve Bayes, and Logistic Regression. The experimental study is performed on the voice dataset of Parkinson's patients which is downloaded from the Kaggle. The prediction is evaluated using evaluation metrics like confusion matrix, precision, recall accuracy, and f1-score. The author used feature selection where the important features are taken into consideration to detect Parkinson's.

## **CHAPTER 2**

### **LITERATURE SURVEY**

#### **2.1 RELATED WORK**

Speech or voice data is assumed to be 90% helpful to diagnose a person for identifying the presence of disease. It is one of the most important problems that have to be detected in the early stages so that the progression rate of the disease is reduced. Many of the researchers work on different datasets to predict the disease more efficiently. In general, Persons with PD suffer from speech problems, which can be categorized into two: hypophonia and dysarthria. Hypophonia indicates a very soft and weak voice from a person and dysarthria indicates slow speech or voice, that can hardly be understood at one time and this causes damage to the central nervous system. So, most of the clinicians who treat PD patients observe dysarthria and check out to rehabilitate with specific treatments to improvise vocal intensity. Lots of researchers did work on the pre-processing data and feature selection in the past.

Anila M and Dr G Pradeepini proposed the paper titled “Diagnosis of Parkinson’s disease using Artificial Neural network” [2]. The main objective of this paper is that the detection of the disease is performed by using the voice analysis of the people affected with Parkinson's disease. For this purpose, various machine learning techniques like ANN, Random Forest, KNN, SVM, XG Boost are used to classify the best model, error rates are calculated, and the performance metrics are evaluated for all the models used. The main drawback of this paper is that it is limited to ANN with only two hidden layers. And this type of neural networks with two hidden layers are sufficient and efficient for simple datasets. They used only one technique for feature selection which reduces the number of features.

Arvind Kumar Tiwari Proposed the paper titled “Machine Learning-based Approaches for Prediction of Parkinson's Disease” [3]. In this paper, minimum redundancy maximum relevance feature selection algorithms were used to select the most important feature among all the features to predict Parkinson diseases. Here, it was observed

that the random forest with 20 number of features selected by minimum redundancy maximum relevance feature selection algorithms provide the overall accuracy 90.3%, precision 90.2%, Mathew's correlation coefficient values of 0.73 and ROC values 0.96 which is better in comparison to all other machine learning based approaches such as bagging, boosting, random forest, rotation forest, random subspace, support vector machine, multilayer perceptron, and decision tree based methods.

Mohamad Alissa Proposed the paper titled "Parkinson's Disease Diagnosis Using Deep Learning" [14]. This project mainly aims to automate the PD diagnosis process using deep learning, Recursive Neural Networks (RNN) and Convolutional Neural Networks (CNN), to differentiate between healthy and PD patients. Besides that, since different datasets may capture different aspects of this disease, this project aims to explore which PD test is more effective in the discrimination process by analysing different imaging and movement datasets (notably cube and spiral pentagon datasets). In general, the main aim of this paper is to automate the PD diagnosis process in order to discover this disease as early as possible. If we discover this disease earlier, then the treatments are more likely to improve the quality of life of the patients and their families. There are some limitations to this paper namely:

- They used the validation set only to investigate the model performance during the training and this reduced the number of samples in the training set.
- RNN training is too slow and this is not flexible in practice work.
- Disconnecting and resource exhaustion: working with cloud services like Google Collaboratory causes many problems like disconnecting suddenly. And because it is shareable service by the world zones, this leads to resource exhaustion error many times.

Afzal Hussain Shahid and Maheshwari Prasad Singh proposed the paper titled "A deep learning approach for prediction of Parkinson's disease progression" [19]. This paper proposed a deep neural network (DNN) model using the reduced input feature space of Parkinson's telemonitoring dataset to predict Parkinson's disease (PD) progression and also proposed a PCA based DNN model for the prediction of Motor-UPDRS and Total-UPDRS in Parkinson's Disease progression. The DNN model was evaluated on

a real-world PD dataset taken from UCI. Being a DNN model, the performance of the proposed model may improve with the addition of more data points in the datasets.

T. J. Wroge, Y. Özkanca, C. Demiroglu, D. Si, D. C. Atkins and R. H. Ghomi, proposed the paper titled “Parkinson’s Disease Diagnosis Using Machine Learning and Voice” [24] is that it explores the effectiveness of using supervised classification algorithms, such as deep neural networks, to accurately diagnose individuals with the disease. Historically, PD has been difficult to quantify and doctors have tended to focus on some symptoms while ignoring others, relying primarily on subjective rating scales. The analysis of this paper provides a comparison of the effectiveness of various machine learning classifiers in disease diagnosis with noisy and high dimensional data. Their peak accuracy of 85% provided by the machine learning models exceeds the average clinical diagnosis accuracy of non-experts (73.8%) and average accuracy of movement disorder specialists (79.6% without follow-up, 83.9% after follow-up) with pathological post-mortem examination as ground truth.

T. J. Wroge, Y. Özkanca, C. Demiroglu, D. Si, D. C. Atkins and R. H. Ghomi, proposed the paper titled “Parkinson’s Disease Diagnosis Using Machine Learning and Voice” [24] is that it explores the effectiveness of using supervised classification algorithms, such as deep neural networks, to accurately diagnose individuals with the disease. Historically, PD has been difficult to quantify and doctors have tended to focus on some symptoms while ignoring others, relying primarily on subjective rating scales. The analysis of this paper provides a comparison of the effectiveness of various machine learning classifiers in disease diagnosis with noisy and high dimensional data. Their peak accuracy of 85% provided by the machine learning models exceeds the average clinical diagnosis accuracy of non-experts (73.8%) and average accuracy of movement disorder specialists (79.6% without follow-up, 83.9% after follow-up) with pathological post-mortem examination as ground truth.

Dr. R. Geetha Ramani, G. Sivagami, and Shomona Gracia Jacob proposed a paper “Feature Relevance Analysis and Classification of Parkinson’s Disease TeleMonitoring

data Through Data Mining” [6]. In this paper, the author used thirteen classification algorithms to diagnose the disease. The author used the Tele-monitoring dataset which contains 16 biomedical voice features for evaluating the system. The aim of this paper is to predict motor UPDRS and total UPDRS from the voice measures.

A. Ozcift, proposed a paper “SVM feature selection based rotation forest ensemble classifiers to improve computer-aided diagnosis of Parkinson disease” [1]. In this paper, the author summarizes that improve the PD diagnosis accuracy with the use of support vector machine feature selection. To evaluate the performances the author used accuracy, kappa statistics, and area under the curve of the classification algorithms. The rotation Forest ensemble of these classifiers used to increase the performance of the system.

T. Swapna, Y. Sravani Devi proposed a paper and titled “Performance Analysis of Classification algorithms on Parkinson’s Dataset with Voice Attributes” [23]. This paper deals with the application of seven classification algorithms on the acquired data set and then drawing out a comparison of the results to one another and also predicting the outcome whether the person is healthy or Parkinson disease effected from the given data. The results of the selected algorithms namely Naïve Bayes, Random Forest, Neural Networks, Decision Trees, AdaBoost, SVM, KNN were compared and tabulated. According to the outputs derived with the help of python, implementing Scikit Libraries. Final accuracy was calculated using these parameters. Random Forest algorithm gives with optimum accuracy of 78.56% which is closely followed by DecisionTree Algorithm with the optimal accuracy of 77.63%. Following the Decision Tree Algorithm is the MLP Classifier with an optimal accuracy of 76.72%, and lastly the Naïve Bayes Algorithm which has the optimal accuracy of 70.82%. Finally, these algorithms can help in classifying whether a person get effected with Parkinson’s disease or not.

## **CHAPTER 3**

### **REQUIRMENT ANALYSIS**

#### **3.1 FEASIBILITY STUDIES OF THE PROJECT**

The preliminary investigation examines project feasibility, the likelihood the system are going to be useful to the organization. The main objective of the feasibility studyis to check the Technical, Operational, and Economical feasibility for adding new modules and debugging old running systems. All systems are possible if they have unlimited resources and infinite time to do a task. There are aspects within the feasibility study portion of the preliminary investigation:

- Economical Feasibility
- Technical Feasibility
- Operational Feasibility

##### **3.1.1 ECONOMIC FEASIBILITY**

The feasibility study for the Parkinson's disease project aims to evaluate its economic viability as a crucial determinant of its potential success. A comprehensive market analysis reveals promising demand for Parkinson's disease-related products and services, indicating a substantial market opportunity. With a clear understanding of the market dynamics, cost estimation becomes imperative. By meticulously breaking down the project's costs—including research and development, manufacturing, marketing, and distribution expenses—a clear picture of the financial requirements emerges.

Revenue projection is another critical component, which involves estimating potential income streams such as product sales, licensing agreements, or service fees. This estimation process entails considering various scenarios and assumptions regarding market penetration and pricing strategies. The subsequent financial analysis involves calculating key metrics like net present value (NPV), internal rate of return (IRR), payback period, and profitability index. These metrics offer insights into the project's potential profitability and return on investment.

Revenue projection is another critical component, which involves estimating potential income streams such as product sales, licensing agreements, or service fees. This estimation process entails considering various scenarios and assumptions regarding market penetration and pricing strategies. The subsequent financial analysis involves calculating key metrics like net present value (NPV), internal rate of return (IRR), payback period, and profitability index. These metrics offer insights into the project's potential profitability and return on investment.

### **3.1.2 TECHNICAL FEASIBILITY**

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves evaluation of the hardware, software, and other technology requirements of the proposed system. This assessment is predicated on an overview design of system requirements, to work out whether the corporate has the technical expertise to handle completion of the project. When writing a feasibility report, the subsequent should be taken to consideration:

- A brief description of the business to assess more possible factors which could affect the study
- The part of the business being examined
- The human and economic factor
- The possible solutions to the problem At this level, the concern is whether the proposal is both technically and legally feasible (assuming moderate cost). The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

### **3.1.3 OPERATIONAL FEASIBILITY**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as a crucial part of the project implementation. Some of the important issues raised are to check the operational feasibility of a project includes the following:

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits? This system is targeted to be in accordance with the above- mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

## **3.2 REQUIREMENT SPECIFICATION**

### **3.2.1 SYSTEM REQUIREMENTS**

A requirement is a feature that the system must have or a constraint that it must to be accepted by the client. Requirement Engineering aims at defining the wants of the system under construction. Requirement Engineering include two main activities requirement elicitation which results in the specification of the system that the client understands and analysis which in analysis model that the developer can unambiguously interpret. A requirement may be a statement about what the proposed system will do.

Requirements can be divided into two major categories:

- Functional Requirements.
- Non-Functional Requirements.

### **3.2.2 FUNCTIONAL REQUIREMENTS**

A Functional Requirement may be a description of the service that the software must offer. It describes a software system or its component. A function is nothing but input to the software, its behavior, and outputs. It are often a calculation, data manipulation, business process, user interaction, or the other specific functionality which defines what function a system is probably going to perform. Functional Requirements describe the interactions between the system and its environment independent of its application.

- Applying the algorithms on the test data.
- Display the result with the description of having Parkinson's or not.

### **3.2.3 NON-FUNCTIONAL REQUIREMENTS**

Non-Functional Requirements specifies the standard attribute of a software. They judge the software supported Responsiveness, Usability, Security, Portability, and other non-functional standards that are critical to the success of the software.

An example of a nonfunctional requirement, "how fast does the website load?" Failing to satisfy non-functional requirements may result in systems that fail to satisfy user needs.

Non-functional Requirements allow you to impose constraints or restrictions on the planning of the system across the varied agile backlogs.

- Accuracy
- Reliability
- Flexibility

### 3.2.4 SOFTWARE REQUIREMENT

- Operating System - Windows 10 and above
- Server-side Script - Python 3.6+
- IDE - PyCharm Notebook
- Libraries Used - Pandas, NumPy, Sci-Kit  
Learn, Matplotlib, Seaborn.

### 3.2.5 HARDWARE REQUIREMENT

- Processor - I5/Intel Processor and above
- RAM - 4GB (min)
- Hard Disk - 256 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - Any

### 3.2.6 PYTHON

Python is an interpreter, high-level, general-purpose programming language. Python is simple and easy to read syntax emphasizes readability and thus reduces system maintenance costs. Python supports modules and packages, which promote system layout and code reuse. It saves space but it takes a rather higher time when its code is compiled. Indentation must be taken care while coding.

Python does the following:

- Python are often used on a server to make web applications.
- It connects the database systems. It also read and modify files.
- It often able to handle big data and perform complex mathematics.
- It can be used for production-ready software development.

Python has many inbuilt library functions that can be used easily for working with machine learning algorithms. All the necessary python libraries must be pre-installed using “pip” command.

### 3.2.7 FLASK FRAMWORK

Web Application Framework or just Web Framework represents a set of libraries and modules that permits an internet application developer to write down applications without having to bother about low-level details such as protocols, thread management, etc. Flask may be a web application framework written in Python. It is developed by Armin Ronacher, who leads a world group of Python enthusiasts named Pocco. Flask is predicated on the Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects. Python 2.6 or higher is usually required for the installation of Flask. Although Flask and its dependencies work well with Python 3, many Flask extensions do not support it properly. Hence, it's recommended that Flask should be installed on Python 2.7. Importing the flask module in the project is mandatory. An object of the Flask class is our WSGI application.

Flask constructor takes the name of current module (`_name__`) as argument.

The `route()` function of the Flask class may be a decorator, which tells the appliance which URL should call the associated function.

The `rule` parameter represents URL binding with the function.

The `options` is an inventory of parameters to be forwarded to the underlying Rule object.

Finally the `run()` method of Flask class runs the application on the local development server.

Open the URL (`localhost:5000`) in the browser. The message will be displayed on it.

A Flask application is started by calling the `run()` method. However, while the appliance is under development, it should be restarted manually for every change within the code. To avoid this inconvenience, enable debug support. The server will then reload itself if the code changes. It will also provide a useful debugger to trace the errors if any, within the application.

The Debug mode is enabled by setting the debug property of the application object to True before running or passing the debug parameter to the run() method.

### **3.2.8 NUMPY**

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the elemental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy also can be used as an efficient multi-dimensional container of generic data.

### **3.2.9 PANDAS**

Pandas is an open-source library that's built on top of NumPy library. It is a Python package that gives various data structures and operations for manipulating numerical data and time series. It is fast and it has high-performance & productivity for users. It provides high-performance and is easy-to-use data structures and data analysis tools for the Python language. Pandas is employed during a wide range of fields including academic and commercial domains including economics, Statistics, analytics, etc.

### **3.2.10 PICKLE**

Python pickle module is employed for serializing and de-serializing a Python object structure. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the thing in another python script. Pickling is beneficial for applications

Where you would like a point of persistency in your data. Your program's state data are often saved to disk, so you'll continue working on it later on.

### **3.2.11 HEAT MAP**

A heat map is a two-dimensional representation of data in which values are represented by colors. A simple heat map provides an immediate visual summary of information. More elaborate heat maps allow the viewer to understand complex data sets.

There can be many ways to display heat maps, but they all share one thing in common -- they use color to communicate relationships between data values that would be much harder to understand if presented numerically in a spreadsheet.

## **CHAPTER 4**

### **DESCRIPTION OF PROPOSED SYSTEM**

#### **4.1 PROPOSED SYSTEM**

The proposed methodology collects audio data from PPMI [21] and UCI about Parkinson's patients voice modulations. Dataset contains information about jitter, shimmer and MDVP of vowel phonations. Data is preprocessed, analyzed and visualized for a thorough understanding of the attributes. Seven models – Logistic regression, SVM, Random Forest, Linear Regression, Decision Tree, XGboost and K nearest neighbors – are trained on 75% of the data. Models are trained to classify given audio data into PD or healthy, based on variations in frequency. Models are tested on 25% of the data and evaluated based on sensitivity, precision, accuracy, confusion matrix [22] and ROC-AUC score. The generic process implemented. It demonstrates the stages of data ingestion from PPMI database, separation of data into testing and training sets, training of Seven models on data and validation of results using test data.

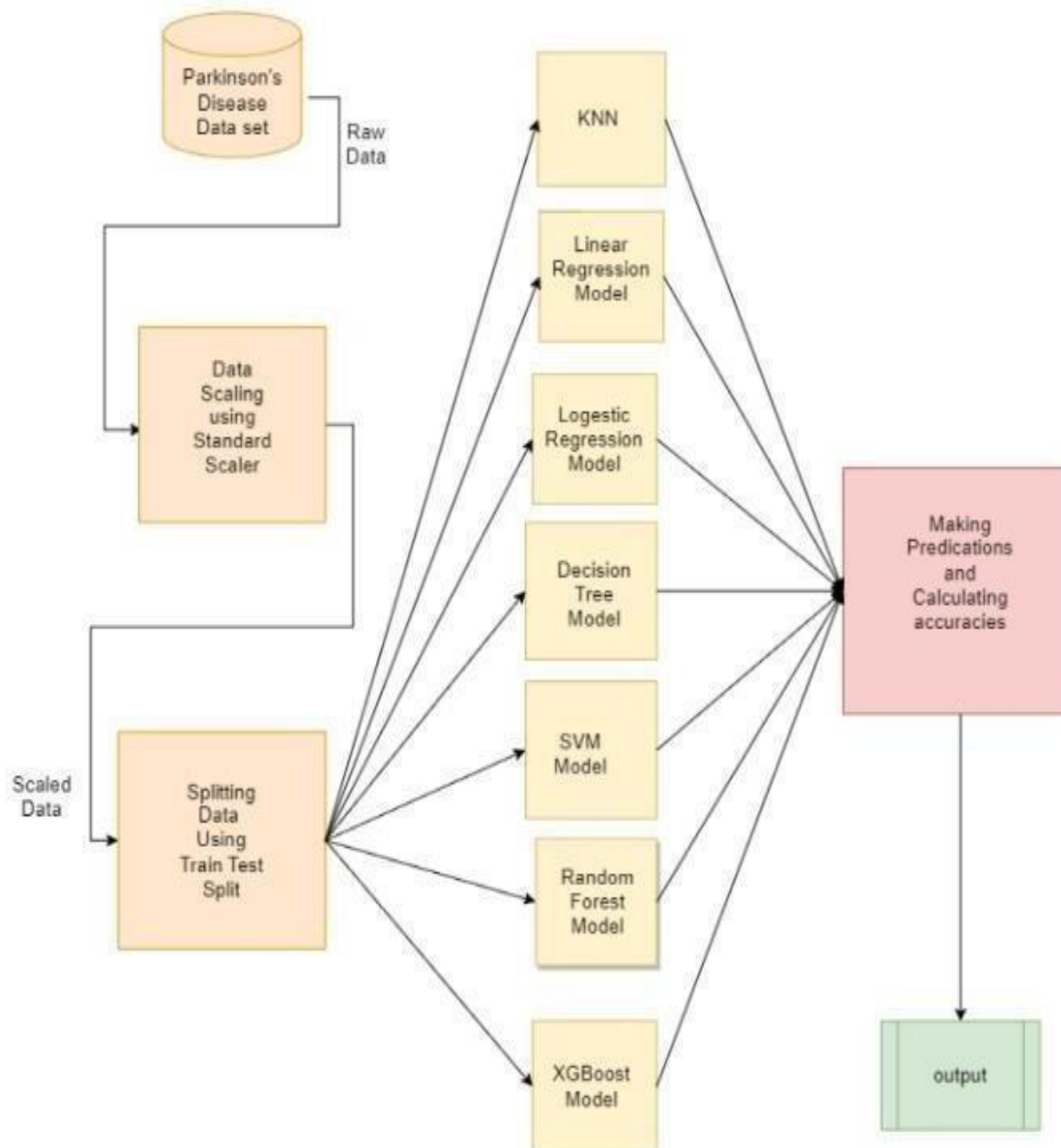
##### **4.1.1 SYSTEM ARCHITECTURE**

Machine learning has given computer systems the ability to automatically learn without being explicitly programmed. In this, the author has used seven machine learning algorithms (Logistic Regression, KNN, Decision Tree, XGBoost, Random Forest Classifier, Linear Regression and SVM). The architecture diagram describes the high-level overview of major system components and important working relationships. It represents the flow of execution and it involves the following five major steps:

- The architecture diagram is defined with the flow of the process which is used to refine the raw data and used for predicting the Parkinson's data.
- The next step is preprocessing the collected raw data into an understandable format.
- Then we have to train the data by splitting the dataset into train data and test data.
- The Parkinson's data is evaluated with the application of a machine

learning algorithm that is Logistic Regression, KNN, Decision Tree, Linear Regression, Random Forest Classifier, XGBoost and SVM algorithm, and the classification accuracy of this model is found.

- After training the data with these algorithms we have to test on the same algorithms.
- Finally, the result of these seven algorithms is compared on the basis of classification accuracy.



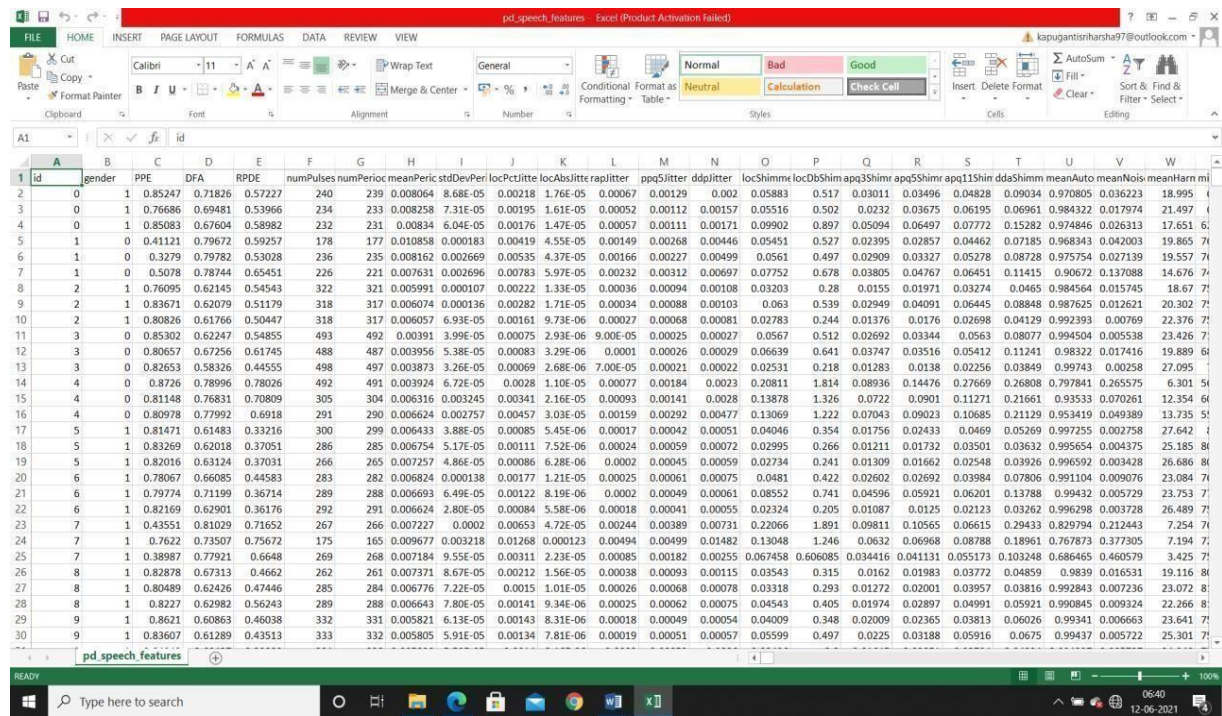
**Fig 4.1 SYSTEM ARICHTECTURE**

## 4.2 MODULES DIVISION

Let us discuss about the various modules in our proposed system and what each module contributes in achieving our goal.

### 4.2.1 SPEECH DATASET

The main aim of this step is to spot and acquire all data-related problems. During this step, we'd like to spot the various data sources, as data are often collected from various sources like files and databases. The number and quality of the collected data will determine the efficiency of the output. The more are going to be the info, the more accurate are going to be the prediction. We've collected our data from the Kaggle website.



The screenshot displays an Excel spreadsheet titled 'pd\_speech\_features'. The spreadsheet contains a table with 30 rows of data. The first two columns are 'id' and 'gender'. The remaining columns represent various voice features, including 'PPE', 'DFA', 'RPDE', 'numPulses', 'numPeriod', 'meanPeric', 'stdDevPeri', 'locPct', 'litte', 'locAbs', 'litte', 'rap', 'litte', 'ppq', 'litte', 'ddp', 'litte', 'locShim', 'locDb', 'Shim', 'apq', '3Shim', 'apq', '5Shim', 'apq', '11Shim', 'dda', 'Shim', 'meanAuto', 'meanNois', and 'meanHarm'. The data is organized into a grid with alternating light blue and white rows.

**Fig 4.2 Speech Dataset From Kaggle**

In the above Fig-3.2, we can see the speech dataset that has collected from kaggle website. This acquired dataset has around 756 patient's data and each row has 755 different voice features.

But in this paper, we chosen 10 main features that required to find the prediction.  
The features are listed below:

- Id
- Gender
- PPE(Pitch Period Entropy)
- DFA(Detrended Fluctuation Analysis)
- RPDE(Recurrent Period Density Entropy)
- NumPulses
- numPeriodPulses
- meanPeriodPulses
- stdDevPeriodPulses
- locPctJitter
- locAbsJitter
- rapJitter
- locShimmer, etc

```
[1] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from google.colab import drive
drive.mount('/content/drive',force_remount = True)
```

Mounted at /content/drive

```
df=pd.read_csv("/content/drive/MyDrive/pd_speech_features.csv")
df.head(10)
```

|   | id | gender | PPE     | DFA     | RPDE    | numPulses | numPeriodsPulses | meanPeriodPulses | stdDevPeriodPulses | locPctJitter | locAbsJitter | rapJitter | ppq5Jitter | ddpJitter | locShimmer | loc |
|---|----|--------|---------|---------|---------|-----------|------------------|------------------|--------------------|--------------|--------------|-----------|------------|-----------|------------|-----|
| 0 | 0  | 1      | 0.85247 | 0.71826 | 0.57227 | 240       | 239              | 0.008084         | 0.000087           | 0.00218      | 0.000018     | 0.00067   | 0.00129    | 0.00200   | 0.05883    |     |
| 1 | 0  | 1      | 0.76686 | 0.69481 | 0.53966 | 234       | 233              | 0.008258         | 0.000073           | 0.00195      | 0.000016     | 0.00052   | 0.00112    | 0.00157   | 0.05516    |     |
| 2 | 0  | 1      | 0.85083 | 0.67604 | 0.58982 | 232       | 231              | 0.008340         | 0.000060           | 0.00176      | 0.000015     | 0.00057   | 0.00111    | 0.00171   | 0.09902    |     |
| 3 | 1  | 0      | 0.41121 | 0.79672 | 0.59257 | 178       | 177              | 0.010858         | 0.000183           | 0.00419      | 0.000046     | 0.00149   | 0.00268    | 0.00446   | 0.05451    |     |
| 4 | 1  | 0      | 0.32790 | 0.79782 | 0.53028 | 236       | 235              | 0.008162         | 0.002669           | 0.00535      | 0.000044     | 0.00166   | 0.00227    | 0.00499   | 0.05610    |     |
| 5 | 1  | 0      | 0.50780 | 0.78744 | 0.65451 | 226       | 221              | 0.007631         | 0.002696           | 0.00783      | 0.000060     | 0.00232   | 0.00312    | 0.00697   | 0.07752    |     |
| 6 | 2  | 1      | 0.76095 | 0.62145 | 0.54543 | 322       | 321              | 0.005991         | 0.000107           | 0.00222      | 0.000013     | 0.00036   | 0.00094    | 0.00108   | 0.03203    |     |
| 7 | 2  | 1      | 0.83671 | 0.62079 | 0.51179 | 318       | 317              | 0.006074         | 0.000136           | 0.00282      | 0.000017     | 0.00034   | 0.00088    | 0.00103   | 0.06300    |     |
| 8 | 2  | 1      | 0.80826 | 0.61766 | 0.50447 | 318       | 317              | 0.006057         | 0.000069           | 0.00161      | 0.000010     | 0.00027   | 0.00068    | 0.00081   | 0.02783    |     |
| 9 | 3  | 0      | 0.85302 | 0.62247 | 0.51855 | 193       | 192              | 0.003010         | 0.000040           | 0.00075      | 0.000003     | 0.00009   | 0.00025    | 0.00027   | 0.05670    |     |

**Fig 4.3 Reading the dataset from the CSV file into notebook**

The dataset we chose is in the form of CSV (Comma Separated Value) file. After acquiring the data our next step is to read the data from the CSV file into the Google colab also called a Python notebook. Python notebook is used in our project for data pre-processing, features selection, and for model comparison. In the fig-3.3, we have shown how to read data from CSV files using the inbuilt python functions that are part of the pandas library.

When compared in genders the Parkinson's disease is mostly found in the male rather than female. As this dataset consists of more male persons we chose. In Fig-3.4, we have shown the male people are more than female.

```
man=df.gender.sum()
total=df.gender.count()
woman=total-man
print("man: "+str(man)+" woman: "+str(woman))

man: 390 woman: 366
```

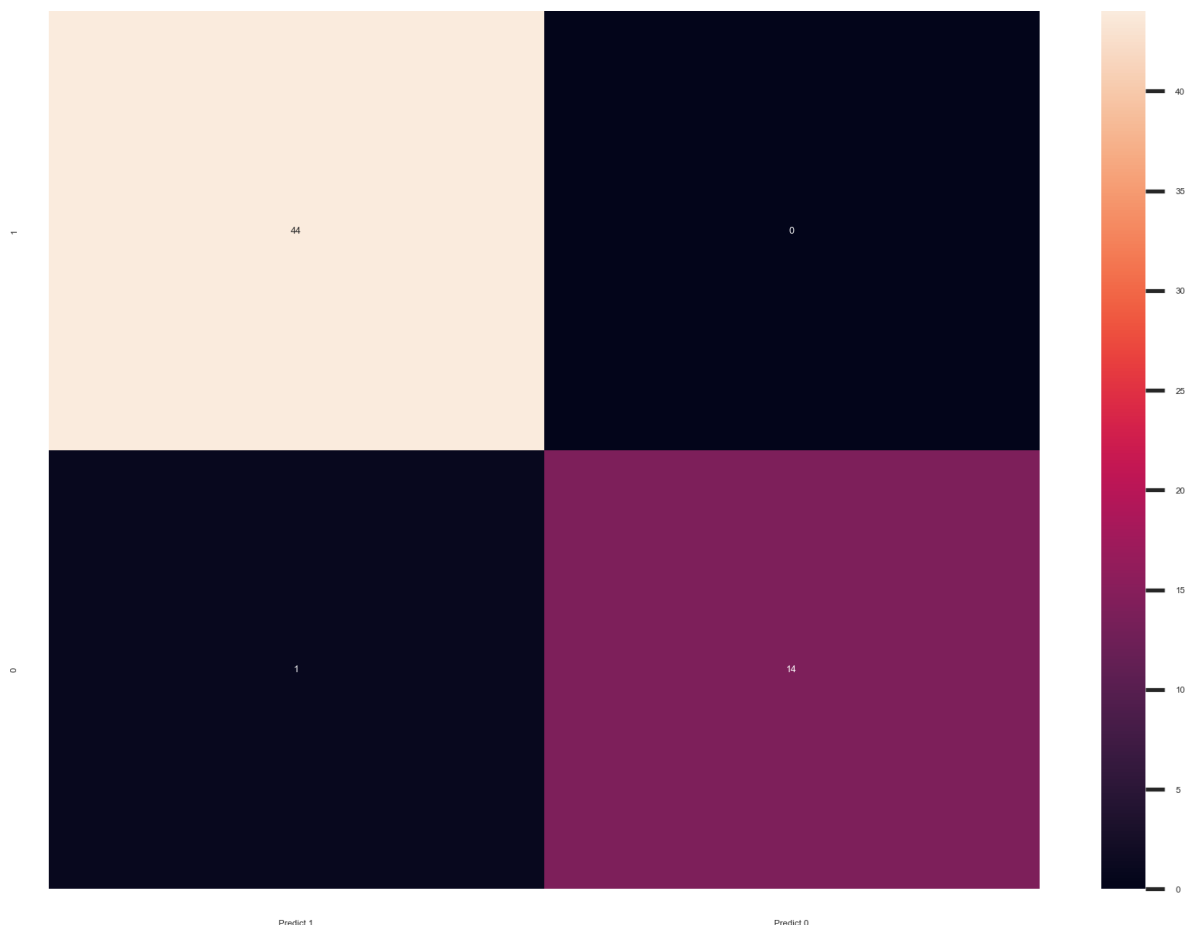
***Fig 4.4 Gender count***

## **4.2.2 DATA PRE-PROCESSING**

The main aim of this step is to study and understand the nature of data that was acquired in the previous step and also to know the quality of data. A real-world data generally contains noises, missing values, and maybe in an unusable format that cannot be directly used for machine learning models. Data pre-processing is a required task for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model. Identifying duplicates in the dataset and removing them is also done in this step.

Actually, in this dataset, we have 755 features out of which some may not be useful in building our model. So, we have to leave out all those unnecessary features which are not responsible to produce the output. If we take more features in this model the

accuracy we got is less. When we check the correlation of the features, some of them are the same. In Fig 4.5, a screenshot of our notebook is shown the correlation of the columns where two of the columns have similar values. So, one of them is removed.



**Fig 4.5 Correlation matrix**

### 4.2.3 FEATURE EXTRACTION

As the correlation values of the two attributes are similar and one of them can be removed. This kind of feature must be dropped. As our data is now stored as a data frame in a python notebook, we can easily drop those unnecessary features using the inbuilt functions. In Fig 4.6, a screenshot of our notebook is shown where we have dropped some features.

```
[10] df.drop(df.iloc[:, 10:755], inplace = True, axis = 1)
df.head()
```

|   | id | gender | PPE     | DFA     | RPDE    | numPulses | numPeriodsPulses | meanPeriodPulses | stdDevPeriodPulses | locPctJitter |
|---|----|--------|---------|---------|---------|-----------|------------------|------------------|--------------------|--------------|
| 0 | 0  | 1      | 0.85247 | 0.71826 | 0.57227 | 240       | 239              | 0.008064         | 0.000087           | 0.00218      |
| 1 | 0  | 1      | 0.76686 | 0.69481 | 0.53966 | 234       | 233              | 0.008258         | 0.000073           | 0.00195      |
| 2 | 0  | 1      | 0.85083 | 0.67604 | 0.58982 | 232       | 231              | 0.008340         | 0.000060           | 0.00176      |
| 3 | 1  | 0      | 0.41121 | 0.79672 | 0.59257 | 178       | 177              | 0.010858         | 0.000183           | 0.00419      |
| 4 | 1  | 0      | 0.32790 | 0.79782 | 0.53028 | 236       | 235              | 0.008162         | 0.002669           | 0.00535      |

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 756 entries, 0 to 755
Data columns (total 10 columns):
#   Column              Non-Null Count  Dtype
---  -
0   id                   756 non-null   int64
1   gender               756 non-null   int64
2   PPE                  756 non-null   float64
3   DFA                  756 non-null   float64
4   RPDE                 756 non-null   float64
5   numPulses            756 non-null   int64
6   numPeriodsPulses     756 non-null   int64
7   meanPeriodPulses     756 non-null   float64
8   stdDevPeriodPulses   756 non-null   float64
9   locPctJitter         756 non-null   float64
dtypes: float64(6), int64(4)
```

**Fig 4.6 Dropping unnecessary features from data frame**

After identifying and dropping some features, the initial 755 features that we have are reduced to 10 features. Those features are as follows:

- Id
- Gender
- PPE(Pitch Period Entropy)
- DFA(Detrended Fluctuation Analysis)
- RPDE(Recurrent Period Density Entropy)
- numPulses
- numPeriodPulses
- meanPeriodPulses
- stdDevPeriodPulses
- locPctJitter

## 4.2.4 FEATEUE SELECTION

After pre-processing the acquired data, the next step is to identify the best features. The identified best features should be able to give high efficiency. In Fig 4.7, a screenshot of our notebook is shown how to select k best features using sklearn. The classes within the sklearn. feature\_selection module are often used for feature

selection/dimensionality reduction on sample sets, either to enhance estimators' accuracy scores or to spice up their performance on very high-dimensional datasets.

```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif
y=df["class"]
x=df.iloc[:,2:10]
xnew1=SelectKBest(f_classif, k=5).fit_transform(x, y)
```

```
x.head()
```

|   | PPE     | DFA     | RPDE    | numPulses | numPeriodsPulses |
|---|---------|---------|---------|-----------|------------------|
| 0 | 0.85247 | 0.71826 | 0.57227 | 240       | 239              |
| 1 | 0.76686 | 0.69481 | 0.53966 | 234       | 233              |
| 2 | 0.85083 | 0.67604 | 0.58982 | 232       | 231              |
| 3 | 0.41121 | 0.79672 | 0.59257 | 178       | 177              |
| 4 | 0.32790 | 0.79782 | 0.53028 | 236       | 235              |

```
x=pd.DataFrame(xnew1)
x.head()
```

|   | 0       | 1       | 2       | 3     | 4     |
|---|---------|---------|---------|-------|-------|
| 0 | 0.85247 | 0.71826 | 0.57227 | 240.0 | 239.0 |
| 1 | 0.76686 | 0.69481 | 0.53966 | 234.0 | 233.0 |
| 2 | 0.85083 | 0.67604 | 0.58982 | 232.0 | 231.0 |

**Fig 4.7 Process of Feature selection and sample data**

## 4.2.5 TRAINING DATA

Splitting the dataset into Training set and testing set:

In machine learning data preprocessing, we have to break our dataset into both training set and test set. This is often one among the crucial steps of knowledge preprocessing as by doing this, we will enhance the performance of our machine learning model.

Suppose, if we've given training to our machine learning model by a dataset and that we test it by a totally different dataset. Then, it'll create difficulties for our model to know the correlations between the models.

If we train our model alright and its training accuracy is additionally very high, but we offer a replacement dataset there to, then it'll decrease the performance. So we always attempt to make a machine learning model which performs well with the training set and also with the test dataset.

```
from sklearn.metrics import classification_report, precision_score, recall_score, f1_score, roc_auc_score, accuracy_score
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state=1)
score_list=[]
```

**Fig 4.8 Splitting dataset into training data and test data**

Usually, we split the dataset into train and test in the ratio of 7:3 i.e., 75 percent of data is used for training and 25 percent of data is used for testing the model. We have done it in the same way and it has been shown in the above Fig 4.8.

## **CHAPTER 5**

### **IMPLEMENTATION DETAILS**

#### **5.1 DEVELOPMENT AND DEPLOYMENT SETUP**

Step 1: Import the required packages.

Step 2: Download the dataset and link it to the Jupyter Notebook,

Step 3: Read the dataset and perform operations on data.

Step 4: Data cleaning.

Step 5: Data Preprocessing.

Step 6: Saving the cleaned car data set after performing operations on data.

Step 7: Start training the Machine learning Model.

Step 8: Split features and target as x and y respectively.

Step 9. Split the new data into 75% of Training data and 25% of Testing data.

Step 10: Train the model with Training data and Testing data.

Step 11: Implementing one hot encoder and column transformer to model.

Step 12: Applying KNN, SVM, Logistic Regression, Linear Regression,

Random Forest, Decision Tree and XGBoost to the model.

Step 13: Fit the all Algorithms Models.

Step 14: If accuracy is good use the model for prediction else fit the model again.  
using other random states.

Step 15: Dump the KNN model into our files using pickle.

Step 16: Open Pycharm and extract the cleaned parkinson.csv and  
KNN Model.pd files into our project.

Step 17: Reading the model and dataset, make the prediction using python and flask from webpage.

## **5.2 ALGORITHMS**

### **5.2.1 KNN**

K-Nearest Neighbors (KNN) is a simple yet effective supervised learning algorithm used for classification and regression tasks. It operates by finding the k-nearest labeled data points to a new, unlabeled point and assigning the majority class label (for classification) or the average value (for regression) among those neighbors. KNN doesn't make any assumptions about the underlying data distribution and is easy to implement. However, its performance can suffer with large datasets and high-dimensional data due to its computational intensity during prediction and sensitivity to the choice of distance metric and value of k. Despite these limitations, KNN finds applications in various domains including recommendation systems, pattern recognition, disease diagnosis, credit risk assessment, and anomaly detection.

### **5.2.2 LINEAR REGRESSION**

Linear regression is a fundamental statistical technique used for modeling the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data. The simplest form, known as simple linear regression, involves only one independent variable, while multiple linear regression deals with more than one. The model assumes that there's a linear relationship between the independent and dependent variables and aims to find the best-fitting line that minimizes the difference between the observed and predicted values. This is typically done by minimizing the sum of the squared differences between the observed and predicted values, a method called ordinary least squares. Linear regression is widely used in various fields, including economics, finance, engineering, and social sciences, for tasks such as forecasting, trend analysis, and identifying relationships between variables. However, it's important to note that linear regression has

assumptions, including linearity, independence, homoscedasticity, and normally distributed residuals, which should be checked before applying the model.

### **5.2.3 LOGESTIC REGRESSION**

Logistic regression is a statistical model used for binary classification tasks, where the outcome variable is categorical and has only two possible outcomes, usually encoded as 0 and 1. Unlike linear regression, which predicts continuous outcomes, logistic regression predicts the probability of the occurrence of a certain event by fitting a logistic function to the observed data. The logistic function, also known as the sigmoid function, maps any real-valued number into the range of  $[0, 1]$ , which is interpreted as the probability of the event happening. Logistic regression estimates the coefficients of the independent variables in such a way that maximizes the likelihood of observing the given outcomes. The model assumes a linear relationship between the independent variables and the log odds of the dependent variable. Logistic regression is widely used in various fields, including medicine, marketing, finance, and social sciences, for tasks such as predicting the likelihood of disease occurrence, customer churn, credit default, and voter choice. It's important to note that logistic regression makes assumptions such as linearity of independent variables and absence of multicollinearity, and it's sensitive to outliers. Additionally, logistic regression can be extended to handle multinomial and ordinal outcomes through multinomial and ordinal logistic regression, respectively.

### **5.2.4 DECISION TREE**

A decision tree is a supervised machine learning algorithm used for both classification and regression tasks. It operates by recursively partitioning the data into subsets based on the values of features, with the goal of minimizing impurity or maximizing information gain at each split. Each node in the tree represents a decision based on a feature, leading to branches representing possible outcomes, and leaf nodes representing the final predictions or classifications. Decision trees are intuitive to interpret and visualize,

making them popular in various domains. They can handle both numerical and categorical data and automatically select important features. However, decision trees are prone to overfitting, especially with complex trees. Techniques like pruning, ensemble methods (e.g., Random Forests), and using appropriate hyperparameters can help mitigate overfitting. Decision trees find applications in fields such as finance, medicine, and marketing for tasks like customer segmentation, fraud detection, and medical diagnosis.

### **5.2.5 SUPPORT VECTOR MACHINE**

Support Vector Machine (SVM) is a versatile supervised learning algorithm widely used for classification, regression, and outlier detection tasks. It operates by identifying the optimal hyperplane that separates data points of different classes, maximizing the margin between classes while minimizing classification errors. SVM is effective in handling linearly separable data and can also handle non-linear data by mapping it into a higher-dimensional space using the kernel trick. This technique allows SVM to capture complex relationships between features. Despite its robustness against overfitting, especially in high-dimensional spaces, SVM requires careful parameter tuning and can be computationally expensive, particularly with large datasets. Nevertheless, SVM finds applications across diverse fields such as text categorization, image classification, bioinformatics, and finance, where clear class separation or dealing with complex data structures is crucial for accurate predictions and analysis.

### **5.2.6 RANDOM FOREST**

Random Forest is an ensemble learning algorithm that operates by constructing a multitude of decision trees during training and outputting the class that is the mode of the classes (classification) or the mean prediction (regression) of the individual trees. Each decision tree is constructed using a random subset of the training data and a random subset of the features. This randomness helps to decorrelate the trees, reducing overfitting and improving the model's generalization performance. The final

prediction is then made by aggregating the predictions of all the trees. Random Forest is known for its high accuracy, robustness against overfitting, and resistance to noise in the data. It is widely used in various fields such as finance, healthcare, and bioinformatics for tasks like classification, regression, and anomaly detection. Additionally, Random Forest provides insights into feature importance, making it a valuable tool for feature selection and understanding the underlying data patterns.

### **5.2.7 XGBOOST**

XGBoost, short for eXtreme Gradient Boosting, is an optimized and scalable implementation of the gradient boosting machine learning algorithm. It's widely used for classification, regression, and ranking problems. XGBoost builds multiple decision trees sequentially, where each tree corrects the errors made by the previous ones. During training, it minimizes a predefined loss function by gradient descent. XGBoost incorporates several techniques to enhance performance, including regularization, parallel computing, and tree pruning, resulting in faster training times and improved accuracy compared to traditional gradient boosting methods. It also supports various objective functions and evaluation metrics, making it flexible and suitable for a wide range of tasks. XGBoost has been highly successful in various data science competitions and is widely adopted in both industry and academia for its exceptional performance and robustness.

## **5.3 TESTING**

Testing plays a crucial role in the development and validation of machine learning (ML) models for predicting Parkinson's disease (PD). Parkinson's disease is a complex neurodegenerative disorder that affects millions of people worldwide, making accurate and early diagnosis essential for effective management and treatment. ML techniques offer promising avenues for improving PD diagnosis by leveraging clinical data, biomarkers, and other relevant features to predict the likelihood of PD occurrence. In the context of PD prediction, testing refers to the process of evaluating the

performance and effectiveness of developed ML models in accurately classifying individuals as either PD-positive or PD-negative based on their features or clinical characteristics. This evaluation is critical for assessing the reliability, generalization ability, and clinical utility of the predictive models, ultimately informing their potential adoption in clinical practice.

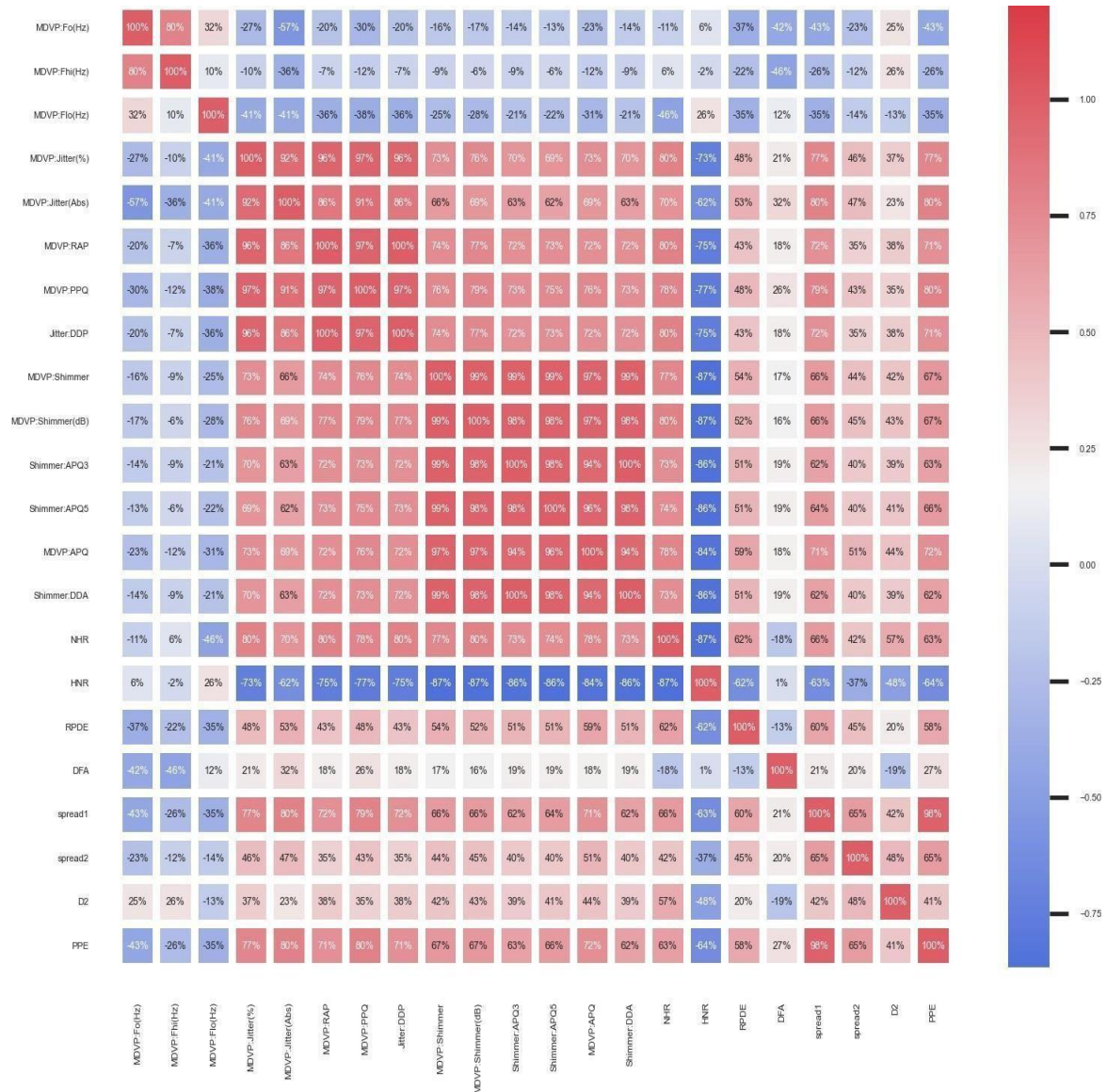
Testing involves several key components, including data preprocessing, model selection, training, evaluation, and performance analysis. Researchers and practitioners employ various evaluation metrics, such as accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC), to assess the predictive performance of ML models. Additionally, techniques like cross-validation and validation on independent datasets are used to ensure the robustness and generalization ability of the predictive models.

The ultimate goal of testing in PD prediction is to develop accurate, reliable, and clinically relevant diagnostic tools that can assist healthcare professionals in early detection and intervention. By rigorously testing ML models on diverse datasets and evaluating their performance against established standards, researchers aim to advance the field of PD diagnosis and contribute to improved patient outcomes and quality of life.

## CHAPTER 6

### RESULT AND DISCUSSION

The performance analysis of the Parkinson's disease datasets has a variety of features. There are 21 features of all in which only 12 features play an important role for predicting the best algorithm in the classification.



**Fig 6.1 Relationship between data values on heat map**

The colors in the graph is to communicate relationships between data values that would be much harder to understand if presented numerically in a spreadsheet. The red squares are the important features which we use them for the classification. The blue squares are the negative values that are considered less in the algorithms.

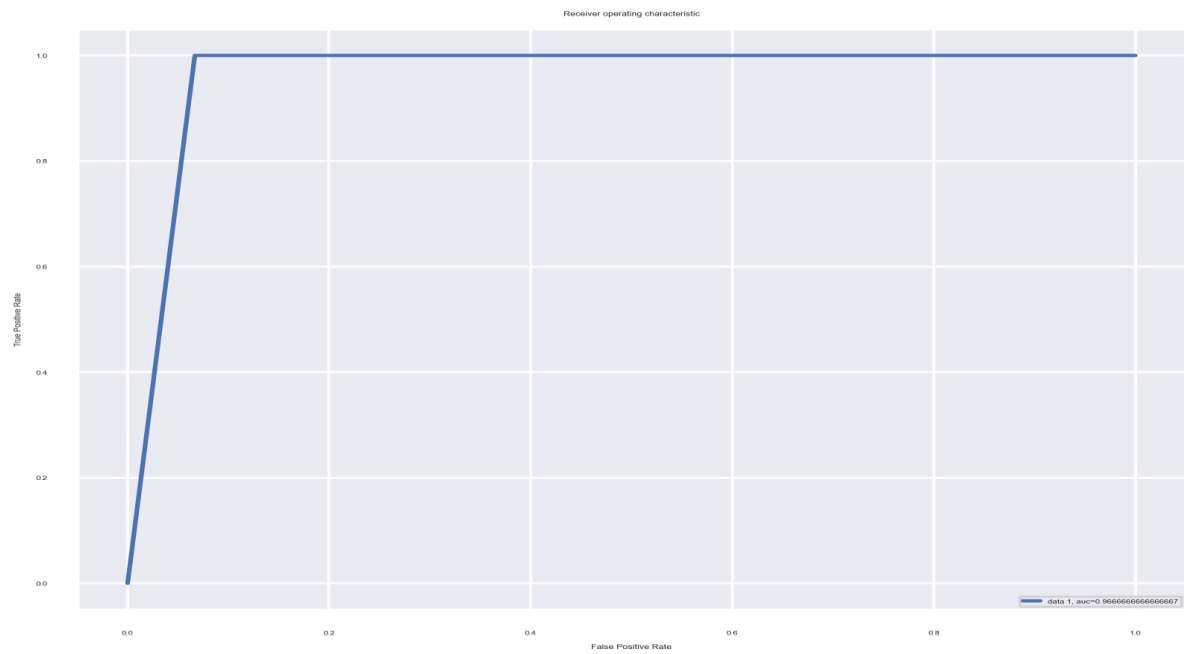
## 6.1 ANALYSIS OF THE RESULTS

In this section, we conducted various experiment to evaluate the seven-machine learning supervised algorithms for detection of Parkinson Disease. The analysis of three classification techniques were evaluated for the detection of Parkinson disease data. shows the accuracy of seven supervised techniques.

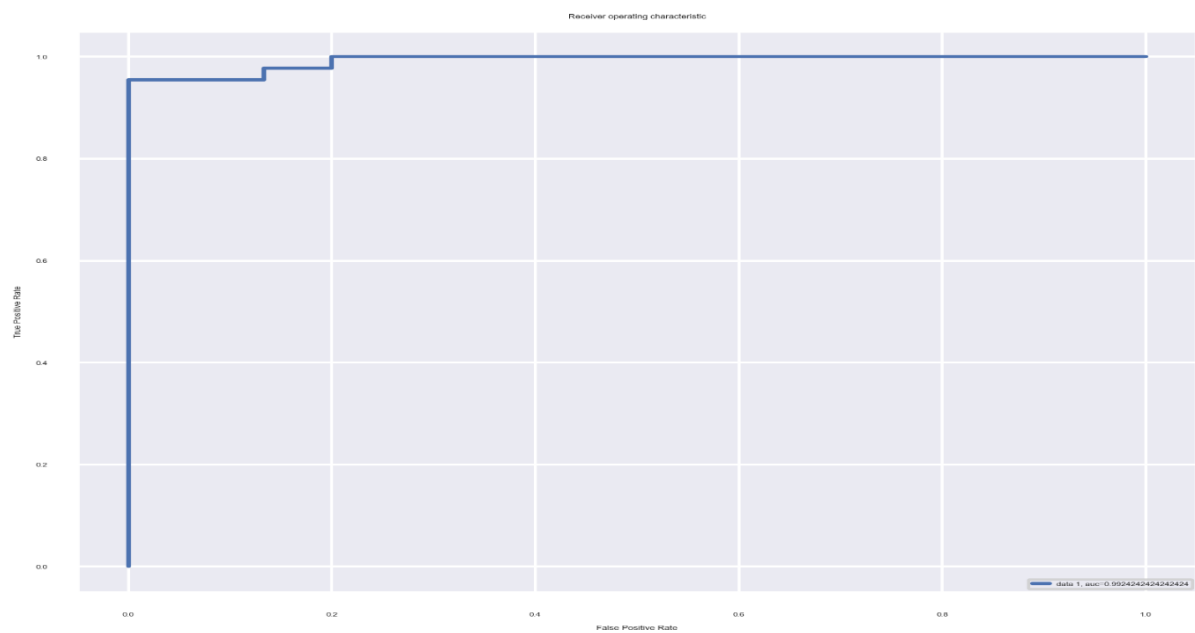
|   | Model Name               | Train Accuracy(%) | Test Accuracy(%) | AUC Score |
|---|--------------------------|-------------------|------------------|-----------|
| 0 | Logistic Regression      | 82.524272         | 72.881356        | 0.889394  |
| 1 | Decision Tree Classifier | 77.184466         | 74.576271        | 0.881818  |
| 2 | AdaBoost                 | 82.038835         | 86.440678        | 0.921970  |
| 3 | Random Forest Classifier | 100.000000        | 89.830508        | 0.972727  |
| 4 | kNN                      | 100.000000        | 98.305085        | 0.966667  |
| 5 | SVM                      | 100.000000        | 94.915254        | 0.992424  |
| 6 | XGBoost                  | 100.000000        | 93.220339        | 0.942424  |

**Fig 6.2 Accuracy Table**

We can say that KNN Model is good for our dataset but SVM giving more AUC. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

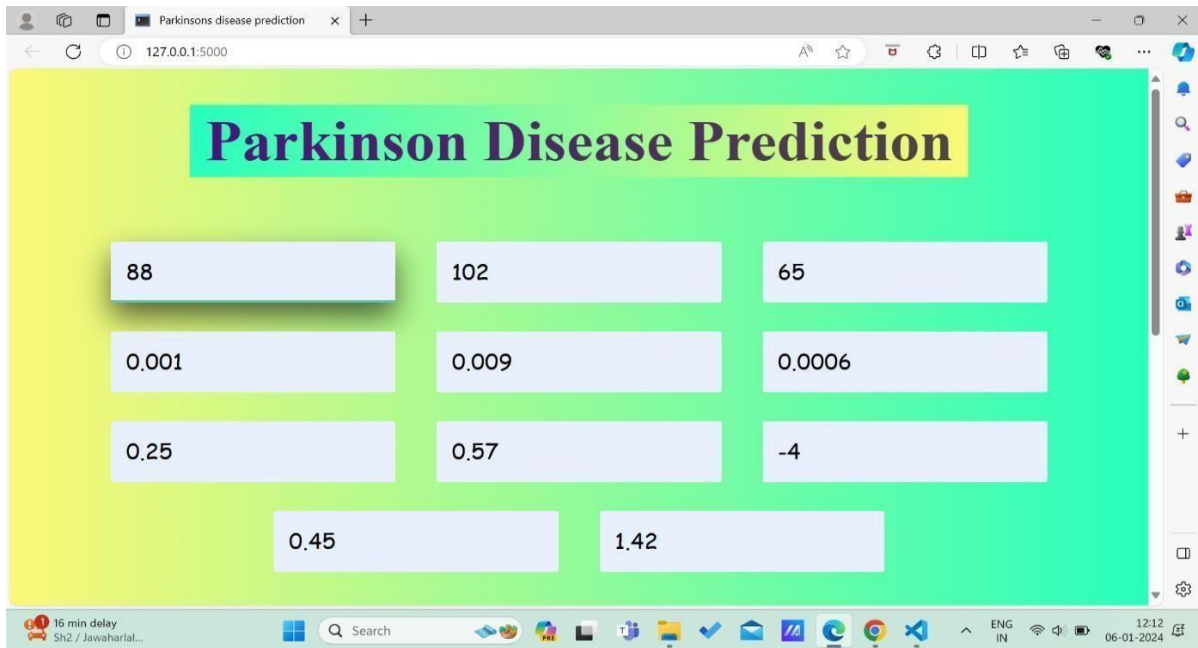


**Fig 6.3 Accuracy graph of KNN**

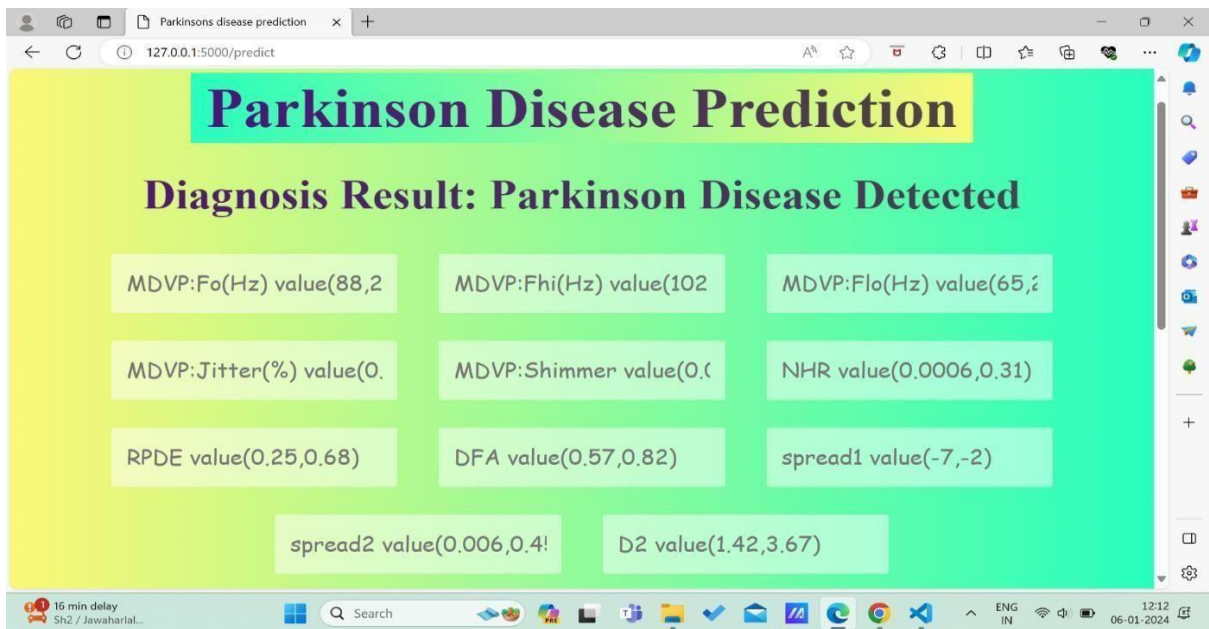


**Fig 6.4 Accuracy graph of KNN**

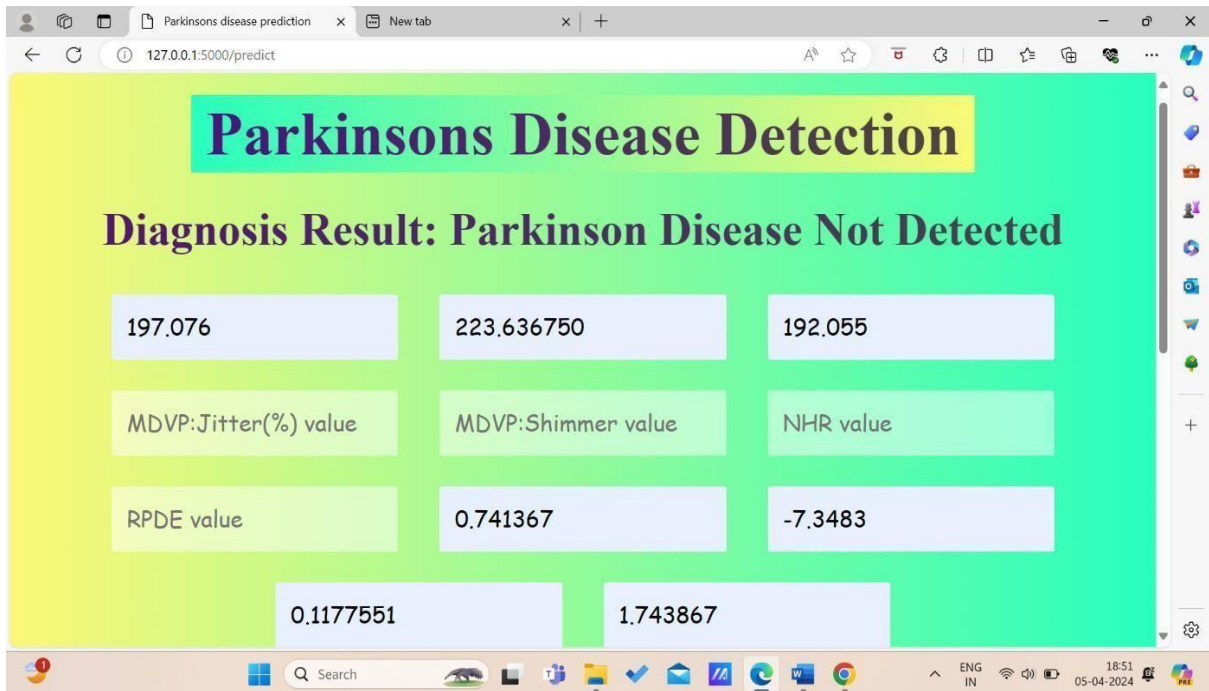
KNN is best model if you want less False Positive, SVM model is best if you are seeing for good AUC Score.



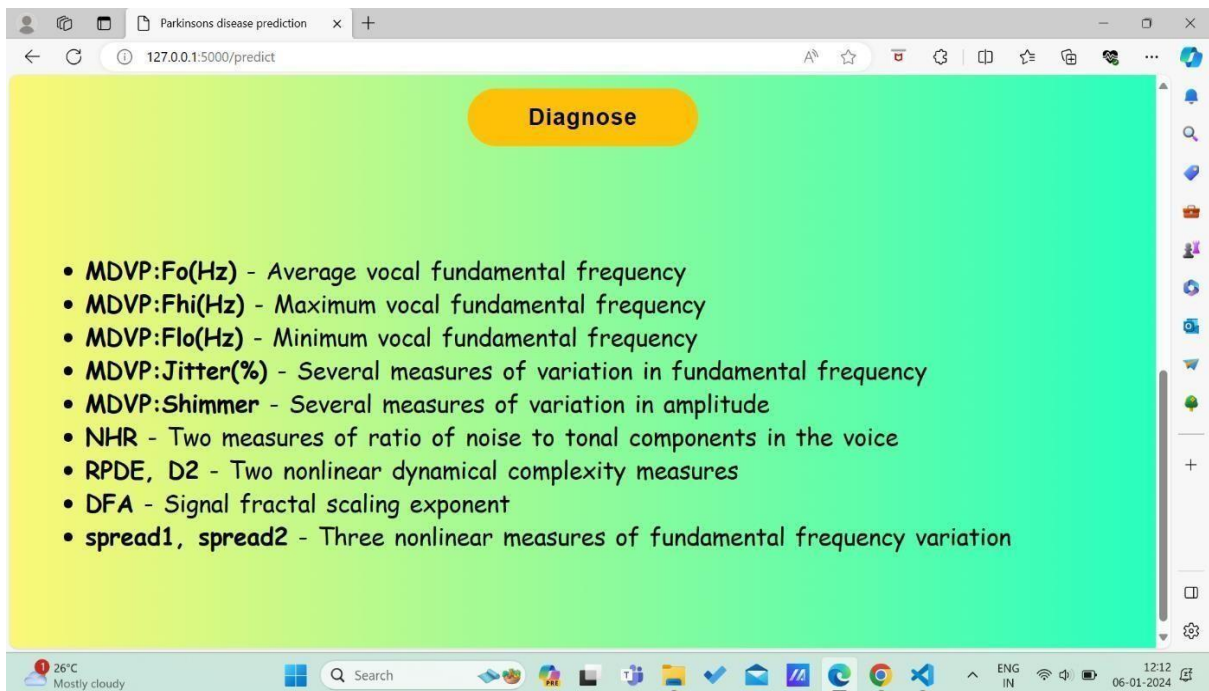
**Fig 6.5 User have to give input**



**Fig 6.6 Parkinson Disease Detected**



**Fig 6.7 Parkinson Disease Not Detected**



**Fig 6.8 Attribute Information**

## **CHAPTER 7**

### **CONCLUSION**

A genetic algorithm is the most common method used to extract voice signal characteristics. For analysis, the common features considered are jittering, shimmering and voice signal frequency. Various algorithms for machine learning have been used in the prediction of Parkinson's disease. The range of prediction accuracy from applied algorithms lies between 88% and 99%, which demonstrates that machine learning algorithms can be used to predict Parkinson's disease by analyzing voice signals. Here the classification accuracy was studied and compared, with good performance and fast implementation, KNN achieved a high accuracy with 96%. This system provides the comparison between machine learning classifiers of Logistic regression, Decision tree, K-Nearest neighbor, Linear Regression, SVM, Random Forest and XGBoost in PD disease diagnosis with high dimensional data MDVP:Jitter(%), Jitter(Abs), MDVP:RAP, MDVP:PPQ, Jitter:DPP, MDVP: Shimmer, MDVP:Shimmer(db), Shimmer APQ3, Shimmer APQ5, MDVP:APQ, Shimmer:DDA, NHR are the selected features from the dataset to achieve high accuracy in prediction.

## REFERENCES

- [1] A. Tsanas, M. A. Little, P. E. McSharry, J. Spielman and L. O. Ramig, "Novel Speech Signal Processing Algorithms for HighAccuracy Classification of Parkinson's Disease," in IEEE Transactions on Biomedical Engineering, vol. 59, no. 5, pp. 1264- 1271, May 2012.
- [2] Abós, A. et al. Discriminating cognitive status in Parkinson's disease through functional connectomics and machine learning. Scientific reports 7, 45347 (2017).
- [3] B. Zhang, J. Ren, Y. Cheng, B. Wang and Z. Wei, "Health Data Driven on Continuous Blood Pressure Prediction Based on Gradient Boosting Decision Tree Algorithm," in IEEE Access, vol. 7, pp. 32423-32433, 2019.
- [4] C. B. Levine, K. R. Fahrbach, A. D. Siderowf, R. P. Estok, V. M. Ludensky, and S. D. Ross, "Diagnosis and treatment of Parkinson's disease: a systematic review of the literature," Evid. Rep. Technol. Assess. (Summ.), no. 57, pp. 1–4, May 2003.
- [5] Center for Machine Learning and Intelligent System, Website:<http://Archive.lcs.UCI.Edu/ML/Datasets/Parkinsons,2009>.
- [6] Dobson, A. J. & Barnett, A. An introduction to generalized linear models. (CRC press, 2008).
- [7] Dr. Geetha R Ramani, G Sivagami,"Parkinson Disease Classification Using Data Mining Algorithms", International Journal Of Computer Applications 32(9):17-22, October 2011.
- [8] Gladence, L. Mary, M. Karthi, and V. Maria Anu. "A statistical comparison of logistic regression and different Bayes classification methods for machine learning." ARPN Journal of Engineering and Applied Sciences 10, no. 14 (2015): 5947-5953.
- [9] H. Ramezani and O. B. Akan, "Rate region analysis of multi-terminal neuronal nanoscale communication channel," in 17th IEEE NANO Conf. IEEE, 2017.

- [10] J. Jankovic, "Parkinson's disease: clinical features and diagnosis" Journal of Neurology, Neurosurgery & Psychiatry, pp.368-376, 2008.
- [11] K. Srilatha and V. Ulagamuthalvi "A Comparative Study on Tumor Classification "Research J. Pharm. and Tech 2019; 12(1): 407-411.
- [12] Mercy Paul Selvan, Nagubadi. Navadurga, Nimmagadda. Lakshmi Prasanna, "An Efficient Model for Predicting Student Dropout using Data Mining and Machine Learning Techniques", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue- 9S2, July 2019.
- [13] Mohammad S Islam ; Imtiaz Parvez ; Hai Deng ; Parijat Goswami," Performance Comparison Of Heterogeneous Classifiers For Detection Of Parkinson's Disease Using Voice Disorder (Dysphonia)", International Conference On Informatics, Electronics & Vision (Iciev), Page(S):1 – 7.
- [14] Mohammad shahbakhti, Danial Taherifar, Linear and Non-Linear speech features for detection of Parkinson's Disease, BMEiCON-2013.
- [15] R. Arefi Shirvan, E. Tahami, Voice analysis for detecting Parkinson's disease using genetic algorithm and KNN classification method, Proc 18th Int Con on Biomedical Engineering, Tehran, pp. 550-555, 2011.
- [16] Rätsch, G., Onoda, T. & Müller, K.-R. Soft margins for AdaBoost. Machine learning 42, 287–320 (2001).
- [17] Resuldas, "A Comparison of Multiple Classification Methods For Diagnosis Of Parkinson Disease", Expert Systems With Applications, Volume 37, Issue 2, March 2010, Pages 1568-1572.
- [18] Shraddha Khonde and V. Ulagamuthalvi(2019), Fusion of feature selection and Random Forest for an Anomaly based intrusion detection system , Journal of Computational and Theoretical Nanoscience , Volume 16, PP 3603-3607.

## APPENDIX

### A. SOURCE CODE

```
# Cross Validation

from sklearn.model_selection import RepeatedStratifiedKFold

from sklearn.model_selection import StratifiedKFold

from sklearn.model_selection import cross_val_score

# Importing necessary libraries used for data cleaning, and data visualization

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

# Ignoring unnecessary warnings

import warnings

warnings.filterwarnings("ignore")

# Chi Square

from sklearn.feature_selection import chi2

import scipy.stats as stats

# RandomOverSampler to handle imbalanced data

from imblearn.over_sampling import RandomOverSampler

# Constant feature checking

from sklearn.feature_selection import VarianceThreshold

from sklearn.model_selection import GridSearchCV
```

```

from sklearn.model_selection import RandomizedSearchCV

# Importing library to split the data into training part and testing part.

from sklearn.model_selection import train_test_split

# Importing library to process the data (Normalize the data)

from sklearn.preprocessing import MinMaxScaler, StandardScaler

# Importing Models (used for making prediction)

from sklearn.linear_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import AdaBoostClassifier

from sklearn.svm import SVC

from xgboost import XGBClassifier

from sklearn.neighbors import KNeighborsClassifier

from scipy.stats import zscore

# Importing metrics used for evaluation of our models

from sklearn import metrics

from sklearn.metrics import classification_report

# Loading the data from CSV file to pandas dataframe
df = pd.read_csv("data.csv")

# Printing first five rows of the data
df.head()

# Printing last five rows of the data
df.tail()

# Checking the number of rows and columns in our dataset
df.shape

```

```

# Getting more information of our dataset
df.info()
df.isnull().sum()
# Getting some statistical information of our data
df.describe()
# distribution of our target variable i.e. "status" column/feature
df["status"].value_counts()
sns.pairplot(df.drop("name", axis=1), hue="status", diag_kind="kde")
sns.set(font_scale=0.25)
cmap = sns.diverging_palette(260, 10, as_cmap=True)
sns.heatmap(df.drop(["name", "status"], axis=1).corr("spearman"), vmax=1.2,
annot=True, square='square', cmap=cmap, fmt = '.0%', linewidths=2)
# With the following function we can select highly correlated features
# It will remove the first feature that is correlated with anything other feature

def correlation(dataset, threshold):
    col_corr = set() # Set of all the names of correlated columns
    corr_matrix = dataset.corr("spearman")
    for i in range(len(corr_matrix.columns)):
        for j in range(i):
            if abs(corr_matrix.iloc[i, j]) > threshold: # we are interested in absolute coeff
value
                colname = corr_matrix.columns[j] # getting the name of column
                col_corr.add(colname)
    return col_corr
corr_features = correlation(df.drop(["name", "status"], axis=1), 0.85)
corr_features

df_anova = df.drop(["name"], axis=1)

grps = pd.unique(df_anova.status.values)

coldrop = []

```

```

for i in range(len(df_anova.columns)-1):

    d_data = {grp:df_anova[df_anova.columns[i]][df_anova.status == grp] for
               grp in grps}

    F, p = stats.f_oneway(d_data[0], d_data[1])

    print("P_Value of {} and status".format(df_anova.columns[i]), p)

    if p < 0.05:

        print("There is relation between {} and status
              \n".format(df_anova.columns[i]))

    else:

        print("There is no relation between {} and status
              \n".format(df_anova.columns[i]))

        coldrop.append(df_anova.columns[i])

df.drop("name", axis=1).columns
corr_features = list(corr_features)

list_drop = corr_features + coldrop
list_drop.append("name")
list_drop

X_train, X_test, y_train, y_test = train_test_split(feature, y, test_size=0.3,
random_state=42)
X_train
y_train.head()

```

```

X_test
cm = metrics.confusion_matrix(y_test, y_pred_lr, labels=[1,0])

df_cm = pd.DataFrame(cm, index = [i for i in ["1","0"]],
                      columns = [i for i in ["Predict 1", "Predict 0"]])
plt.figure(figsize = (7,5))
sns.heatmap(df_cm, annot=True, fmt='g')
y_pred_proba = lr.predict_proba(X_test)[:,1]
fpr, tpr, _ = metrics.roc_curve(y_test, y_pred_proba)
fpr
tpr

auc = metrics.roc_auc_score(y_test, y_pred_proba)
auc

plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
def dtree_grid_search(X, y):
    #create a dictionary of all values we want to test
    param_grid = { 'criterion':['gini','entropy'],'max_depth': np.arange(2, 15)}
    cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)

    # decision tree model
    dtree = DecisionTreeClassifier()

    #use gridsearch to test all values

```

```

    dtree_gscv = GridSearchCV(dtree, param_grid, cv=cv, n_jobs=-1,
scoring='accuracy')
    #fit model to data
    dtree_gscv.fit(X, y)

    return dtree_gscv.best_params_
dtree_grid_search(X_train, y_train)
dTree = DecisionTreeClassifier(criterion = 'entropy', max_depth = 2)
dTree.fit(X_train, y_train)

print(dTree.score(X_train,y_train))
print(dTree.score(X_test,y_test))

y_pred_dtree = dTree.predict(X_test)

print(metrics.classification_report(y_test, y_pred_dtree))

def knn_grid_search(X, y):
    #create a dictionary of all values we want to test
    k_range = list(range(1, 31))
    param_grid = dict(n_neighbors=k_range)
    cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)

    knn = KNeighborsClassifier()

    #use gridsearch to test all values
    knn_gscv = GridSearchCV(knn, param_grid, cv=cv, scoring='accuracy', n_jobs=-1)
    #fit model to data
    knn_gscv.fit(X, y)

    return knn_gscv.best_params_

```

```

knn_grid_search(X_train, y_train)
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(X_train, y_train)

y_pred_knn = knn.predict(X_test)

print(knn.score(X_train, y_train))
print(knn.score(X_test, y_test))

print(metrics.classification_report(y_test, y_pred_knn))
knn_tacc = knn.score(X_test, y_test)
knn_train_acc = knn.score(X_train, y_train)
cm = metrics.confusion_matrix(y_test, y_pred_knn, labels=[1,0])

df_cm = pd.DataFrame(cm, index = [i for i in ["1","0"]],
                      columns = [i for i in ["Predict 1", "Predict 0"]])
plt.figure(figsize = (7,5))
sns.heatmap(df_cm, annot=True, fmt='g')
y_pred_proba = knn.predict_proba(X_test)[:,:1]
fpr, tpr, _ = metrics.roc_curve(y_test, y_pred_proba)
fpr
tpr

auc = metrics.roc_auc_score(y_test, y_pred_proba)
auc

plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')

```

```

plt.legend(loc="lower right")
knn_auc= auc
knn_auc

def svm_grid_search(X, y):
    #create a dictionary of all values we want to test
    param_grid = {'C': [0.1,1, 10, 100], 'gamma': [1,0.1,0.01,0.001, 0.4, 0.2, 0.8],'kernel':
    ['rbf', 'poly', 'sigmoid']}
    cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)

    svm = SVC()

    #use gridsearch to test all values
    svm_gscv = RandomizedSearchCV(estimator = svm,
                                  param_distributions = param_grid,
                                  scoring = 'accuracy',
                                  cv = cv,
                                  n_jobs = -1)

    #fit model to data
    svm_gscv.fit(X, y)

    return svm_gscv.best_params_
svm_grid_search(X_train, y_train)

from sklearn import svm
svm = SVC(gamma=0.4, C=100, kernel='rbf', probability=True)

svm.fit(X_train, y_train)

y_pred_svm = svm.predict(X_test)
print(svm.score(X_train, y_train))

```

```

print(svm.score(X_test, y_test))

print(metrics.classification_report(y_test, y_pred_svm))

svm_tacc = svm.score(X_test, y_test)
svm_train_acc = svm.score(X_train, y_train)

cm = metrics.confusion_matrix(y_test, y_pred_svm, labels=[1,0])

df_cm = pd.DataFrame(cm, index = [i for i in ["1","0"]],
                      columns = [i for i in ["Predict 1", "Predict 0"]])
plt.figure(figsize = (7,5))
sns.heatmap(df_cm, annot=True, fmt='g')

y_pred_proba = svm.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test, y_pred_proba)
fpr
tpr

auc = metrics.roc_auc_score(y_test, y_pred_proba)
auc

plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")

```

```

def comp_model(model_list, model_train_acc_list, model_test_acc_list,
model_auc_list):
data = {"Model Name": model_list, "Train Accuracy(%)": [i*100 for i in
model_train_acc_list], "Test Accuracy(%)": [i*100 for i in model_test_acc_list], "AUC
Score": model_auc_list}
Comparision = pd.DataFrame(data)
    return Comparision

```

```

model_list = ["Logistic Regression", "Decision Tree Classifier", "AdaBoost", "Random
Forest Classifier", "kNN", "SVM", "XGBoost"]
model_train_acc_list = [lr_train_acc, dt_train_acc, ada_train_acc, rf_train_acc,
knn_train_acc, svm_train_acc, xgb_train_acc]
model_test_acc_list = [lr_tacc, dt_tacc, ada_tacc, rf_tacc, knn_tacc, svm_tacc,
xgb_tacc]
model_auc_list = [lr_auc, dt_auc, ada_auc, rf_auc, knn_auc, svm_auc, xgb_auc]
comp_model(model_list, model_train_acc_list, model_test_acc_list, model_auc_list)

```

#input data and transform into numpy array

```
in_data= np.asarray(tuple(map(float,input().rstrip().split(','))))
```

#reshape and scale the input array

```
in_data_re = in_data.reshape(1,-1)
```

```
in_data_sca = scaler.transform(in_data_re)
```

#print the predicted output for input array

```
print("Parkinson's Disease Detected" if knn.predict(in_data_sca) else "No Parkinson's
Disease Detected")
```

#input data and transform into numpy array

```
in_data= np.asarray(tuple(map(float,input().rstrip().split(','))))
```

```

#reshape and scale the input array
in_data_re = in_data.reshape(1,-1)
in_data_sca = scaler.transform(in_data_re)

#print the predicted output for input array
print("Parkinson's Disease Detected" if svm.predict(in_data_sca) else "No Parkinson's
Disease Detected")

def build_model(hp):
    model = keras.Sequential()
    model.add(layers.Dense(units=hp.Int('units_inp', min_value=32, max_value=512,
step=32),
                           activation=hp.Choice( 'activation', ['tanh', 'relu', 'LeakyReLU',
'elu']),
                           input_dim = 11
                        )
    )

    for i in range(hp.Int('num_layers', 2, 20)):          # Number of layers
        model.add(layers.Dense(units=hp.Int('units_' + str(i),
min_value=32,
max_value=512,      # Number of neuron, here it is 32-
512
step=32),
activation=hp.Choice( 'activation', ['tanh', 'relu', 'LeakyReLU',
'elu'])))

    if hp.Boolean("dropout"):
        model.add(layers.Dropout(rate=0.25))

    model.add(layers.Dense(1, activation='sigmoid'))

```

```

model.compile
optimizer=keras.optimizers.Adam(
    hp.Choice('learning_rate', [1e-2, 1e-3, 1e-4])),
    loss='binary_crossentropy',
    metrics=['accuracy'])
return model
y_test_pred_probs = best_model.predict(X_test)
FPR, TPR, _ = roc_curve(y_test,y_test_pred_probs)

auc = metrics.roc_auc_score(y_test, y_test_pred_probs)
auc

plt.plot(FPR,TPR,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.plot([0,1],[0,1], '--',color='black')
plt.title('ROC Curve')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')

```

## API DEPLOYMENT

```

import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
import pandas as pd

app = Flask(__name__)
model = pickle.load(open('model_knn.pkl','rb'))
scaler = pickle.load(open('scaling.pkl','rb'))

@app.route('/')
def home():

```

```

return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():

    int_features = [float(x) for x in request.form.values()]
    final_features = [np.array(int_features)]

    final_features = scaler.transform(final_features)
    prediction = model.predict(final_features)

    output = prediction

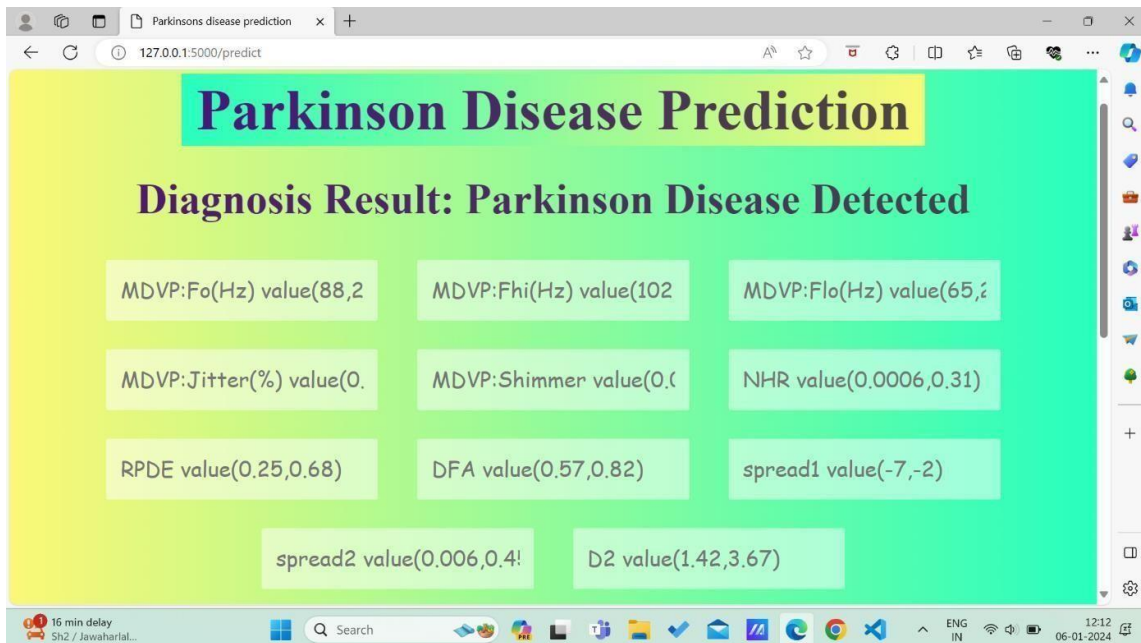
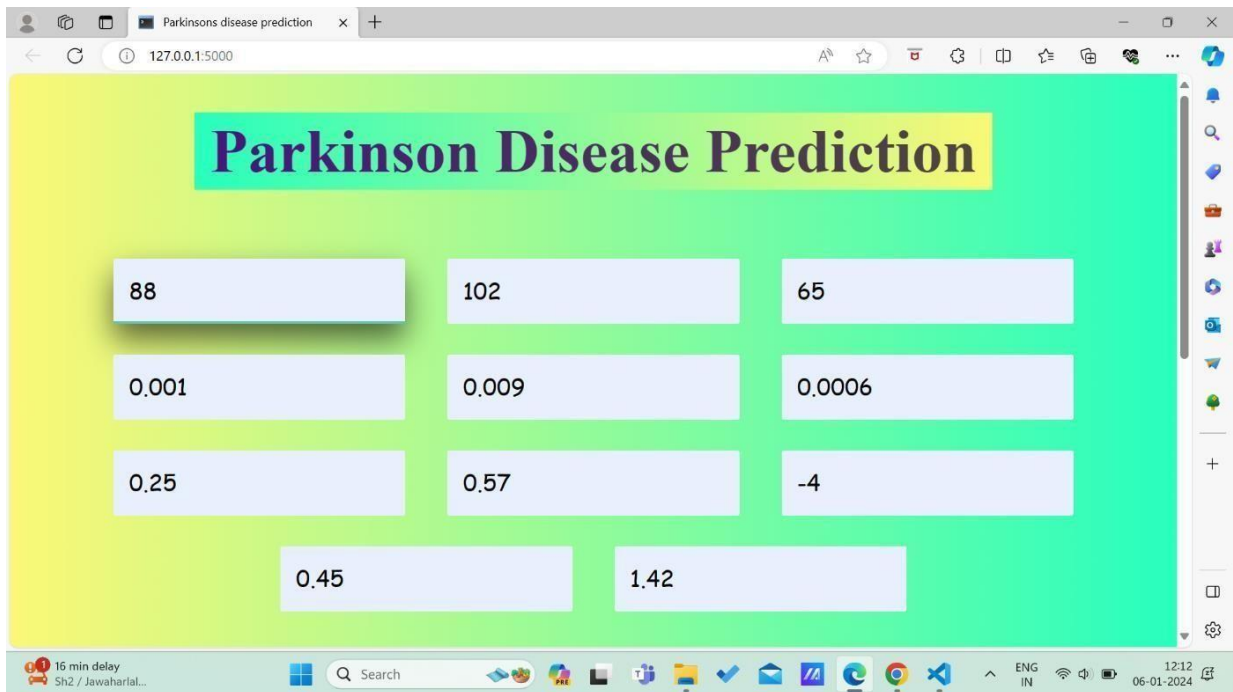
    if output == [0]:
        output = "Parkinson Disease Not Detected"
    elif output == [1]:
        output = "Parkinson Disease Detected"

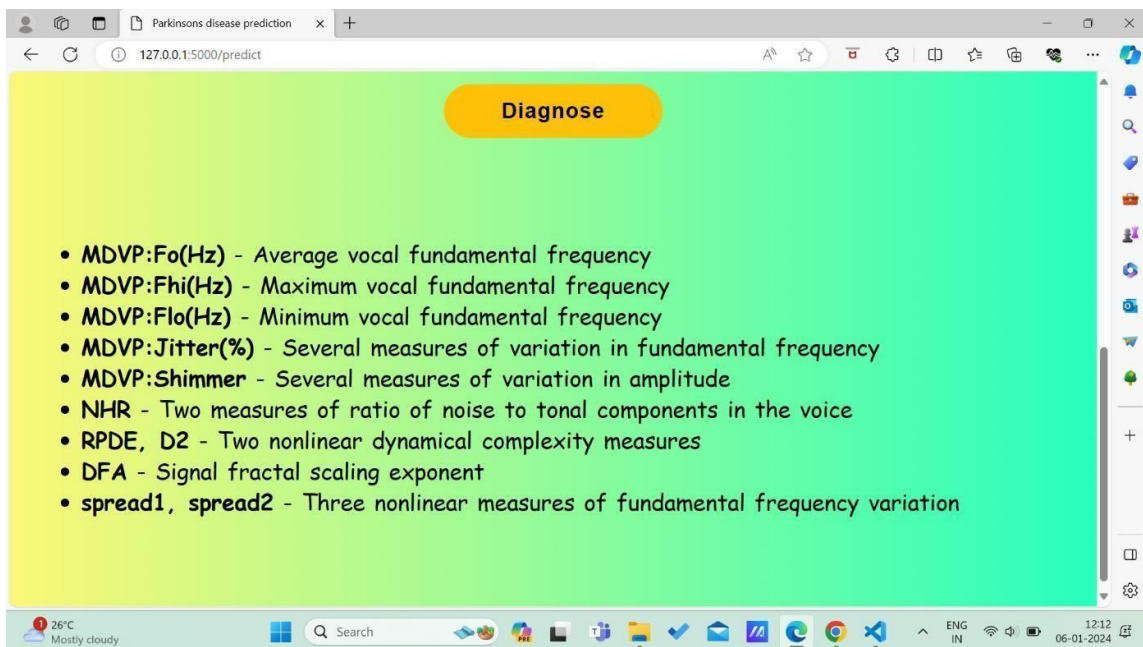
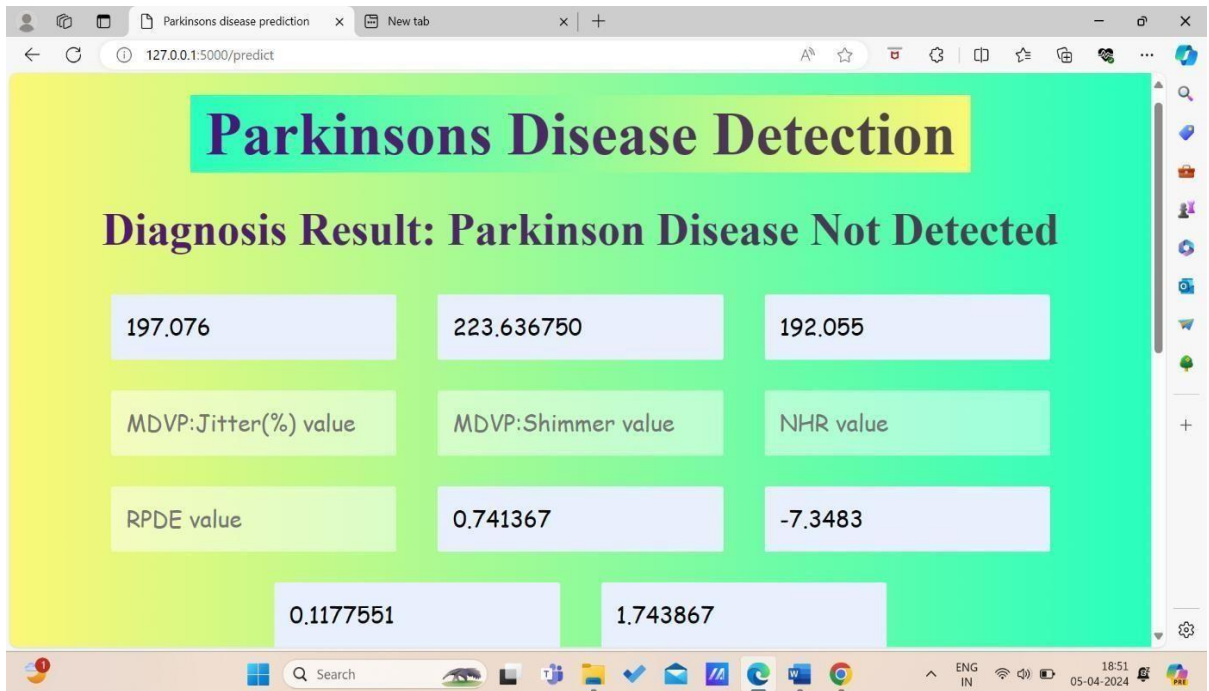
    return render_template('index.html', prediction_text='Diagnosis Result:
{}'.format(output))

if __name__ == "__main__":
    app.run(debug=True)

```

## B. SCREENSHORT





## C. RESEARCH PAPER

### Parkinson's Disease Prediction Using Machine Learning Technique

Vivek Kumar

Department of computer science  
and Engineering  
Sathyabama Institute of Science  
and Technology Chennai, India  
[vivekkumarjaiswal12@gmail.com](mailto:vivekkumarjaiswal12@gmail.com)

Abhishek Kumar

Department of computer science  
and Engineering  
Sathyabama Institute of Science  
and Technology Chennai, India  
[smileabhi@gmail.com](mailto:smileabhi@gmail.com)

Sankari M

Department of computer science  
and Engineering  
Sathyabama Institute of Science  
and Technology Chennai, India  
[sankari.cse@sathyabama.ac.in](mailto:sankari.cse@sathyabama.ac.in)

**Abstract:** Various data sources include sizable medical datasets that are utilized for disease identification. One of the most deadly and progressive disorders of the neurological system that impairs movement is Parkinson's. This neurological condition, which is the second most prevalent, shortens life expectancy, causes disability, and is now incurable. Speech problems are seen in over 90% of those afflicted by this illness. Information is created in real-world applications through a variety of Machine Learning approaches. From it, machine learning algorithms facilitate the production of valuable content. Older people's life expectancy is increased by the application of machine learning algorithms to detect diseases early on. When considering the word "Parkinson's," speech traits are the primary idea. Several machine learning techniques, including KNN, Decision Tree, Ada Boosting, Random Forest, SVM, XG boost, and Logistic Regression, are being utilized in this research to forecast Parkinson's disease (PD) based on user input. The data collection serves as the algorithm's input.

#### I. INTRODUCTION

A neurological condition that affects millions of individuals worldwide is Parkinson's disease (PD). It appears as a variety of motor and non-motor symptoms and progresses gradually. For PD patients to receive prompt and efficient therapy and enhance their quality of life, early diagnosis and precise prediction of the disease are essential. This is the point at which machine learning techniques become useful.

Machine learning, a crucial aspect of artificial intelligence, has significantly altered the landscape of healthcare by enabling the generation of models that forecast patient outcomes based on their data. The deployment of machine learning approaches for Parkinson's disease plays a key role in recognizing patients susceptible to the condition early on and facilitates the customization of treatment strategies. By amalgamating a broad spectrum of data sources, including healthcare documentation, neuroimaging studies, and genetic profiles, machine learning algorithms are instrumental in the rapid detection of Parkinson's disease (PD) and in monitoring its progression.

This project aims to explore the potential of machine

learning in predicting Parkinson's disease. It involves the collection and analysis of data from various sources, the selection of relevant features, and the development of predictive models that can accurately detect the presence or risk of PD. By harnessing the power of data-driven algorithms, we strive to create a tool that aids healthcare professionals in making informed decisions and offers individuals at risk the opportunity for early intervention.

In this project, we aim to investigate the approaches, obstacles, and possible outcomes associated with employing machine learning strategies for predicting Parkinson's disease. Our objective is that this work will further the progress in early detection and tailored treatment for those affected by or susceptible to Parkinson's disease.

The importance of accurately forecasting Parkinson's Disease (PD) using machine learning cannot be overstated. Parkinson's disease is a debilitating condition with no known cure, and early diagnosis is pivotal for improving the lives of patients. By developing accurate prediction models, we can achieve several vital objectives:

1. **Early Intervention:** Early diagnosis enables timely interventions and treatments that can slow down the disease's progression. This may result in those suffering from Parkinson's disease having a higher quality of life.
2. **Personalized Care:** Machine learning models facilitate the tailoring of treatment plans to individual patients. Since the disease manifests differently in each person, personalized care can optimize treatment strategies and outcomes.
3. **Healthcare Cost Reduction:** Early diagnosis and personalized treatment plans can lead to cost savings in healthcare by reducing hospitalizations and long-term care requirements.

#### II. RELATED WORKS

##### a) OPEN PROBLEMS IN EXISTING SYSTEM

##### i) Limited and Imbalanced Datasets:

Robust machine learning models require the availability of vast and diverse datasets for training. But when it comes to Parkinson's disease, datasets may be unbalanced and constrained, which means that particular classes or stages of the disease may be underrepresented.

#### ii) Feature Selection and Extraction:

There are variations in how each person experiences Parkinson's disease. One of the challenges that never goes away is to create tailored models that can adjust to the special traits of every patient. This entails taking age, gender, and comorbidities into consideration.

#### iii) Patient-Specific Models:

Parkinson's disease progression varies among individuals. Creating personalized models that can adapt to the unique characteristics of each patient is an ongoing challenge. This involves accounting for factors like age, gender, and comorbidities.

#### iv) Multi-Modal Data Integration:

Combining information from multiple sources, such as clinical data, neuroimaging, and genetic data, is a complex task. Integrating diverse data types while preserving their individual characteristics poses a challenge in developing comprehensive predictive models.

#### v) Real-Time Monitoring:

Developing models for real-time monitoring of Parkinson's disease symptoms is challenging but essential for continuous patient care. This involves integrating data from wearable devices and other continuous monitoring tools.

#### vi) Clinical Integration:

Bridging the gap between machine learning research and clinical practice is essential. Implementing predictive models in real-world clinical settings and demonstrating their efficacy in improving patient outcomes is a significant challenge.

#### b) Description Of Proposed System

Voice modulations of Parkinson's patients are captured in audio data from UCI and PPMI [21] by the suggested methods. Jitter, shimmer, and MDVP of vowel phonations are all included in the dataset. To fully comprehend the attributes, preprocessed, analyzed, and displayed data is used. models—KNN, Random Forest Regressor, SVM, and Logistic Regression—are trained on 75% of the data. Models are trained to categorize audio data into two groups: healthy and Parkinson's disease-related, based on differences in frequency. A ROC-AUC score, confusion matrix [22], sensitivity, precision, accuracy, and accuracy on 25% of the data are used to evaluate models. The general procedure that was used is shown in Figure 1. It shows how data is ingested from the PPMI database, divided into training and testing sets, four models are trained using the data, and test data is used to validate the findings.

### III. LITERATURE SURVEY

In an effort to find a solution, experts and automakers have investigated this problem statement in great detail over the past five years. Their approaches range from examining different patterns of distracting habits to examining the driver's vital signs.

Research focusing on auditory impairments for the early detection of Parkinson's Disease (PD) is less prevalent, in contrast to earlier works that relied on MRI imaging, locomotion analysis, and genetic markers for prognosis.

In their study, Bilal et al. [7] explored the potential of using the Support Vector Machine (SVM) model to predict the onset of PD in older individuals based on genetic information. They managed to train their SVM model to achieve perfect accuracy. However, the SVM model discussed in our study shows a slightly adjusted accuracy rate of 99.34%, further highlighting the efficacy of auditory data over genetic information for the classification of PD. These findings underscore the potential benefits of employing auditory data for PD diagnosis.

#### a) INFERENCES FROM LITREATURE SURVEY

1. Diverse Approaches: The literature survey reveals that a variety of machine learning techniques and approaches have been employed in predicting Parkinson's disease. Decision trees, support vector machines, deep learning, and data mining techniques are a few of these. Researchers have used a wide range of algorithms to improve accuracy.

2. Multimodal Data: Many studies in the literature survey utilized multiple data sources, including clinical records, medical imaging, genetic information, and voice recordings. This emphasizes how crucial a multimodal strategy is to capture comprehensive patient information for accurate predictions.

3. Early Diagnosis: The majority of research in the literature emphasizes the significance of early diagnosis. Machine learning models have the potential to detect the disease in its early stages, which can lead to more effective interventions and improved patient outcomes.

4. Data Privacy and Ethics: Several papers discuss the ethical considerations associated with using patient data for machine learning in healthcare. Ensuring data privacy, obtaining informed consent, and mitigating bias are vital concerns in this domain.

5. Personalized Care: The literature survey suggests that machine learning can be leveraged to provide personalized care for Parkinson's patients. Models can help tailor treatment plans to individual needs, recognizing that the disease's progression varies among patients.

6. Voice-Based Diagnostics: Voice recordings have gained attention as a non-invasive tool for Parkinson's disease prediction. Machine learning models applied to voice data show promise for early detection.

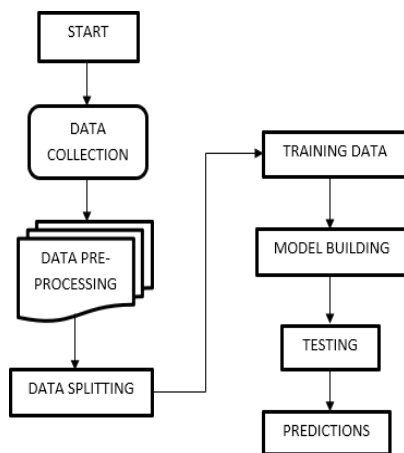
7. Deep Learning in Medical Imaging: Deep learning techniques, particularly in the analysis of medical

imaging data (e.g., MRI and PET scans), have been shown to improve accuracy in diagnosing Parkinson's disease. These models can extract subtle patterns that may not be apparent through traditional methods.

**8. The Importance of Expanding Datasets:** Many research pieces underline the constraints posed by limited datasets, emphasizing the critical requirement for more extensive and varied datasets. Such expansion is pivotal to improving the applicability and broadening the scope of machine learning techniques in predicting Parkinson's disease more accurately. This approach not only aims at enhancing predictive accuracy but also seeks to ensure that the insights derived are relevant across diverse patient populations, thereby offering a more robust foundation for tackling Parkinson's disease through predictive analytics.

**9. Continual Advancements:** The literature survey indicates that the field of Parkinson's disease prediction is continually evolving. Researchers are exploring new algorithms, feature selection methods, and data sources to improve the accuracy of predictive models.

These inferences illustrate the dynamic and evolving landscape of using machine learning techniques to predict Parkinson's disease. Researchers are addressing various challenges and exploring new avenues to enhance early diagnosis and personalized care for individuals with Parkinson's disease.



**Fig.1 Block Diagram of the PD Model**

#### IV. METHODOLOGY

##### a) Selected Methodology or Process Model

In this investigation, the primary approach involves assessing the performance and precision of various machine learning algorithms. These include Linear Discriminant Analysis (LDA), Random Forest, Decision Trees, Support Vector Machine (SVM), and Logistic Regression. This comparative analysis is central to identifying the most efficient algorithm for accurately predicting outcomes within the context of the study.

##### b) System Architecture

Machine learning enables computers to acquire knowledge autonomously, eliminating the need for

explicit coding. In this context, the author employs a suite of machine learning techniques, such as Logistic Regression, AdaBoost, Random Forest, Convolutional Neural Networks (CNN), Decision Trees, Support Vector Machines (SVM), and XGBoost. An architectural diagram illustrates the system's core components, their main functions, and provides a high-level summary. The diagram delineates a five-step sequence representing the execution pathway:

This sequence begins with leveraging the architectural layout to predict Parkinson's data and to refine unprocessed data. The initial phase involves preprocessing the collected raw data into an analyzable format. Subsequently, the dataset is segmented into training and testing portions. The model's precision is then evaluated by applying a machine learning technique to the Parkinson's dataset, which includes algorithms such as KNN, Decision Trees, AdaBoost, Random Forest, SVM, XGBoost, and Logistic Regression. It's essential to utilize the same algorithms for testing the data after its training phase. Ultimately, an assessment is conducted to compare the classification accuracy of these algorithms.

##### c) Algorithms for Implementation and Testing Plan of The Proposed Model / System

- KNN
- Support Vector Machine
- Random Forest
- Decision Tree
- Logistic Regression
- XG boost
- ADA Boosting

##### d) Project Management Plan

- Collect Data: Acquire the relevant dataset.
- Prepare Data:
- Import necessary libraries.
- Visualize the dataset for insights.
- Train Model: Use selected algorithms to train the dataset.
- Evaluate the model and determine the algorithms' accuracy.
- Predict the accuracy of all algorithms of Parkinson's using vocal dataset.

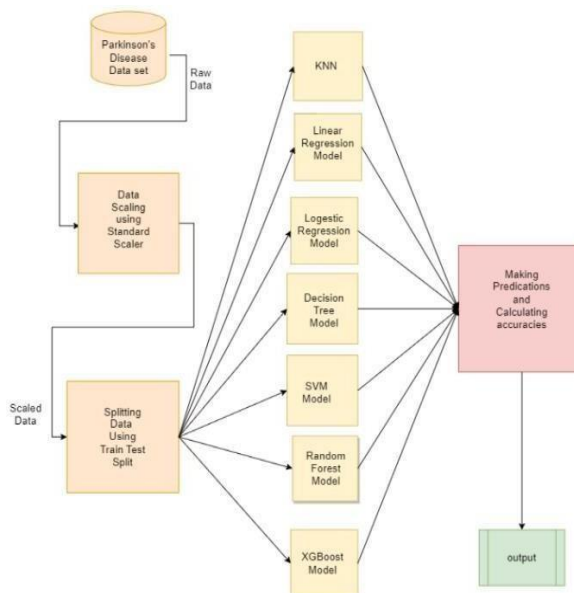
##### The project contains two parts:

i). Dataset collection.

ii). Train and test the model.

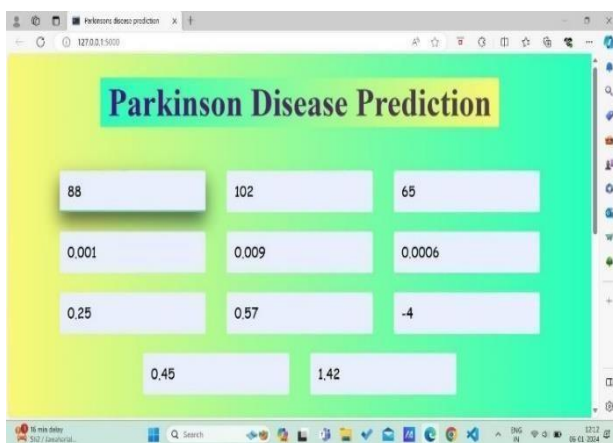
• **Dataset Collection-** Kaggle notebooks provided us with a collection of datasets. The dataset includes the associated disease's symptoms.

• **Model Training and Testing:** - To train the dataset, we utilized KNN, Logistic Regression, Decision Trees, and XG Boost. We evaluated the model after training and discovered its accuracy.

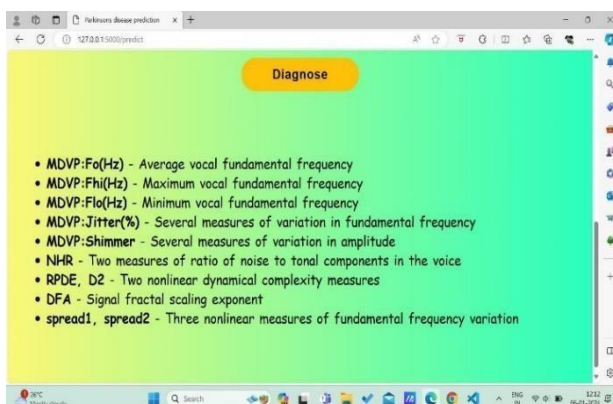


**Fig.2 System Architecture**

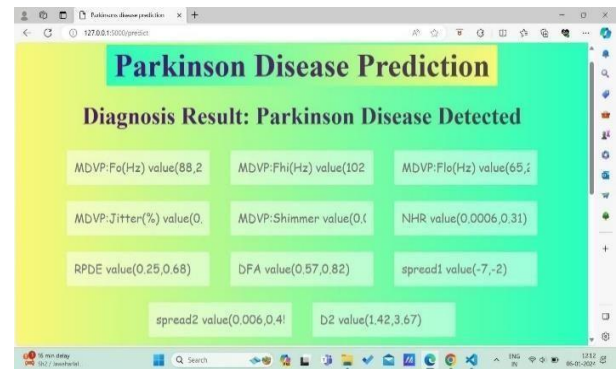
## V.RESULTS AND DISCUSSION



**Fig.3 passing data**



**Fig.4 attribute name**



**Fig.5 output**

## VI. CONCLUSION

Parkinson's disease, ranking as the second most prevalent neurodegenerative disorder for which current treatments are ineffective, demands innovative approaches for its prediction. To address this, several machine learning prediction models have been explored, including, but not limited to, Convolutional Neural Networks (CNN), Decision Trees, Random Forests, Support Vector Machines (SVM), XGBoost, AdaBoost, and Logistic Regression. These models are trained on datasets to ascertain the most accurate in reflecting the data's characteristics. A comparative analysis is conducted among different methodologies to evaluate their performance. Included in this analysis are XGBoost, K-Nearest Neighbors (KNN), AdaBoost, Random Forest, Decision Tree, SVM, and Logistic Regression. Notably, the KNN model emerged superior in terms of accuracy and the Area Under the Receiver Operating Characteristics Curve (AUC/ROC) score.

## REFERENCES: -

- [1] A. Ozcift, "SVM feature selection-based rotation forest ensemble classifiers to improve computer-aided diagnosis of Parkinson disease" *Journal of medical systems*, vol-36, no. 4, pp.2141-2147,2016.
- [2] Anila M Department of CS1, Dr G Pradeepini Department of CSE, "DIAGNOSIS OF PARKINSON'S DISEASE USING ARTIFICIAL NEURAL NETWORK", *JCR*, 7(19): 7260- 7269,2020.
- [3] Arvind Kumar Tiwari, "Machine Learning based Approaches for Prediction of Parkinson's Disease" *Machine Learning and Applications: An International Journal (MLAU)*vol. 3, June 2016.
- [4] Carlo Ricciardi, et al, "Using gait analysis' parameters to classify Parkinsonism: A data mining approach" *Computer Methods and Programs in Biomedicine* vol. 180, Oct. 2019.
- [5] Dr. Anupam Bhatia and Raunak Sulekh, "Predictive Model for Parkinson's Disease through Naive Bayes Classification" *International Journal of Computer Science & Communication* vol-9, Dec. 2017, pp. 194-

202, Sept 2017 - March 2018.

[6] Dr. R. Geetha Ramani, G. Sivagam Shomona Gracia Jacob. "Feature Relevance Analysis and to Classification of Parkinson's Disease Tele-Monitoring Data Through Data Mining" International Journal of Advanced Research in Computer Science and Software Engineering, vol-2, Issue3, March2015.

[7] Dragana Miljkovic et al, "Machine Learning and Data Mining Methods for Managing Parkinson's Disease" LNAI 9605, pp. 209-220, 2016.

[8] Farhad-Soleimani-Gharehehpoogh, Peyman-Mohammadi, "A Case Study of Parkinson's Disease Diagnosis Using Artificial Neural Networks" International Journal of Computer Applications, Vol-73, No.19, July 2017.

[9] Heisters. D, "Parkinson's: symptoms, treatments and research". British Journal of Nursing, 20(9), 548–554. doi:10.12968/bjon.2011.20.9.548, 2016.

[10] M. Abdar and M. Zomorodi-Moghadam, "Impact of Patients' Gender on Parkinson's disease using Classification Algorithms" Journal of AI and Data Mining, vol-6, 2018.

[11] M. A. E. Van Stiphout, J. Marinus, J. J. Van Hilten, F. Lobbezoo, and C. De Baat, "Oral health of Parkinson's disease patients: a case-control study" Parkinson's disease, vol- 2018, Article ID9315285,8pages,2018.

[12] Md. Redone Hassan, et al, "A Knowledge Base Data Mining based on Parkinson's Disease" International Conference on System Modelling & Advancement in Research Trends, 2019.

[13] Mandal, Indrajit, and N. Sairam. "New machine-learning algorithms for prediction of Parkinson's disease" International Journal of Systems Science 45.3: 647-666, 2014.

[14] Mohamad Alissa," Parkinson's Disease Diagnosis Using Deep Learning", August 2018.

[15] Peyman Mohammadi, Abdolreza Hatamlou and Mohammed Msdaris "A Comparative Study on Remote Tracking of Parkinson's Disease Progression Using Data Mining Methods" International Journal in Foundations of Computer Science and Technology (IJFCST), vol-3, No.6, Nov 2013.

[16] Satish Srinivasan, Michael Martin & Abhishek Tripathi, "ANN based Data Mining Analysis of Parkinson's Disease"International Journal of Computer Applications, vol. 168, June 2017.

[17] M. A. E. Van Stiphout, J. Marinus, J. J. Van Hilten, F. Lobbezoo, and C. De Baat, "Oral health of Parkinson's disease patients: a case-control study,"

Parkinson's Disease, vol. 2018, Article ID 9315285, 8 pages, 2018.

[18] Md. Redone Hassan et al, "A Knowledge Base Data Mining based on Parkinson's Disease" International Conference on System Modelling & Advancement in Research Trends, 2019.

[19] Mehrbakhsh Nilashi et al, "A hybrid intelligent system for the prediction of Parkinson's Disease progression using Machine Learning techniques" Biocybernetics and Biomedical Engineering 2017.

[20] R. P. Duncan, A. L. Leddy, J. T. Cavanaugh et al., "Detecting and predicting balance decline in Parkinson disease: a prospective cohort study," Journal of Parkinson's Disease, vol. 5, no. 1, pp. 131–139, 2015.

[21] Ramzi M. Sadek et al., "Parkinson's Disease Prediction using Artificial Neural Network" International Journal of Academic Health and Medical Research, vol. 3, Issue 1, January 2019.

[22] Rathod, Hemant Kumar , et al. "Automated attendance system using machine learning approach." 2017 International Conference on Nascent Technologies in Engineering (ICNTE). IEEE, 2017.

[23] Siswanto, Adrian Rhesa Septian, Anto Satriyo Nugroho, and Maulahikmah Galinium. "Implementation of face recognition algorithm for biometrics based time attendance system." 2014 International Conference on ICT For Smart Society (ICISS). IEEE, 2016.

[24] Wang Li Design and implementation of video surveillance system based on ARM and OpenCV [D]. Xi'an University of Electronic Science and Technology, 2019.

[25] Wang Yan, Wang Shuangyin. Research on face information enhancement recognition based on convolution neural network [J], Computer Science, 2018.