A

Project Report

on

ADAPTION OF BEST METHODOLOGY BY COMPARING DIIFFERENT ALGORTHM TO DETECT PARKINSON DISEASE

Submitted

in Partial Fulfilment of the Requirements for

The Degree of

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

VAIBHAV KESARWANI (1900540100180)

RISHABH KUMAR CHAUHAN (1900540100137)

SAURABH KUMAR SINGH (1900540100149)

SANJEEV RANJAN (1900540100147)

Under the supervision of

Neha Chauhan

(Assistant Professor)

Department of Computer Science and Engineering

BBDITM

BABU BANARASI DAS

INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(Recognized by AICTE, Govt. of India & Affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow)

AKTU College Code-054

May, 2023

CERTIFICATE

This is to certify that the project entitled **“Adaption of best methodology by comparing different Algorithm to Detect Parkinson Disease”** submitted by **“VAIBHAV KESARWANI (1900540100180) , RISHABH KUMAR CHAUHAN (1900540100137) , SAURABH KUMAR SINGH (1900540100149) and SANJEEV RANJAN (1900540100147)”** to Babu Banarasi Das Institute of Technology & Management, Lucknow, in partial fulfillment for the award of the degree of B.Tech in Computer Science and Engineering is a bonafide record of project work carried out by him/her under my/our supervision. The contents of this report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree.

Neha Chauhan	**Dr. Anurag Tiwari**
Assistant Professor	*Head of the Department*
Dept. of Computer	*Dept. of Computer*
Science and Engineering	*Science and Engineering*

Date:

Place:

DECLARATION

We declare that this project report titled **“Adaption of best methodology by comparing different Algorithm to Detect Parkinson Disease”** submitted in partial fulfillment of the degree of **B.Tech in Computer Science and Engineering** is a record of original work carried out by us under the supervision of **Neha Chauhan** and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

Date:

Signature:

Vaibhav Kesarwani
1900540100180
Rishabh Kumar Chauhan
1900540100137
Saurabh Kumar Singh
1900540100149
Sanjeev Ranjan
1900540100147

ACKNOWLEDGMENT

It gives us a great sense of pleasure to present the report of the B.Tech. Project undertaken during B.Tech. Final Year. We owe special debt of gratitude to Mrs. Neha Chauhan (Assistant Professor) and Dr. Anurag Tiwari (Head, Department of Computer Science and Engineering), Babu Banarasi Das Institute of Technology and Management, Lucknow for their constant support and guidance throughout the course of our work. Their sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only their cognizant efforts that our endeavors have seen light of the day. We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least; we acknowledge our family and friends for their contribution in the completion of the project.

ABSTRACT

Parkinson's disease is a progressive neurodegenerative disorder that greatly impacts the quality of life for many individuals. Primarily affecting motor functions, it gives rise to a cluster of symptoms known as "parkinsonism" or "parkinsonian syndrome." These symptoms manifest gradually over time and include shaking, rigidity, slowness of movement, and difficulties with walking. Furthermore, Parkinson's disease can also affect cognitive abilities, resulting in changes in thinking and behavior. It is not uncommon for individuals with Parkinson's to experience depression and anxiety alongside these symptoms.

In recent years, researchers have developed models that utilize voice analysis for the detection of Parkinson's disease. By examining voice patterns, these models can potentially identify markers indicative of the disease. Specifically, deflections or deviations in vocal characteristics have been explored as potential indicators of Parkinson's disease. Factors such as pitch, intensity, and speech rate are analyzed to detect any irregularities that may suggest the presence of the condition. However, it is important to note that while these models show promise, they should be used as supplementary tools alongside other clinical assessments for an accurate diagnosis.

This project showed 96% efficiency. In our model, a huge amount of data is collected from the normal person and also previously affected person by Parkinson’s disease. these data is trained using machine learning algorithms. From the whole data 60% is used for training and 40% is used for testing. The data of any person canbe entered in db to check whether the person is affected by Parkinson’s disease or not. There are 24 columns in the data set each column will indicate the symptom values of a patient except the status column. The status column has 0's and 1's those values will decide the person is effected with Parkinson’s disease. I's indicate personis effected, 0’s indicate normal conditions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS/ NOTATIONS/ NOMENCLATURE

The abbreviations used in this report are listed below:

M.L.	Machine Learning
UCI	University of California Irvine
KNN	K-Nearest Neighbors
DT	Decision Tree
RF	Random Forest
SVM	Support Vector Machine
UI	User Interface
CNN	Convolutional Neural Networks
CSV	Comma-Separated Values

CHAPTER 1

INTRODUCTION

After Alzheimer’s disease (AD), Parkinson’s disease (PD) is the world’s second most prevalentneurodegenerative disorder . It has been reported that PD prevails at a rate of 0.3% in of the entire population in industrialized countries, while in elder population (60 or above age), the PD prevalence rate is 1% . Impairments in voice have been reported to be the early biomarkersof the disease. Additionally, the proposed intelligent system has the capability to be used as an instrument for predromal diagnosis. Notably, patients with REM sleep behavior disorder (RBD) represent a good model as they develop PD with a high probability. It has been shown that slight speech and voice impairment may be a sensitive marker of preclinical PD .

The Parkinson treatment is likely very costly. This causes most of the patients cannot afford the cost of the Parkinson disease. Nowadays, Parkinson disease prediction is most critical matter for clinical practitioners to take accurate decision of such disease. It’s a great exercise at present time, machine learning based extensive platform can detect Parkinson disease. Medical data has grown a vast scale of volume from different clinical areas including health care services. To handle this data and attaining insights from this data there is a need for Big Data analysis through Machine learning that aims to solve a diverse medicinal and clinical issue. Officially, a significant number of investigations demonstrate that machine learning methods have picked up genuinely superior in classification based medical issues.People with PD face numerous symptoms including movement impairments (gait and tremors),poor balance, bradykinesia which is slowness of movement, and rigidity. As discussed above, the lack of reliable tests for diagnosis of PD has made the diagnosis of PD a challenging task . However, recent research reported that PD patients manifest impairments in voice and speech. However, these voice defects cannot be detected in clinics by medical practitioners. Hence, automated signal processing tools are required to capture these impairments in voice and to detect PD in its early stages. Recent research shows that machine learning and signal processing algorithms are successful in automated disease detection through automated risk factors extraction and classification. Motivated by these studies, in this paper, we also attempt to develop a method based on machine learning and signal processing algorithms for PD detection.

This project aims to develop an advanced ML model for the detection and classification of Parkinson's disease, leveraging the power of data analysis and pattern recognition. By utilizing a diverse dataset consisting of clinical records, neuroimaging data, and motor function assessments, we can train a robust ML model to accurately identify individuals with PD and distinguish them from healthy individuals. Machine learning algorithms have the ability to learn patterns and make predictions based on large amounts of data. In the context of PD, ML algorithms can analyze various features and biomarkers extracted from different modalities, such as voice recordings, gait analysis, and handwriting samples, to identify characteristic markers of the disease. By combining these diverse data sources, our ML model can provide a comprehensive and multi-modal approach to PD detection.

The potential benefits of this project are significant. Firstly, it can contribute to early detection and diagnosis of Parkinson's disease, enabling healthcare professionals to intervene at an earlier stage when treatments are most effective. Additionally, the ML model can assist in monitoring disease progression and evaluating the efficacy of treatment interventions over time. Furthermore, the development of a reliable ML model for PD detection can potentially reduce the reliance on subjective clinical assessments, which can vary among healthcare professionals. By leveraging objective and quantifiable measures, we aim to provide a more standardized and accurate approach to PD diagnosis. In conclusion, this project represents a novel and innovative approach to Parkinson's disease detection, leveraging the power of machine learning to analyze diverse data sources. By developing an advanced ML model, we aim to improve early diagnosis, enhance treatment strategies, and ultimately improve the quality of life for individuals living with Parkinson's disease.

1.1 Symptoms of Parkinson's Disease

Parkinson's disease is characterized by a range of symptoms, primarily related to motor function, that can be used to diagnose the condition. It is estimated that up to 80% of dopaminergic cells in the nigro-striatal pathway are destroyed before the cardinal motor signs of PD become evident. Diagnosis is made using criteria established by the UK PD Brain Bank. The earliest motor signs are slow voluntary movements, which progressively reduce in speed and amplitude (bradykinesia). In addition, one other symptom, such as muscle rigidity, resting tremor, or postural instability, is required for diagnosis. The diagnostic process involves ruling out symptoms that could indicate other conditions, such as Parkinsonian syndromes with their own unique neuropathological changes, and establishing supportive criteria for Parkinson's disease, such as the unilateral onset of symptoms, persistent asymmetry of clinical symptoms, good response to levodopa treatment, and induction of dyskinesias by dopaminergic treatment. The progression of PD symptoms typically begins on one side of the body and advances to the opposite side over time. Walking becomes increasingly difficult, with stooped posture, axial and limb rigidity (often with cogwheel phenomena), a shuffling gait, and reduced arm movement. Bradykinesia can cause an expressionless face (hypomimia) and reduced handwriting amplitudes (micrographia). Approximately 80% of people with PD experience limb tremors, most commonly described as a resting pill-rolling tremor of the hands. The thumb and index finger tend to come into contact and move in a circular motion, known as pill rolling. Tremors may also affect the legs, and other types of tremors can develop. Other gait disorders, such as blocking, hesitating, and gait festination, in which steps get smaller and faster over time, can also occur, potentially leading to loss of balance and falls. After several years of onset, between 25% and 60% of individuals report freezing of movements.

1.2 Treatment of Parkinson's Disease

The treatment of Parkinson's disease is focused on managing the symptoms, as there is no known cure for the disease. The underlying causes of Parkinson's are complex and not yet fully understood. Dopaminergic medications are typically used to treat the motor abnormalities associated with the disease. The most common initial medication is levodopa, a prodrug to dopamine, which often yields positive results. However, as the disease progresses and the capacity to store dopamine decreases, patients may experience a shorter duration of response to individual doses, alternate phases of good and poor response to medication, and involuntary movements of the head, trunk, or limbs, known as dyskinesias, among other motor complications. To address these fluctuations, other dopaminergic drugs such as monoamine oxidase type B inhibitors, catechol-O-methyltransferase inhibitors, amantadine, an NMDA receptor antagonist, and dopamine receptor agonists may be used. In cases where medicinal therapy is ineffective at managing symptoms, surgical therapy may be an option for a small percentage of patients. Deep brain electrical stimulation is one such option. However, it is important to note that surgical intervention is typically reserved for severe cases and comes with its own set of risks and potential complications. Overall, the management of Parkinson's disease requires ongoing evaluation and adjustments to treatment as symptoms evolve over time.

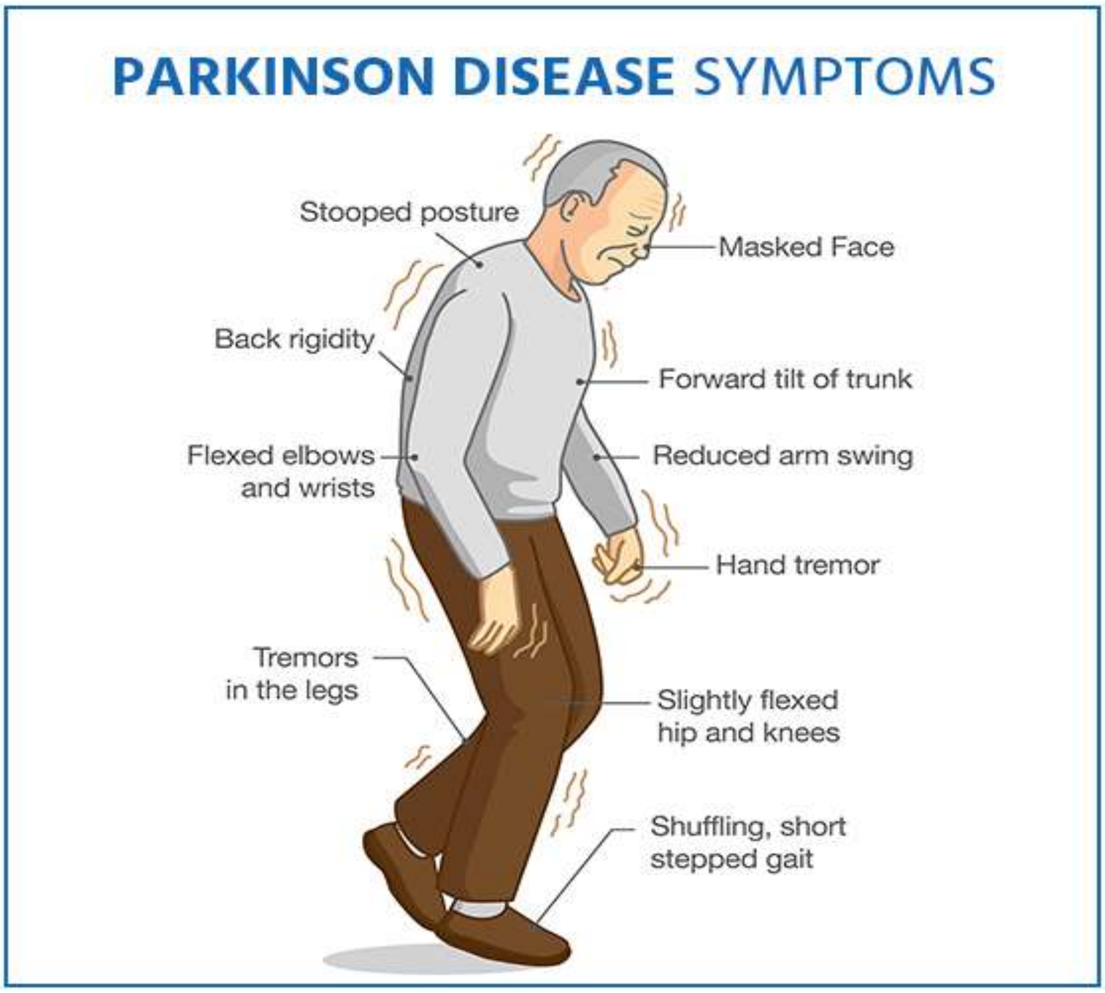


Fig1.2.1 Parkinson disease symptom

1.3 What is Machine Learning?

Machine Learning is a field within Artificial Intelligence (AI) that focuses on the ability of IT systems to autonomously solve problems by identifying patterns in databases. In essence, Machine Learning empowers IT systems to recognize patterns in existing algorithms and datasets, enabling them to develop appropriate solution concepts. This process involves generating artificial knowledge based on experience. However, the initial involvement of humans is crucial to enable software to independently generate solutions.

For instance, in Machine Learning, it is essential to input the necessary algorithms and data into the systems beforehand and define the corresponding analysis rules for pattern recognition in the dataset. Once these preparatory steps are completed, the system can perform various tasks through Machine Learning, such as finding, extracting, and summarizing relevant data, making predictions based on the analyzed data, and calculating probabilities for specific outcomes.

Algorithms play a pivotal role in Machine Learning as they are responsible for both pattern recognition and generating solutions. These algorithms can be categorized into different types.

i. Supervised learning:

In supervised learning, predefined example models are established to ensure appropriate knowledge allocation to specific algorithmic model groups. These models require specific specifications to facilitate effective learning. The system learns by observing given input and output pairs. During supervised learning, a programmer acts as a teacher and provides the correct values for specific inputs. The aim is to train the system through successive calculations using different inputs and outputs, enabling it to establish connections. Supervised learning involves using algorithms to learn the mapping function from input variables (X) to an output variable (Y), represented as Y = f(X). The objective is to approximate the mapping function accurately so that when a new input (X) is given, the algorithm can predict the corresponding output variables (Y) for that data. The term "supervised learning" is derived from the concept of the algorithm learning from a training dataset where the correct answers are known, similar to a teacher supervising the training process. The algorithm makes iterative predictions on the training data and receives corrections based on the known answers. The learning process continues until the algorithm achieves a satisfactory level of performance.Supervised machine learning techniques include linear regression, logistic regression, multi-class classification, decision trees, and support vector machines. Supervised learning problems can be further categorized into regression and classification problems. Regression deals with predicting a numerical dependent attribute, while classification involves predicting a categorical dependent attribute.

- **Regression:**
  Linear regression is a type of linear model that assumes a linear relationship between the input variables (x) and the output variable (y). In other words, it assumes that y can be calculated as a linear combination of the input variables (x). When there is only one input variable, it is referred to as simple linear regression. In this case, the goal is to find the best-fit line that represents the relationship between the input variable and the output variable.
  On the other hand, when there are multiple input variables, it is called multiple linear regression. In multiple linear regression, the aim is to determine the best-fit hyperplane that represents the relationship between the multiple input variables and the output variable. Linear regression is a widely used technique in statistics and machine learning for modeling and predicting continuous variables. It is based on the assumption that the relationship between the

variables can be described by a linear equation. By estimating the parameters of the linear model, such as the slope and intercept, the model can make predictions for new input values.

- **Classification:**

  Classification is a process that involves grouping a given dataset into distinct classes or categories. This process can be applied to both structured and unstructured data. The main objective of classification is to predict the class or category to which each data point belongs. These classes are commonly referred to as targets, labels, or categories. In essence, classification entails predicting the categorical class labels or assigning data to specific classes based on the training set and the values of classifying attributes. This learned information is then utilized to classify new, unseen data points.

  There are various classification models available for different scenarios. Some commonly used classification models include Logistic Regression, Decision Tree, Random Forest, Gradient Boosted Tree, One-vs-One, and Naïve Bayes. These models employ different algorithms and techniques to classify data accurately. In short, classification is a fundamental task that involves categorizing data into classes. It aims to predict class labels or assign data to specific categories based on learned patterns from the training set. Several classification models are available, each with its own strengths and suitable applications.

ii. Unsupervised learning:

Unsupervised learning is a branch of AI where the system learns without predefined target values or rewards. It is primarily used for tasks like data segmentation or clustering. In this type of learning, the machine aims to organize and categorize the input data based on certain inherent characteristics. For example, a machine could learn, in a simplistic manner, to sort coins of different colors based on the characteristic of "color."

Unsupervised machine learning algorithms are employed when the training data is neither classified nor labeled. The system does not have access to the correct outputs but instead explores the data and derives insights from unlabeled datasets to discover hidden structures. Unsupervised learning involves training machines using unclassified and unlabeled information, allowing the algorithm to operate on this data without guidance.

Unsupervised learning can be categorized into two main types of algorithms:

- **Clustering**: These algorithms aim to group similar data points together based on their characteristics or patterns. Common clustering algorithms include k-means clustering, hierarchical clustering, and density-based clustering.
- **Dimensionality reduction** :These algorithms focus on reducing the dimensionality of the input data while preserving relevant information. By reducing the number of variables or features, dimensionality reduction techniques simplify the data representation. Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE) are examples of dimensionality reduction algorithms.

1.4 Applications of Machine Learning:

1) Virtual Personal Assistants:

Siri, Alexa, and Google Now are among the well-known examples of virtual personal assistants. These intelligent assistants are designed to provide information and perform tasks based on voice commands. Machine learning plays a vital role in enhancing the capabilities of these personal

assistants by continuously collecting and refining knowledge through user interactions. This accumulated dataset is then utilized to deliver results and recommendations tailored specifically to individual preferences. These virtual personal assistants leverage machine learning techniques to understand and analyze user behavior, preferences, and patterns. By learning from your previous interactions, they adapt and improve their responses over time. The more you engage with these assistants, the better they become at understanding your needs and providing personalized assistance.
Through machine learning, virtual personal assistants are able to recognize speech patterns, interpret user queries, and retrieve relevant information from vast datasets. They can assist with tasks such as setting reminders, answering questions, playing music, providing weather updates, and much more. The underlying algorithms continuously learn and evolve, ensuring that the virtual personal assistants adapt to individual users and deliver a more personalized and seamless experience.

2) Image Recognition:

Image recognition is a widely utilized application of machine learning that involves the identification of objects, individuals, locations, and digital images, among others. A prevalent use case of image recognition is found in the automatic friend tagging suggestion feature, such as the one offered by Facebook.

When we upload a photo on Facebook that includes our friends, the platform employs machine learning algorithms for face detection and recognition. These algorithms analyze the uploaded image, detect faces within it, and then recognize those faces by matching them with known patterns and features. Based on this analysis, Facebook generates automated tagging suggestions that include the names of our friends. This technology relies on sophisticated machine learning models that have been trained on extensive datasets containing diverse facial images. By leveraging deep learning algorithms and neural networks, the system learns to accurately identify and differentiate between various individuals in photos.

3) Search Engine Result Refining:

Google and various other search engines leverage machine learning techniques to enhance the quality of search results provided to users. With each search query, behind the scenes, algorithms monitor and analyze how users interact with the search results. By observing user behavior, these algorithms make inferences about the relevance and usefulness of the displayed results.Speech Recognition: When users click on the top search results and spend a significant amount of time on the webpage, the algorithms interpret this as an indication that the displayed results were aligned with the user's query and met their needs. On the other hand, if users navigate to the second or third page of search results without clicking on any of the links, the algorithms infer that the presented results did not match the user's requirements.

4) **Product recommendations**:

Machine learning is widely used by various e-commerce and entertainment companies such as **Amazon**, **Netflix**, etc., for product recommendation to the user. Whenever we search for some product on Amazon, then we started getting an advertisement for the same product while internet surfing on the same browser and this is because of machine learning.
Google understands the user interest using various machine learning algorithms and suggests the product as per customer interest.
As similar, when we use Netflix, we find some recommendations for entertainment series, movies, etc., and this is also done with the help of machine learning.

5) Online Fraud Detection:

Machine learning is making our online transaction safe and secure by detecting fraud transaction. Whenever we perform some online transaction, there may be various ways that a fraudulent transaction can take place such as fake accounts, fake ids, and steal money in the middle of a transaction. So to detect this, Feed Forward Neural network helps us by checking whether it is a genuine transaction or a fraud transaction.

For each genuine transaction, the output is converted into some hash values, and these values become the input for the next round. For each genuine transaction, there is a specific pattern which gets change for the fraud transaction hence, it detects it and makes our online transactions more secure.

1.5 Motivation of the work:

A significant portion of individuals aged 65 years or older experience the unfortunate reality of living with a neurodegenerative disease, for which there is currently no cure. However, early detection of the disease holds the potential for effective management and control. Approximately 30% of patients are currently grappling with this incurable condition. Treatment options exist for those with milder symptoms, but if the disease goes undetected during its early stages, it can lead to devastating outcomes, including death.
The primary cause of Parkinson's disease can be attributed to the accumulation of misfolded protein molecules in neurons. Researchers have made progress in identifying the symptoms and the underlying causes that contribute to the development of this disease. While some symptoms have been addressed with available treatments, numerous symptoms still lack effective solutions. Given the increasing prevalence of Parkinson's disease in our society, it becomes crucial to develop predictive solutions that can identify it during its early stages. In this era of rising Parkinson's disease cases, finding a solution capable of detecting the disease in its early stages becomes imperative. Early detection would not only enable proactive management but could also provide opportunities for novel interventions and potential breakthroughs in treatment. By focusing on predicting and identifying the disease at its earliest manifestations, we can offer better care and improve the quality of life for individuals affected by Parkinson's disease.

CHAPTER 2

LITERATURE REVIEW

[1] "**Parkinson's Disease Prediction Using Machine Learning Approaches" by B.K. Varghese, G.B. Amali D*, and Uma Devi K.S. presented at the Fifth International Conference on Advanced Computing (ICoAC) in 2013.** This study aimed to explore the potential of machine learning techniques for predicting Parkinson's disease. The authors begin by highlighting the significance of early diagnosis and prediction of Parkinson's disease, as it can lead to better management and treatment outcomes for patients. They acknowledge that existing diagnostic methods, such as clinical assessments and neuroimaging, have limitations and may not always provide accurate predictions. The literature survey in this paper provides an overview of previous research in the field of Parkinson's disease prediction using machine learning. It encompasses various studies that have utilized different machine learning algorithms and input features to develop predictive models. The authors discuss several machine learning algorithms employed in previous studies, including decision trees, support vector machines (SVM), artificial neural networks (ANN), and logistic regression. They describe the advantages and limitations of each algorithm and highlight the need for feature selection and optimization techniques to enhance the performance of these models. Additionally, the authors summarise the different input features used in previous studies, such as demographic data, clinical symptoms, genetic information, and neuroimaging data. They emphasize the importance of feature engineering and selection to identify the most relevant features for accurate prediction. The literature survey also covers the evaluation metrics utilized in Parkinson's disease prediction studies, such as accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC). The authors discuss the significance of these metrics in assessing the performance of machine learning models and comparing results across different studies. Overall, the literature survey in this paper provides a comprehensive overview of the previous research conducted in the field of Parkinson's disease prediction using machine learning approaches. It highlights the diverse range of algorithms, input features, and evaluation metrics employed in these studies, contributing to the existing knowledge in the domain.

[2] "Parkinson's Disease Diagnosis Using Machine Learning and Voice" by Timothy J. Wroge, Yasin Ozkanca, C. Demiroglu, Dong Si, David C. Atkins, and R. Ghomi. The study explores the application of machine learning (ML) techniques for the diagnosis of Parkinson's disease (PD) using voice analysis. The paper was published in the IEEE Signal Processing in Medicine and Biology Symposium (SPMB) in 2018.The authors provide an introduction to Parkinson's disease and the significance of early diagnosis. They highlight the potential of voice analysis as a non-invasive and cost-effective method for PD detection.The paper discusses the extraction of various acoustic features from voice recordings, including pitch, jitter, shimmer, and spectral parameters. These features serve as important indicators of vocal abnormalities associated with Parkinson's disease.The study explores the utilization of ML algorithms, such as support vector machines (SVM), random forests (RF), and k-nearest neighbors (KNN), for PD diagnosis using voice data. The authors evaluate the performance of these algorithms in terms of accuracy, sensitivity, specificity, and area under the curve (AUC).The paper describes the dataset used for training and testing the ML models, consisting of voice recordings from individuals with and without Parkinson's disease. The experiments evaluate the efficacy of voice-based PD diagnosis using the ML algorithms mentioned above.The authors compare the performance of the ML models in accurately classifying individuals with and without PD based on voice features. They discuss the accuracy rates achieved and highlight the potential of voice analysis as a valuable tool in early PD detection.The paper concludes by emphasizing the potential of ML-based voice analysis for PD diagnosis. The authors discuss the limitations of the study and propose future research directions, including the integration of additional data sources and the exploration of deep learning techniques.

In conclusion, this literature survey presents a comprehensive overview of the paper "Parkinson's Disease Diagnosis Using Machine Learning and Voice." It highlights the significance of voice analysis in PD diagnosis and discusses the application of ML algorithms for this purpose. The findings contribute to the advancement of non-invasive and accessible methods for early detection of Parkinson's disease.

[3] "Parkinson's Disease Prediction Using Machine Learning Approaches" by Gokul.S. and Sivachitra.M. was presented at the Fifth International Conference on Advanced Computing (ICoAC) in 2013. This study aimed to explore the potential of machine learning techniques for predicting Parkinson's disease. The authors begin by emphasizing the importance of early detection and accurate prediction of Parkinson's disease, as they can significantly impact the management and treatment of patients. They highlight the limitations of traditional diagnostic methods and propose the utilization of machine learning algorithms as a promising alternative.

The literature survey in this paper provides a comprehensive review of previous research conducted in the field of Parkinson's disease prediction using machine learning approaches. The authors discuss various studies that have employed different machine learning algorithms and input features to develop prediction models.

Several machine learning algorithms are explored in the literature survey, including decision trees, support vector machines (SVM), artificial neural networks (ANN), and logistic regression. The authors discuss the strengths and weaknesses of each algorithm and emphasize the need for feature selection and optimization techniques to enhance the performance of these models.

The survey also covers the different types of input features utilized in previous studies, such as demographic data, clinical symptoms, genetic information, and neuroimaging data. The authors highlight the significance of feature engineering and selection to identify the most relevant features for accurate prediction.

Evaluation metrics used in Parkinson's disease prediction studies are also discussed, including accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC). The authors emphasize the importance of these metrics in assessing the performance of machine learning models and comparing results across different studies.

Furthermore, the survey highlights the challenges and future directions in the field. The authors mention the need for larger and more diverse datasets, standardized evaluation protocols, and the integration of multimodal data sources for improved prediction accuracy.

In conclusion, the literature survey in this paper provides a comprehensive overview of previous research on Parkinson's disease prediction using machine learning approaches. It covers various machine learning algorithms, input features, and evaluation metrics employed in these studies. The survey contributes to the existing knowledge in the field and identifies areas for future research, ultimately paving the way for improved prediction and early detection of Parkinson's disease.

[4] "Early Prediction of Parkinson's Disease Using Machine Learning and Deep Learning Approaches" by Harshvardhan Tiwari, Shiji K Shridhar, Preeti V Patil, K R Sinchana, and G Aishwarya, which explores the application of machine learning (ML) and deep learning (DL) techniques for the early prediction of PD.The authors present a comprehensive review of various ML and DL methodologies employed in PD prediction. They highlight the significance of feature selection and extraction techniques for improving the performance of predictive models. The paper discusses the utilization of diverse datasets, including clinical assessments, genetic information, and voice recordings, to train and evaluate prediction models.The study outlines several ML algorithms, such as support vector machines (SVM), random forests (RF), and logistic regression (LR), which have been successfully applied in PD prediction. Additionally, the authors explore the emerging field of DL, including convolutional neural networks (CNN) and recurrent neural networks (RNN), and their potential for capturing intricate patterns within PD data. The paper further emphasizes the importance of feature engineering and pre-processing techniques, such as principal component analysis (PCA) and wavelet transform, in enhancing the discriminatory power of predictive models. It also discusses the integration of multimodal data sources, such as clinical and imaging data, to improve the accuracy of PD prediction. Moreover, the authors address the challenges associated with PD prediction, including data imbalance, limited sample sizes, and the need for robust validation protocols. They propose potential solutions and future research directions to address these challenges effectively.

In conclusion, this literature survey highlights the significance of ML and DL approaches for the early prediction of Parkinson's disease. It provides a valuable overview of the existing techniques, datasets, and challenges in this field. The findings presented in this paper contribute to the advancement of PD prediction methodologies and pave the way for improved diagnosis and treatment strategies.

[5]"A Comparative Analysis of Parkinson's Disease Prediction Using Machine Learning Approaches" by F.M. Javed Mehedi Shamrat, Md. Asaduzzaman, A.K.M. Sazzadur Rahman, Raja Tariqul Hasan Tusher, and Zarrin Tasnim The study investigates and compares various machine learning (ML) techniques utilized for PD prediction. The authors present a comprehensive review of the existing literature on ML-based PD prediction models. They explore different ML algorithms such as support vector machines (SVM), random forests (RF), k-nearest neighbors (KNN), and decision trees (DT) that have been widely applied in PD prediction research. The paper highlights the significance of feature selection and extraction methods for enhancing the predictive performance of these models. Furthermore, the authors discuss the importance of utilizing diverse datasets comprising clinical, genetic, and neuroimaging data for training and evaluating the ML models. They also explore the integration of multimodal data to improve the accuracy and reliability of PD prediction. The study provides a comparative analysis of the performance of various ML algorithms in PD prediction tasks. It examines the strengths and limitations of each approach and identifies the most effective techniques based on evaluation metrics such as accuracy, sensitivity, specificity, and area under the curve (AUC). The authors discuss the potential of ensemble learning techniques and deep learning approaches for further enhancing the predictive accuracy of PD models.Moreover, the paper addresses the challenges associated with PD prediction, including data imbalance, small sample sizes, and the need for robust validation strategies. The authors propose future research directions and potential solutions to overcome these challenges and improve the effectiveness of ML-based PD prediction models.

In conclusion, this literature survey provides valuable insights into the comparative analysis of ML approaches for PD prediction. It highlights the importance of feature selection, the integration of multimodal data, and the potential of ensemble learning and deep learning techniques. The findings presented in this paper contribute to advancing the field of PD prediction and have implications for early diagnosis and intervention strategies.

[6]"Comparative Analysis of the Early Detection of Parkinson's Disease" by J. Olivia Capitola. The study provides a comprehensive comparative analysis of various approaches for the early detection of Parkinson's disease (PD). It emphasizes the significance of early biomarker identification, the integration of multimodal data, and the utilization of machine learning (ML) and deep learning (DL) techniques.The paper highlights the crucial role of early detection in Parkinson's disease management, emphasizing the benefits of timely intervention and improved treatment outcomes.The author presents a comparative analysis of various techniques for PD detection, including clinical assessments, genetic analysis, neuroimaging, ML algorithms (e.g., SVM, RF, ANN, DT), and DL approaches (e.g., CNN, RNN).The study emphasizes the significance of feature selection and extraction techniques in enhancing the performance of ML models and enabling accurate PD detection.The paper explores the integration of diverse data sources, such as clinical assessments, genetic information, and neuroimaging data, to improve the accuracy and reliability of PD detection models.The author discusses the potential of DL techniques, particularly CNN and RNN, in capturing complex patterns and correlations within PD data and facilitating early diagnosis.The paper addresses challenges associated with early PD detection, such as data variability and limited sample sizes, and proposes future research directions to overcome these challenges and enhance the effectiveness of detection models.

In conclusion, this literature survey provides valuable insights into the comparative analysis of approaches for the early detection of Parkinson's disease. It emphasizes the importance of early detection, feature selection, integration of multimodal data, and the potential of ML and DL techniques. The findings contribute to the advancement of PD detection methodologies, enabling timely interventions and improved management of the disease.

[7] "Support Vector Regression and its Mathematical Implementation" by Rahul Rastogi, published on Medium in 2020. The study provides a comprehensive overview of Support Vector Regression (SVR) and its mathematical implementation. The article begins by introducing SVR as a powerful machine learning algorithm for regression tasks. It highlights the significance of SVR in handling non-linear and high-dimensional data by employing a kernel function. Rastogi delves into the mathematical foundation of SVR, explaining the underlying concepts and equations involved. The survey covers essential components of SVR, including the optimization problem formulation, the definition of the loss function, and the introduction of slack variables to allow for deviations from the regression line. Furthermore, the article explores the mathematical implementation of SVR, focusing on the steps involved in solving the optimization problem using Lagrange multipliers and the kernel trick. It provides a detailed explanation of the dual form of the SVR problem and the corresponding mathematical equations. The survey also discusses the significance of tuning parameters in SVR, such as the regularization parameter and the kernel function selection. It highlights the importance of cross-validation techniques in determining optimal parameter values for SVR models. Rastogi emphasizes the advantages of SVR, such as its ability to handle small and noisy datasets and its robustness against outliers. The article also addresses some limitations of SVR, including its computational complexity for large-scale datasets.

In conclusion, this literature survey presents a comprehensive overview of SVR and its mathematical implementation. It highlights the key mathematical concepts, optimization problem formulation, and steps involved in solving SVR using Lagrange multipliers and the kernel trick. The findings contribute to a deeper understanding of SVR and its application in regression tasks.

[8] "Detection of Parkinson Disease Using Clinical Voice Data Mining" by TSaloni, R. K. Sharma, and A. K. Gupta, published in 2015. The study investigates the application of clinical voice data mining for the detection of Parkinson's disease (PD). The authors provide an introduction to Parkinson's disease and the significance of early detection for effective management. They emphasize the potential of clinical voice data mining as a non-invasive and cost-effective approach for PD diagnosis. The paper discusses the collection of clinical voice data from individuals with and without Parkinson's disease. It highlights the importance of appropriate data pre-processing techniques to remove noise and artifacts that could impact the accuracy of PD detection. The authors identify relevant acoustic features from the clinical voice recordings, such as fundamental frequency, formants, jitter, and shimmer. These features serve as key indicators of vocal abnormalities associated with Parkinson's disease. The study applies data mining techniques, including decision trees, artificial neural networks, and support vector machines, to analyse the extracted voice features and classify individuals as PD-positive or PD-negative. The authors evaluate the performance of these techniques in terms of accuracy, sensitivity, and specificity. The paper presents the results of the performance evaluation of the data mining techniques. It compares the accuracy rates achieved by different algorithms and discusses their effectiveness in accurately detecting Parkinson's disease based on clinical voice data. The authors conclude by highlighting the potential of clinical voice data mining for PD detection. They discuss the limitations of the study and propose future research directions, such as the integration of additional clinical data sources and the exploration of advanced machine learning algorithms.

In summary, this literature survey provides an overview of the paper "Detection of Parkinson's Disease Using Clinical Voice Data Mining." It emphasizes the significance of clinical voice data analysis for early PD detection and discusses the application of data mining techniques in this context. The findings contribute to the development of non-invasive and accessible methods for accurate diagnosis of Parkinson's disease.

[9]. The research paper is titled "**Intelligent Parkinson Disease Prediction Using Machine Learning Algorithms" by Tarigoppula. V.S. Sriram, M. Venkateswara Rao, G. V. Satya Narayana, DSVGK Kaladhar, and T. Pandu Ranga Vital, published in the International Journal of Engineering and Innovative Technology (IJEIT) in September 2013.**
The study focuses on the application of machine learning algorithms for intelligent prediction of Parkinson's disease (PD). The paper begins by highlighting the significance of early detection and accurate prediction of Parkinson's disease for effective management and treatment. It emphasizes the potential of machine learning techniques in analyzing diverse data sources to aid in PD prediction. Sriram et al. explore various machine learning algorithms employed for PD prediction, including support vector machines (SVM), artificial neural networks (ANN), decision trees (DT), and naive Bayes (NB). The paper discusses the strengths and limitations of each algorithm, highlighting their effectiveness in capturing patterns and features indicative of PD. Furthermore, the study discusses the selection and extraction of relevant features from the dataset to enhance the performance of the machine learning models. It emphasizes the importance of identifying key biomarkers and clinical factors associated with Parkinson's disease for accurate prediction. The authors describe the dataset used for training and testing the machine learning models, which includes clinical data from individuals diagnosed with and without Parkinson's disease. The experiments evaluate the performance of the different algorithms in terms of accuracy, sensitivity, specificity, and other relevant evaluation metrics. Moreover, the paper addresses the challenges associated with PD prediction, such as data variability, sample size limitations, and the need for robust validation techniques. It suggests potential solutions and future research directions to improve the effectiveness and reliability of machine learning-based PD prediction models.

In conclusion, this literature survey provides an overview of the paper "Intelligent Parkinson Disease Prediction Using Machine Learning Algorithms." It highlights the application of machine learning algorithms for intelligent prediction of Parkinson's disease. The findings contribute to the development of advanced prediction models that can aid in early detection and improved management of Parkinson's disease.

[10] "Pathoanatomy of Parkinson's Disease" by H. Braak, E. B. R. Braak, published in the Journal of Neurology in 2000. The study provides an in-depth analysis of the pathoanatomy, or the anatomical changes associated with Parkinson's disease (PD).The paper begins by introducing Parkinson's disease as a neurodegenerative disorder characterized by the progressive loss of dopaminergic neurons in the substantia nigra region of the brain. Braak highlights the importance of understanding the pathoanatomical changes underlying PD for improved diagnosis and treatment.Braak describes the stages of PD progression based on neuropathological examinations. The study introduces the concept of the "Braak staging system," which categorizes the disease into six stages, starting with the early involvement of the olfactory bulb and lower brainstem structures, followed by the ascending involvement of the substantia nigra and cortical regions.The author discusses the accumulation of Lewy bodies, abnormal protein aggregates, and the loss of dopaminergic neurons in different regions of the brain as the disease advances. Braak highlights the significance of these pathological changes in understanding the clinical manifestations and progression of PD. Furthermore, the paper explores the non-motor symptoms of PD, including cognitive impairments and autonomic dysfunctions, and their correlation with the underlying pathoanatomic. Braak discusses the involvement of various brain regions, such as the limbic system and cortical areas, in the manifestation of these non-motor symptoms. The study also addresses the potential mechanisms and etiology of PD, including genetic factors, environmental influences, and the interplay between neuronal vulnerability and protein aggregation.

In conclusion, this literature survey provides an overview of the paper "Pathoanatomy of Parkinson's Disease." It emphasizes the importance of understanding the anatomical changes associated with PD for improved diagnosis and treatment strategies. The findings contribute to a deeper understanding of the progressive nature of Parkinson's disease and its impact on various brain regions, shedding light on potential targets for therapeutic interventions.

[11] "An Analysis of Nonlinear and Fractal Measures in Vocal Frequencies for Developing Models to Predict Parkinson's Disease" by V. Matthews, published in 2016. The study investigates the application of nonlinear and fractal measures in vocal frequencies for developing predictive models for Parkinson's disease (PD).

The paper begins by highlighting the significance of early detection and accurate prediction of Parkinson's disease for effective management. It emphasizes the potential of vocal analysis as a non-invasive and cost-effective method for PD prediction.Matthews explores the use of nonlinear and fractal measures in analyzing vocal frequencies for detecting subtle changes associated with Parkinson's disease. The study discusses the relevance of these measures in capturing the complex dynamics and variability of the vocal signal, which can serve as potential markers for PD.

The author describes the dataset used for training and testing the predictive models, which consists of vocal recordings from individuals diagnosed with and without Parkinson's disease. The paper discusses the selection and extraction of relevant nonlinear and fractal measures from the vocal data, such as entropy, correlation dimension, and detrended fluctuation analysis.

Furthermore, the study evaluates the performance of different machine learning algorithms in predicting Parkinson's disease based on the extracted measures. It compares the accuracy rates, sensitivity, specificity, and other evaluation metrics achieved by the models, highlighting the potential of nonlinear and fractal measures for accurate PD prediction.

The paper also discusses the limitations and challenges associated with vocal analysis for PD prediction, including data variability, sample size limitations, and the need for robust validation techniques.

In conclusion, this literature survey provides an overview of the paper "An Analysis of Nonlinear and Fractal Measures in Vocal Frequencies for Developing Models to Predict Parkinson's Disease." It emphasizes the application of nonlinear and fractal measures in vocal analysis for developing predictive models for PD. The findings contribute to the advancement of non-invasive and accessible methods for early detection and accurate prediction of Parkinson's disease.

[12]"Epidemiology of Parkinson's Disease" by L. M. de Lau and B. M. M. Bloem, published in The Lancet in 2006 The study provides a comprehensive analysis of the epidemiology, or distribution and determinants, of Parkinson's disease (PD).

The authors introduce Parkinson's disease as a chronic neurodegenerative disorder characterized by motor symptoms such as tremors, bradykinesia, and postural instability. They highlight the global burden of PD and the importance of understanding its epidemiology for public health planning.The paper discusses the prevalence and incidence rates of Parkinson's disease across different populations and geographical regions. It explores variations in PD prevalence due to factors such as age, gender, and ethnicity. The authors highlight the increasing burden of PD in an aging population. The study examines various risk factors associated with Parkinson's disease, including genetic predisposition, environmental exposures, and lifestyle factors. The authors discuss the influence of these factors on the development and progression of PD.The paper explores the association of Parkinson's disease with comorbidities such as dementia, depression, and cardiovascular diseases. It also discusses the impact of PD on mortality rates and life expectancy. The authors address methodological challenges in studying the epidemiology of PD, including diagnostic criteria, case ascertainment, and data collection methods. They emphasize the importance of standardized approaches to ensure reliable and comparable epidemiological studies. The paper concludes by discussing future research directions in PD epidemiology, including the need for longitudinal studies, biomarker research, and investigations into modifiable risk factors. The authors highlight the potential of epidemiological data to inform preventive strategies and improve the management of Parkinson's disease.

In conclusion, this literature survey provides an overview of the paper "Epidemiology of Parkinson's Disease." It emphasizes the significance of understanding the distribution and determinants of PD for public health planning. The findings contribute to a deeper understanding of the prevalence, risk factors, comorbidities, and mortality associated with Parkinson's disease, paving the way for future research and interventions in this field.

CHAPTER 3

PROPOSED WORK

3.1 PROBLEM STATEMENT

Voice disorders can significantly impact individuals' lives, particularly professional voice users like singers, actors, and broadcasters. These disorders can arise from various causes, including physiological diseases, psychological disorders, accidents, vocal misuse, or surgery affecting the vocal folds. Assessment and treatment of voice disorders often involve acoustic tools that record changes in acoustic pressure at the lips or inside the vocal tract. These tools provide objective measures of voice function, complementing the subjective judgments made by clinicians Acoustic examination, although not the sole determinant of voice function, serves multiple practical purposes in clinical settings. These objective measurements can assist in evaluating the effectiveness of surgical procedures, therapy interventions, differential diagnosis, and screening processes. They can also enhance subjective voice quality assessments, such as the GRB (Grade, Roughness, and Breathiness) scale. By utilizing these objective measures, clinicians can construct "hoarseness" diagrams for clinical applications. Moreover, various techniques exist for automated screening of voice disorders based on these measures. In this study, classical measures such as jitter, shimmer, and HNR (Noise-to-Harmonics Ratio) are calculated for the purpose of comparison. These measures are commonly calculated using algorithms available in software packages like Praat. They rely on autocorrelation methods to determine the pitch period. Additionally, methods described in research papers by Michaelis and Tietze are employed. These methods involve calculating measures like EPQ (Energy Perturbation Quotient), PPQ (Pitch Perturbation Quotient), GNE (Glottal to Noise Excitation Ratio), and mean correlation coefficient between successive cycles. These measures are computed over overlapping frames of the speech signal and require the estimation of pitch period using waveform matching algorithms. By utilizing these acoustic measures and algorithms, clinicians and researchers can obtain objective insights into voice function, aiding in diagnosis, treatment planning, and monitoring progress. The combination of subjective and objective assessments enhances the overall understanding of voice disorders and facilitates effective interventions for patients.

The primary objective of this research is to develop an efficient prediction model that can greatly benefit patients suffering from Parkinson's disease by reducing the percentage of the disease and improving patient outcomes. Early identification of Parkinson's disease is crucial as proper treatment in the early stages can lead to better outcomes for patients. The main focus of this research is to identify the best machine learning technique for accurately distinguishing Parkinson's patients from healthy individuals. The study utilizes three popular techniques: K-nearest neighbors (KNN), Naïve Bayes, and Logistic Regression. These techniques are applied to a voice dataset of Parkinson's patients obtained from the UCI repository. To evaluate the performance of the prediction model, several evaluation metrics are employed, including the confusion matrix, precision, recall, accuracy, and F1-score. These metrics provide a comprehensive assessment of how well the model performs in correctly classifying individuals as Parkinson's patients or healthy individuals. Through this research, the aim is to develop a robust prediction model that can effectively differentiate between Parkinson's patients and healthy individuals. By identifying the best machine learning technique and employing feature selection, the research endeavors to enhance the accuracy of early Parkinson's detection, ultimately benefiting patients by enabling timely intervention and treatment.

3.2 WORK FLOW

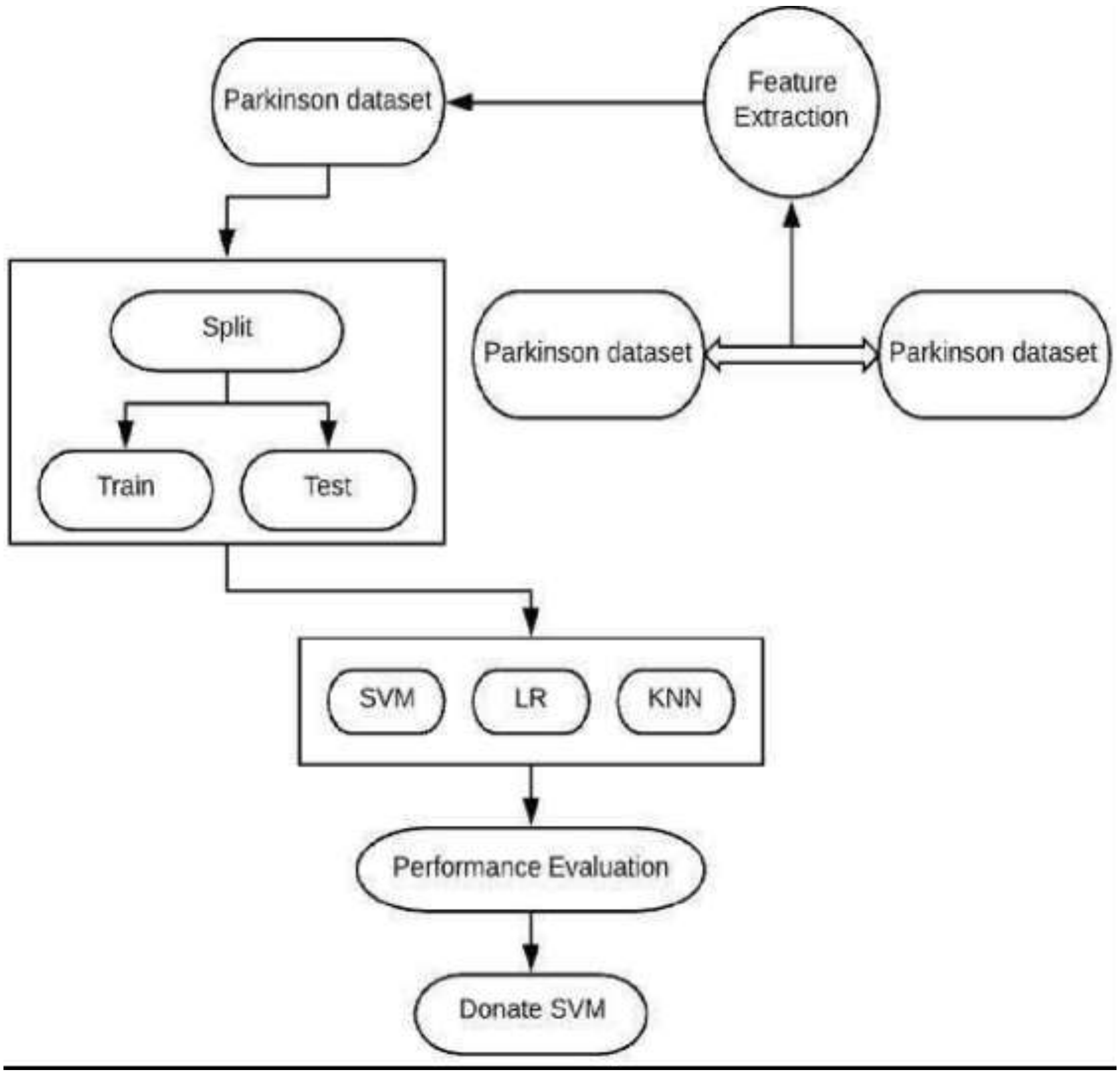


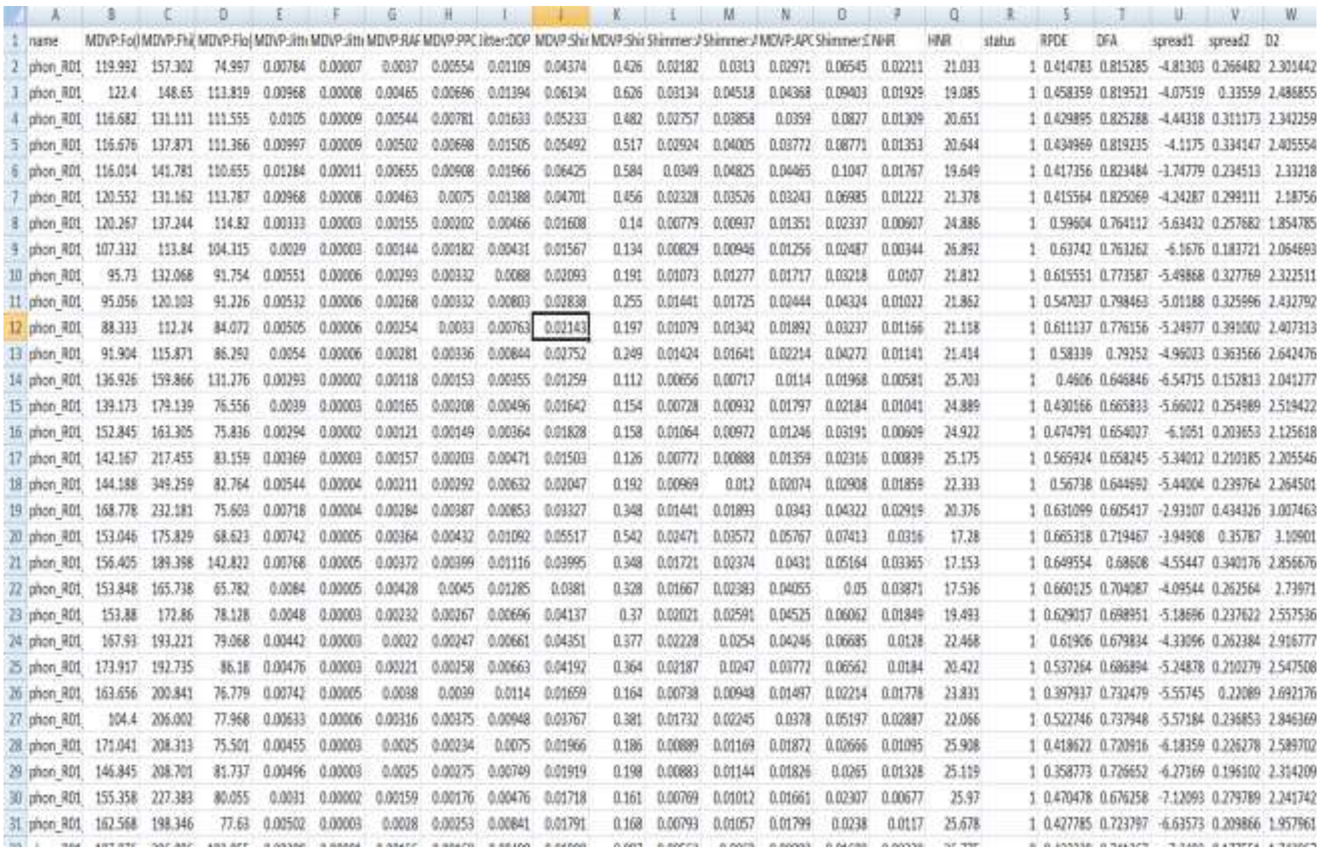
Fig.3.2.1 Work Flow of Proposed Model

3.3 Modules Division

Certainly! Let's discuss the various modules in the proposed system and their contributions to achieving the goal of accurately predicting Parkinson's disease.

3.3.1 Speech Dataset:

The primary objective of this phase is to identify and address any data-related challenges that may arise. During this step, the focus is on identifying the different data sources, as data can be obtained from a variety of sources, including files and databases. The efficiency of the system's output is greatly influenced by the quantity and quality of the collected data. Increased data availability leads to more accurate predictions. In our research, we have obtained the necessary data from the UCI website, which serves as a reliable and widely-used source for datasets. By accessing the UCI website, we ensure the availability of a diverse and representative dataset for our analysis. The significance of this step lies in recognizing potential data-related issues. These issues could include data inconsistencies, missing values, outliers, or biases that may affect the accuracy of the prediction models. By thoroughly identifying and addressing these problems, we ensure the reliability and integrity of the dataset. The data collected from various sources, particularly from the UCI website, contributes to the effectiveness of our research. The more extensive and comprehensive the dataset, the greater the potential for accurate predictions. By leveraging a diverse range of data sources, we enhance the robustness of our analysis and enable more precise identification of Parkinson's disease. Overall, this step in the research process is crucial in identifying data-related problems, selecting appropriate data sources, and ensuring the quantity and quality of the collected data. These efforts form the foundation for subsequent stages of the research, ultimately leading to more accurate predictions and improved outcomes for individuals with Parkinson's disease.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	name	MDVP:Fo(	MDVP:Fhi(	MDVP:Flo(	MDVP:Jitt	MDVP:Jitt	MDVP:RAF	MDVP:PPC	Jitter:DDP	MDVP:Shir	MDVP:Shir	Shimmer:A	Shimmer:A	MDVP:APC	Shimmer:D	NHR	HNR	status	RPDE	DFA	spread1	spread2	D2
2	phon_R01	119.992	157.302	74.997	0.00784	0.00007	0.0037	0.00554	0.01109	0.04374	0.426	0.02182	0.0313	0.02971	0.06545	0.02211	21.033	1	0.414783	0.815285	-4.81303	0.266482	2.301442
3	phon_R01	122.4	148.65	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	0.626	0.03134	0.04518	0.04368	0.09403	0.01929	19.085	1	0.458359	0.819521	-4.07519	0.33559	2.486855
4	phon_R01	116.682	131.111	111.555	0.0105	0.00009	0.00544	0.00781	0.01633	0.05233	0.482	0.02757	0.03858	0.0359	0.0827	0.01309	20.651	1	0.429895	0.825288	-4.44318	0.311173	2.342259
5	phon_R01	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	0.517	0.02924	0.04005	0.03772	0.08771	0.01353	20.644	1	0.434969	0.819235	-4.1175	0.334147	2.405554
6	phon_R01	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	0.584	0.0349	0.04825	0.04465	0.1047	0.01767	19.649	1	0.417356	0.823484	-3.74779	0.234513	2.33218
7	phon_R01	120.552	131.162	113.787	0.00968	0.00008	0.00463	0.0075	0.01388	0.04701	0.456	0.02328	0.03526	0.03243	0.06985	0.01222	21.378	1	0.415564	0.825069	-4.24287	0.299111	2.18756
8	phon_R01	120.267	137.244	114.82	0.00333	0.00003	0.00155	0.00202	0.00466	0.01608	0.14	0.00779	0.00937	0.01351	0.02337	0.00607	24.886	1	0.59604	0.764112	-5.63432	0.257682	1.854785
9	phon_R01	107.332	113.84	104.315	0.0029	0.00003	0.00144	0.00182	0.00431	0.01567	0.134	0.00829	0.00946	0.01256	0.02487	0.00344	26.892	1	0.63742	0.763262	-6.1676	0.183721	2.064693
10	phon_R01	95.73	132.068	91.754	0.00551	0.00006	0.00293	0.00332	0.0088	0.02093	0.191	0.01073	0.01277	0.01717	0.03218	0.0107	21.812	1	0.615551	0.773587	-5.49868	0.327769	2.322511
11	phon_R01	95.056	120.103	91.226	0.00532	0.00006	0.00268	0.00332	0.00803	0.02838	0.255	0.01441	0.01725	0.02444	0.04324	0.01022	21.862	1	0.547037	0.798463	-5.01188	0.325996	2.432792
12	phon_R01	88.333	112.24	84.072	0.00505	0.00006	0.00254	0.0033	0.00763	0.02143	0.197	0.01079	0.01342	0.01892	0.03237	0.01166	21.118	1	0.611137	0.776156	-5.24977	0.391002	2.407313
13	phon_R01	91.904	115.871	86.292	0.0054	0.00006	0.00281	0.00336	0.00844	0.02752	0.249	0.01424	0.01641	0.02214	0.04272	0.01141	21.414	1	0.58339	0.79252	-4.96023	0.363566	2.642476
14	phon_R01	136.926	159.866	131.276	0.00293	0.00002	0.00118	0.00153	0.00355	0.01259	0.112	0.00656	0.00717	0.0114	0.01968	0.00581	25.703	1	0.4606	0.646846	-6.54715	0.152813	2.041277
15	phon_R01	139.173	179.139	76.556	0.0039	0.00003	0.00165	0.00208	0.00496	0.01642	0.154	0.00728	0.00932	0.01797	0.02184	0.01041	24.889	1	0.430166	0.665833	-5.66022	0.254989	2.519422
16	phon_R01	152.845	163.305	75.836	0.00294	0.00002	0.00121	0.00149	0.00364	0.01828	0.158	0.01064	0.00972	0.01246	0.03191	0.00609	24.922	1	0.474791	0.654027	-6.1051	0.203653	2.125618
17	phon_R01	142.167	217.455	83.159	0.00369	0.00003	0.00157	0.00203	0.00471	0.01503	0.126	0.00772	0.00888	0.01359	0.02316	0.00839	25.175	1	0.565924	0.658245	-5.34012	0.210185	2.205546
18	phon_R01	144.188	349.259	82.764	0.00544	0.00004	0.00211	0.00292	0.00632	0.02047	0.192	0.00969	0.012	0.02074	0.02908	0.01859	22.333	1	0.56738	0.644692	-5.44004	0.239764	2.264501
19	phon_R01	168.778	232.181	75.603	0.00718	0.00004	0.00284	0.00387	0.00853	0.03327	0.348	0.01441	0.01893	0.0343	0.04322	0.02919	20.376	1	0.631099	0.605417	-2.93107	0.434326	3.007463
20	phon_R01	153.046	175.829	68.623	0.00742	0.00005	0.00364	0.00432	0.01092	0.05517	0.542	0.02471	0.03572	0.05767	0.07413	0.0316	17.28	1	0.665318	0.719467	-3.94908	0.35787	3.10901
21	phon_R01	156.405	189.398	142.822	0.00768	0.00005	0.00372	0.00399	0.01116	0.03995	0.348	0.01721	0.02374	0.0431	0.05164	0.03365	17.153	1	0.649554	0.68608	-4.55447	0.340176	2.856676
22	phon_R01	153.848	165.738	65.782	0.0084	0.00005	0.00428	0.0045	0.01285	0.0381	0.328	0.01667	0.02383	0.04055	0.05	0.03871	17.536	1	0.660125	0.704087	-4.09544	0.262564	2.73971
23	phon_R01	153.88	172.86	78.128	0.0048	0.00003	0.00232	0.00267	0.00696	0.04137	0.37	0.02021	0.02591	0.04525	0.06062	0.01849	19.493	1	0.629017	0.698951	-5.18696	0.237622	2.557536
24	phon_R01	167.93	193.221	79.068	0.00442	0.00003	0.0022	0.00247	0.00661	0.04351	0.377	0.02228	0.0254	0.04246	0.06685	0.0128	22.468	1	0.61906	0.679834	-4.33096	0.262384	2.916777
25	phon_R01	173.917	192.735	86.18	0.00476	0.00003	0.00221	0.00258	0.00663	0.04192	0.364	0.02187	0.0247	0.03772	0.06562	0.0184	20.422	1	0.537264	0.686894	-5.24878	0.210279	2.547508
26	phon_R01	163.656	200.841	76.779	0.00742	0.00005	0.0038	0.0039	0.0114	0.01659	0.164	0.00738	0.00948	0.01497	0.02214	0.01778	23.831	1	0.397937	0.732479	-5.55745	0.22089	2.692176
27	phon_R01	104.4	206.002	77.968	0.00633	0.00006	0.00316	0.00375	0.00948	0.03767	0.381	0.01732	0.02245	0.0378	0.05197	0.02887	22.066	1	0.522746	0.737948	-5.57184	0.236853	2.846369
28	phon_R01	171.041	208.313	75.501	0.00455	0.00003	0.0025	0.00234	0.0075	0.01966	0.186	0.00889	0.01169	0.01872	0.02666	0.01095	25.908	1	0.418622	0.720916	-6.18359	0.226278	2.589702
29	phon_R01	146.845	208.701	81.737	0.00496	0.00003	0.0025	0.00275	0.00749	0.01919	0.198	0.00883	0.01144	0.01826	0.0265	0.01328	25.119	1	0.358773	0.726652	-6.27169	0.196102	2.314209
30	phon_R01	155.358	227.383	80.055	0.0031	0.00002	0.00159	0.00176	0.00476	0.01718	0.161	0.00769	0.01012	0.01661	0.02307	0.00677	25.97	1	0.470478	0.676258	-7.12093	0.279789	2.241742
31	phon_R01	162.568	198.346	77.63	0.00502	0.00003	0.0028	0.00253	0.00841	0.01791	0.168	0.00793	0.01057	0.01799	0.0238	0.0117	25.678	1	0.427785	0.723797	-6.63573	0.209866	1.957961

Fig-3.3.1.1 Sample of acquired speech dataset from UCI

In the above Fig-3.2, we can see the speech dataset that has collected from UCI website.The features are listed below:

1. **Freatures of dataset**

There are 24 columns also known as attributes in the dataset. The description of all columns are given below:

- **Name -** Subject name and recording number.
- **MDVP:Fo(Hz)** – Average vocal fundamental frequency.

- **MDVP:Fhi(Hz)** – Maximum vocal fundamental frequency.
- **MDVP:Flo(Hz)** – Minimum vocal fundamental frequency.
- **MDVP:Jitter(%) , MDVP:Jitter(Abs), MDVP:RAP, MDVP:PPQ, Jitter:DDP** – Several measures of variation in fundamental frequency.
- **MDVP:Shimmer, MDVP:Shimmer(dB), Shimmer:APQ3, Shimmer:APQ5, MDVP:APQ, Shimmer:DDA** – Several measures of variation in amplitude.
- **NHR, HNR** – Two measures of ratio of noise to tonal components in the voice.
- **RPDE, D2** – Two nonlinear dynamical complexity measures.
- **DFA** – Signal fractal scaling exponent.
- **Spread1, Spread2, PPE** – Three nonlinear measures of fundamental frequency variation.
- **Status -** Health status of the subject, (one) - Parkinson's, (zero) – Healthy.

2. Measures Explanation

a.) Fundamental Frequency

The lowest frequency of a sound wave is a Fundamental frequency. (Matthews, 2016) Our brain interprets sound waves as a single pitch when we speak and listen to each other, even though they actually contain various frequencies (source). For this dataset, the lowest frequency of the subject's voice is captured. (Matthews, 2016)

b.) Ratio to Noise and Tonal Components

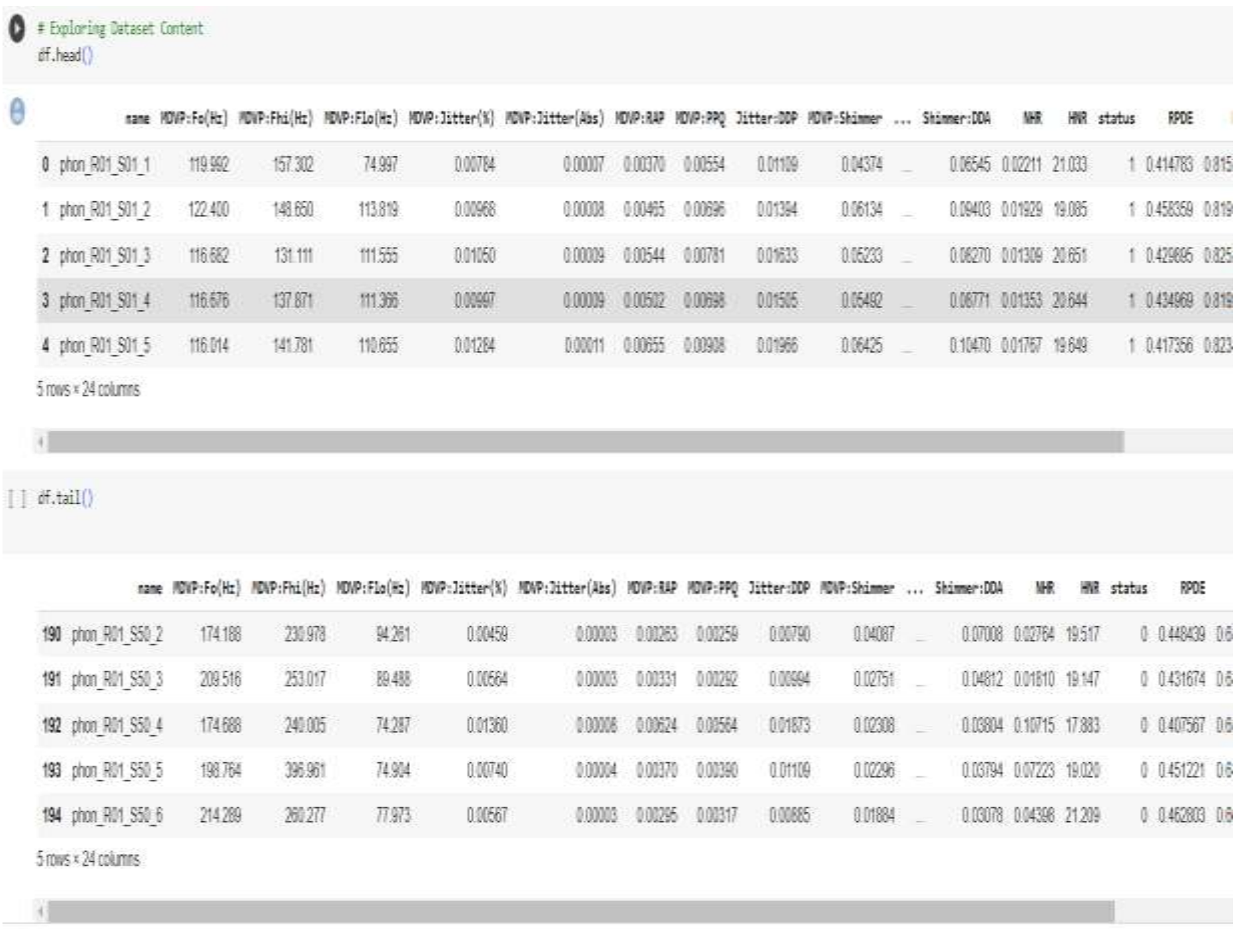
The tone refers to one frequency and a noise refers to multiple frequencies. (Matthews, 2016)

c.) Dynamical Complexity Measure

Entropy measure in an individual's voice. A low score means less entropy. (Matthews, 2016)

d.) Signal Fractal Scaling Component

The measure looks at the pattern of a time-ordered series of real numbers that repeats itself. A low score indicates that a person's voice has more entropy. (Matthews, 2016)



```python
# Exploring Dataset Content
df.head()
```

	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	...	Shimmer:DDA	NHR	HNR	status	RPDE	
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109	0.04374	...	0.06545	0.02211	21.033	1	0.414783	0.815
1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	...	0.09403	0.01929	19.085	1	0.458359	0.819
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633	0.05233	...	0.08270	0.01309	20.651	1	0.429895	0.825
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	...	0.08771	0.01353	20.644	1	0.434969	0.819
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	...	0.10470	0.01767	19.649	1	0.417356	0.823

5 rows × 24 columns

```python
df.tail()
```

	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	...	Shimmer:DDA	NHR	HNR	status	RPDE	
190	phon_R01_S50_2	174.188	230.978	94.261	0.00459	0.00003	0.00263	0.00259	0.00790	0.04087	...	0.07008	0.02764	19.517	0	0.448439	0.6
191	phon_R01_S50_3	209.516	253.017	89.488	0.00564	0.00003	0.00331	0.00292	0.00994	0.02751	...	0.04812	0.01810	19.147	0	0.431674	0.6
192	phon_R01_S50_4	174.688	240.005	74.287	0.01360	0.00008	0.00624	0.00564	0.01873	0.02308	...	0.03804	0.10715	17.883	0	0.407567	0.6
193	phon_R01_S50_5	198.764	396.961	74.904	0.00740	0.00004	0.00370	0.00390	0.01109	0.02296	...	0.03794	0.07223	19.020	0	0.451221	0.6
194	phon_R01_S50_6	214.289	260.277	77.973	0.00567	0.00003	0.00295	0.00317	0.00885	0.01884	...	0.03078	0.04398	21.209	0	0.462803	0.6

5 rows × 24 columns

Fig-3.3.1.2 Reading the dataset from the CSV file into notebook

The data from the CSV file will be read into the Google colab, also known as a Python notebook, as the following step. In our project, Python notebook is utilised for model comparison, feature selection, and data pre-processing. We've demonstrated in Fig. 3.3 how to read data from CSV files using built-in Python methods from the Pandas package.

3.3.2 Data Pre-Processing:

The key objective of this stage is to thoroughly assess and comprehend the characteristics of the collected data from the previous step, as well as to evaluate its overall quality. Real-world data often exhibits various imperfections such as noise, missing values, and undesirable formats, making it unsuitable for direct utilization in machine learning models. Hence, data preprocessing becomes crucial to cleanse and transform the data, ensuring its compatibility with the subsequent machine learning stages and enhancing the accuracy and effectiveness of the models. Furthermore, this stage involves the identification and elimination of any duplicate entries within the dataset. In order to conduct a comprehensive analysis, I utilized the ".shape" function, revealing that the dataset consists of 195 rows and 24 columns. This information provides valuable insights into the dataset's size and dimensions. Additionally, I employed the '.info' method to validate the data types associated with each column, a vital step for subsequent data processing operations. Furthermore, I diligently examined the presence of null values, which indicate missing data, utilizing the .isnull() function to identify any occurrences of such missing values. By diligently performing these steps, we ensure that the data is prepared and refined, paving the way for more accurate and reliable machine learning models and insights derived from the dataset.

3.3.3 Data Visualization

I represented the number of rows (or individuals) affected by Parkinson's disease in this segment. I discovered that 48 row values (person) indicate health, but 147 row values (person) imply Parkinson illness. In order to explain this finding more well, I also created a pie graph, which can be seen in Fig. 3.3. After that, I created a heat map (fig. 3.3) showing the correlation between all characteristics and columns in the dataset. Numerous independent characteristics are highly associated with one another, as can be seen in the correlation heatmap. We may eliminate any unnecessary duplicate characteristics from our dataset by using this heat map to understand the relationships between the different elements of the dataset.

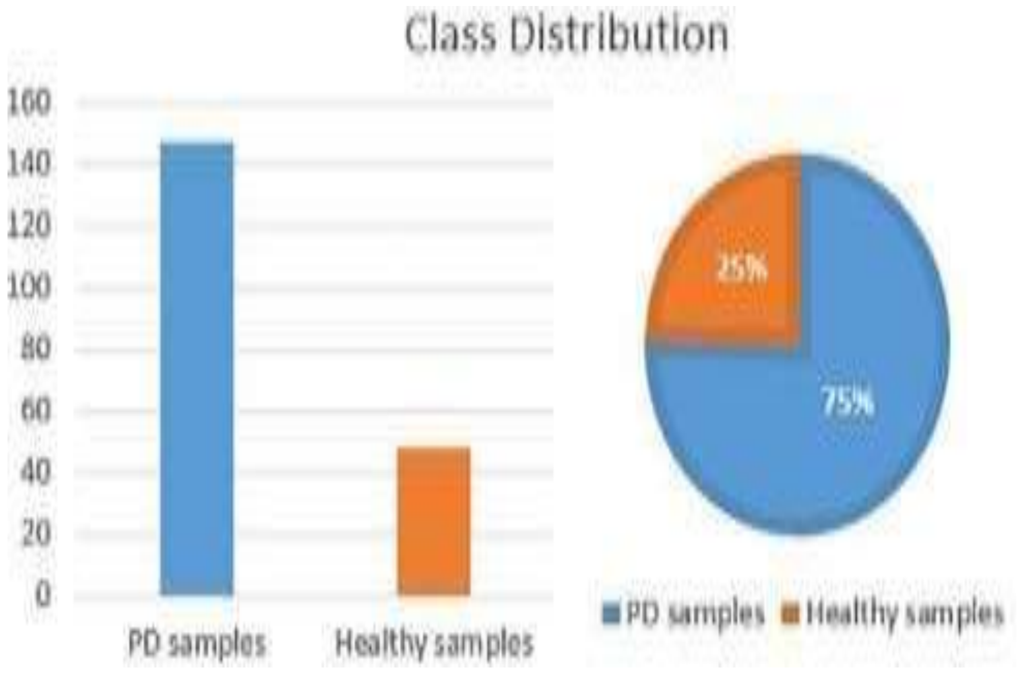


Fig-3.3.3.1 Recorded phonetics class distribution in the dataset of Parkinson disease

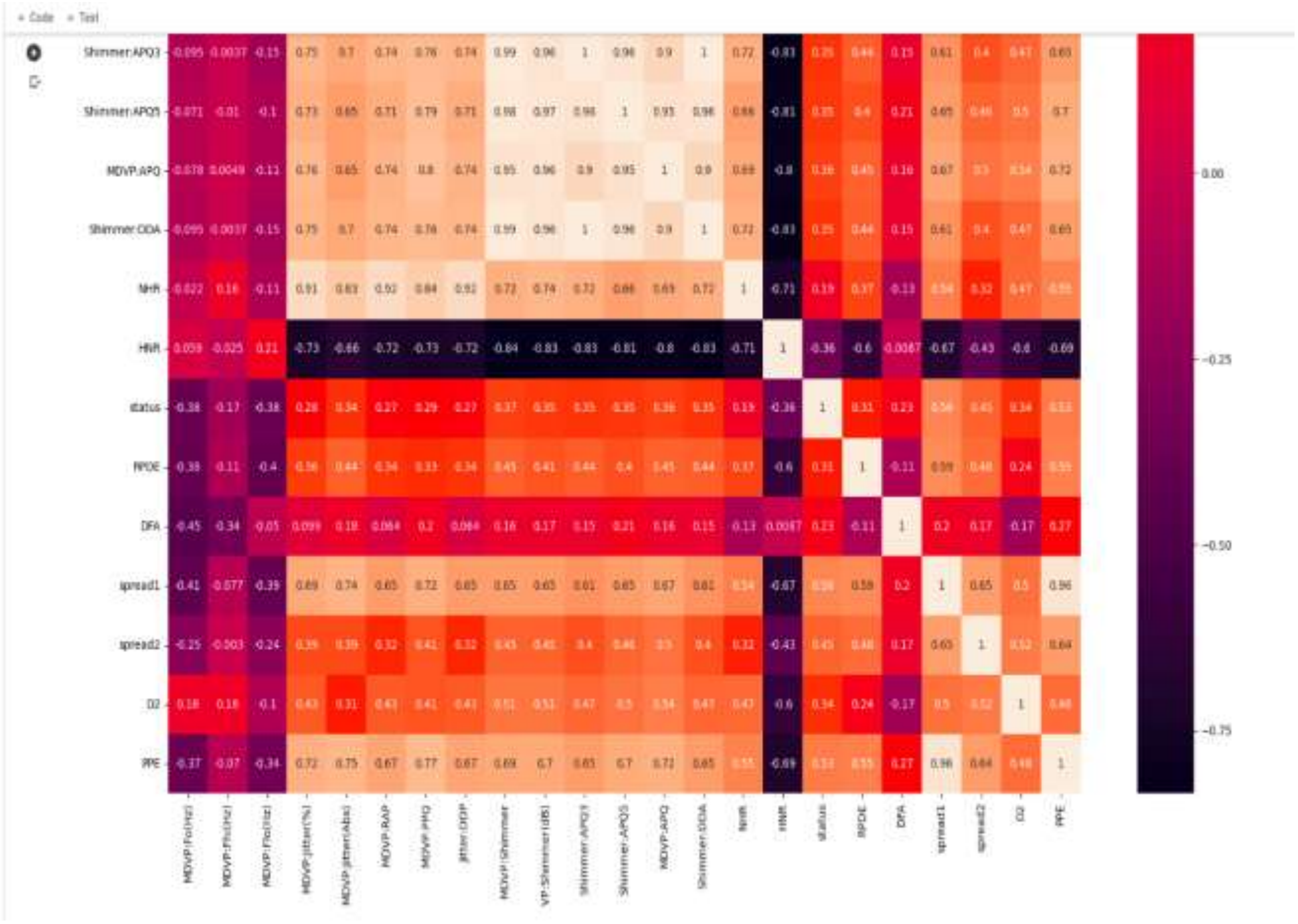


Fig-3.3.3.2 Correlation Matrix

3.3.4 Balancing Dataset

Balancing a dataset refers to the process of equalizing the distribution of different classes or categories within the dataset. This is often done when dealing with imbalanced datasets, where one class is significantly more prevalent than the others. Imbalanced datasets can lead to biased models that perform poorly on the underrepresented classes.

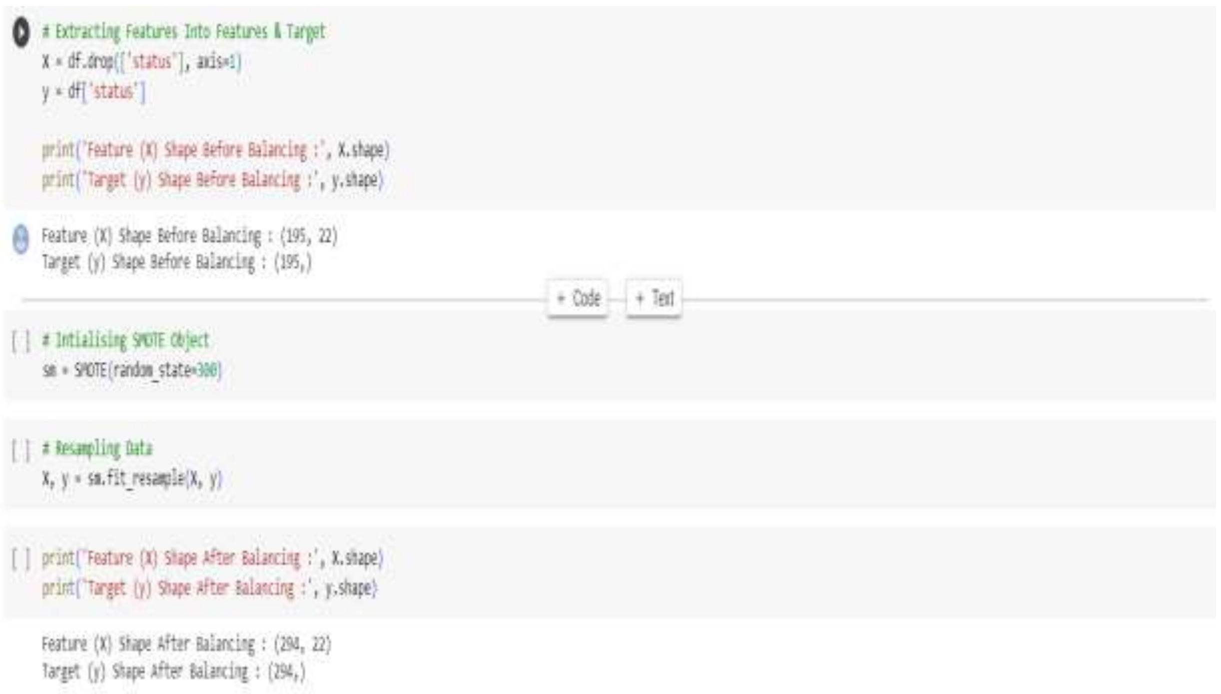
There are several techniques you can use to balance a dataset:

1. **Random under sampling**: This involves randomly removing samples from the majority class until its distribution matches that of the minority class. This technique can be effective but runs the risk of losing important information by discarding data.
2. **Random oversampling:** This technique involves randomly duplicating samples from the minority class until its distribution matches that of the majority class. However, this approach may lead to overfitting and the generation of synthetic data points that are too similar to existing samples.
3. **Synthetic minority oversampling technique (SMOTE):** SMOTE generates synthetic samples for the minority class by interpolating between neighboring instances. It creates new samples by taking a random instance from the minority class, identifying its nearest neighbors, and creating new samples along the line connecting the instance and its neighbors.
4. **Class-weighting:** Instead of modifying the dataset, you can assign different weights to each class during model training. By giving higher weights to the minority class, you can make the model pay more attention to its examples. Most machine learning algorithms and

frameworks provide an option to incorporate class weights.

5. **Collect more data :** If possible, collecting more data for the underrepresented class can help balance the dataset naturally. However, this approach may not always be feasible.

It's important to note that the choice of balancing technique depends on the specific characteristics of your dataset and the problem you're trying to solve. Experimentation and evaluation of different methods are crucial to determine which approach works best for your particular scenario. It is observed that the Dataset Is Heavily Imbalanced, with Number of Samples of Parkinson Disease Samples being 147, and Non-Parkinson Being only 48. Hence, in this section, we make use of SMOTE to Oversample and Balance the dataset. It can be seen in the Fig-3.5 before and after the balancing the dataset result.



```
# Extracting Features Into Features & Target
X = df.drop(['status'], axis=1)
y = df['status']

print('Feature (X) Shape Before Balancing :', X.shape)
print('Target (y) Shape Before Balancing :', y.shape)
```

```
Feature (X) Shape Before Balancing : (195, 22)
Target (y) Shape Before Balancing : (195,)
```

```
# Intialising SMOTE Object
sm = SMOTE(random_state=300)
```

```
# Resampling Data
X, y = sm.fit_resample(X, y)
```

```
print('Feature (X) Shape After Balancing :', X.shape)
print('Target (y) Shape After Balancing :', y.shape)
```

```
Feature (X) Shape After Balancing : (294, 22)
Target (y) Shape After Balancing : (294,)
```

Fig-3.3.4.1 Balancing the dataset.

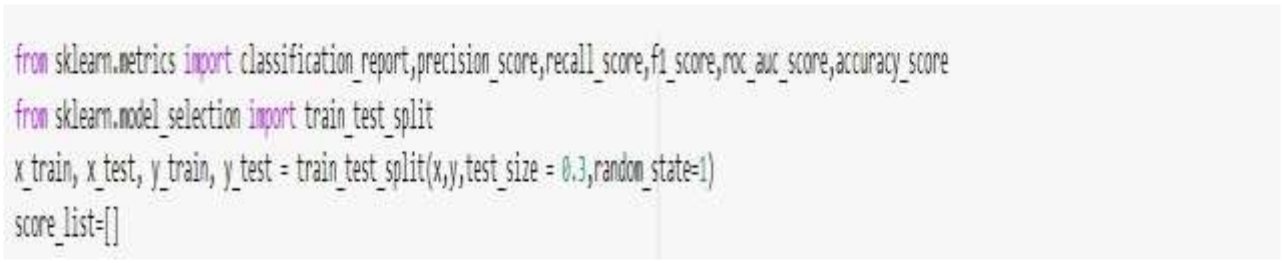
3.3.5 Normalization of Data

Normalization avoids problems with raw data and many datasets by creating new values and maintaining a wide distribution and ratio in the data. To improve the efficiency and accuracy of machine learning models, it also makes use of a variety of techniques and methods. I then divided the data into training and testing groups. I decided to divide the data into 70% training data and 30% testing data.

3.3.6 Training Data:

Making a training set and a testing set from the dataset:
We must separate our dataset into a training set and a test set for machine learning data preparation. This is frequently one of the most important knowledge preparation phases since by doing so, we will improve the functionality of our machine learning model.

Consider the case when we train our machine learning model on one dataset and then test it on a completely other dataset. The knowledge about the relationships between the models will then be tough for our model to understand. If we train our model well and it has a high training accuracy in addition, but we also provide a replacement dataset, the performance will suffer. Therefore, we always strive to create a machine learning model that does well with both the training set and the test dataset. While testing or validation data are used to assess your model's correctness, training data is the initial dataset you use to train a machine learning programme to recognize patterns or perform to your criteria..



```
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-3.3.6.1 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., 70 percent of data is used for training and 30 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 3.3.7.1

3.3.7 Applied Machine Learning Algorithms:

We currently have the train and test data. The next stage is to identify potential training approaches and train our models. We employed three distinct classification methods—KNN, SVM, RF, and DT—because this is frequently a classification challenge. Each method has been applied to the Training dataset, and its accuracy performance is assessed along with the prediction that the Testing data set would be wiped out.

SVM --

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane.

SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

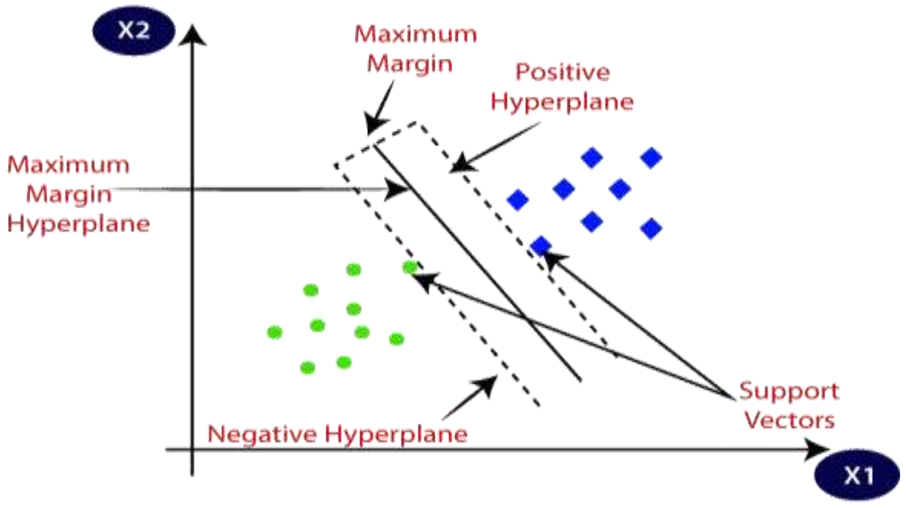


Fig 3.3.7.1- SVM Figure

<u>KNN</u> --

- K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

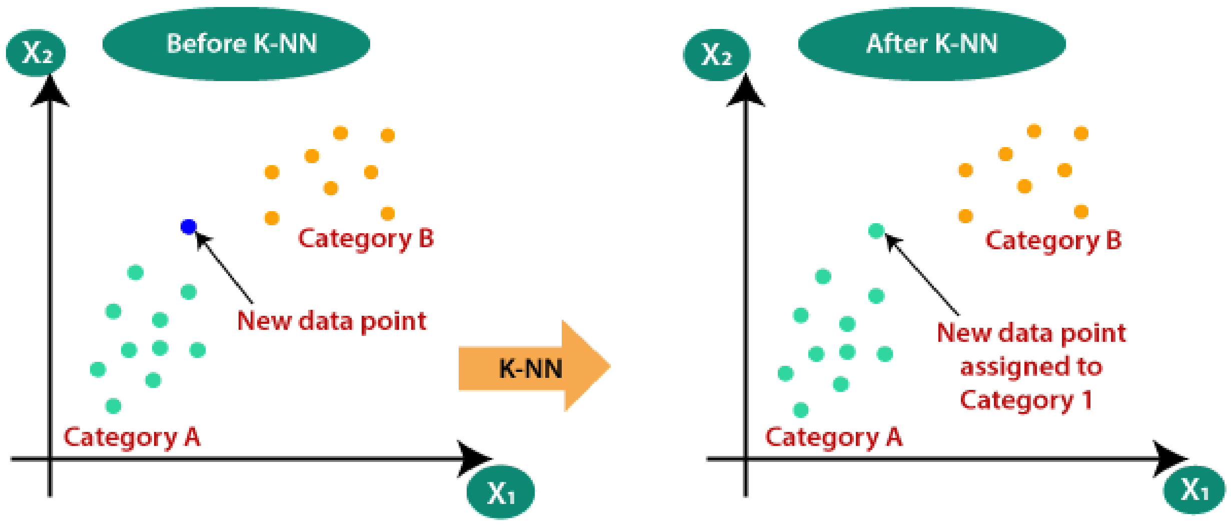


Fig 3.3.7.2 KNN Figure

<u>DT</u> –

- Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**
- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.**
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

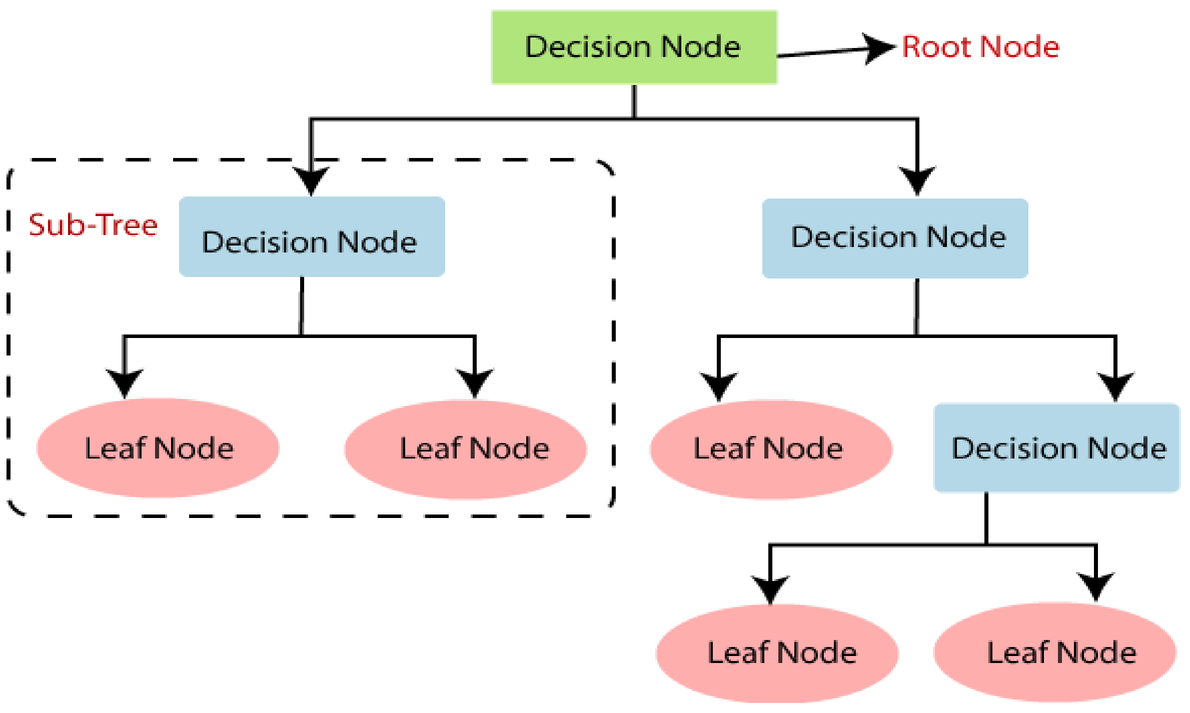


Fig-3.3.7.3 Decision Tree Figure

<u>Random Forest</u> –

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "**Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.**" Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. **The greater number of trees in the forest leads to higher accuracy and prevents the problem of over fitting.**

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

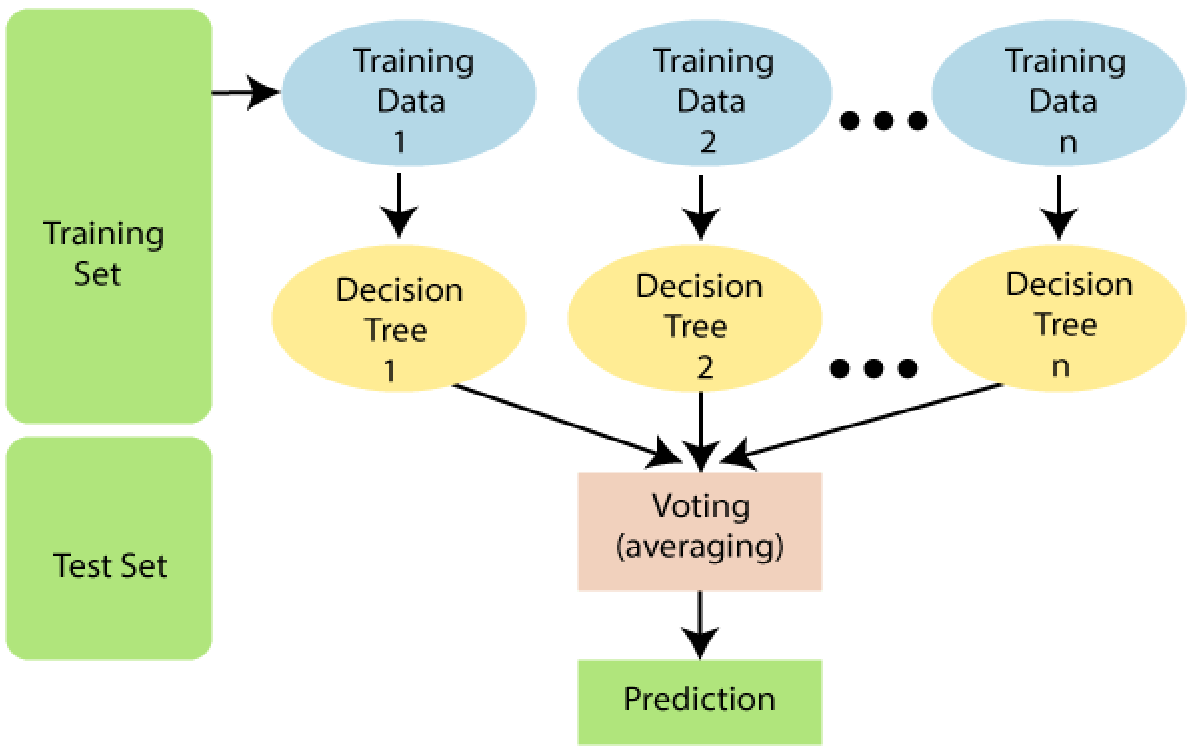


Fig. 3.3.7.4 RF Figure

3.4FACILITIES REQUIRED FOR PROPOSED WORK

1) Software Requirements

- Operating System: Windows Environments
- Browser: Any efficient fast-paced modern-day browser

2) Hardware Requirements

- Processor: Pentium IV Processor or higher
- Hard Disk: 400Mb minimum hard disk storage
- RAM: 2GB or more

3) Developer Requirements

- Operating System: Windows Environments
- Language: Python 3
- Package Installer: Streamlit, Matplotlib, NumPy, Scikit-Learn, Pickle

- Server Base: Local Host
- IDE: Visual Studio Code, Google Collaboratory
- RAM: 4Gb Minimum
- Processor: Pentium IV Processor or higher

CHAPTER 4

PROGRESS WORK DETAILS

4.1 Parkinson Disease Detection Dashboard

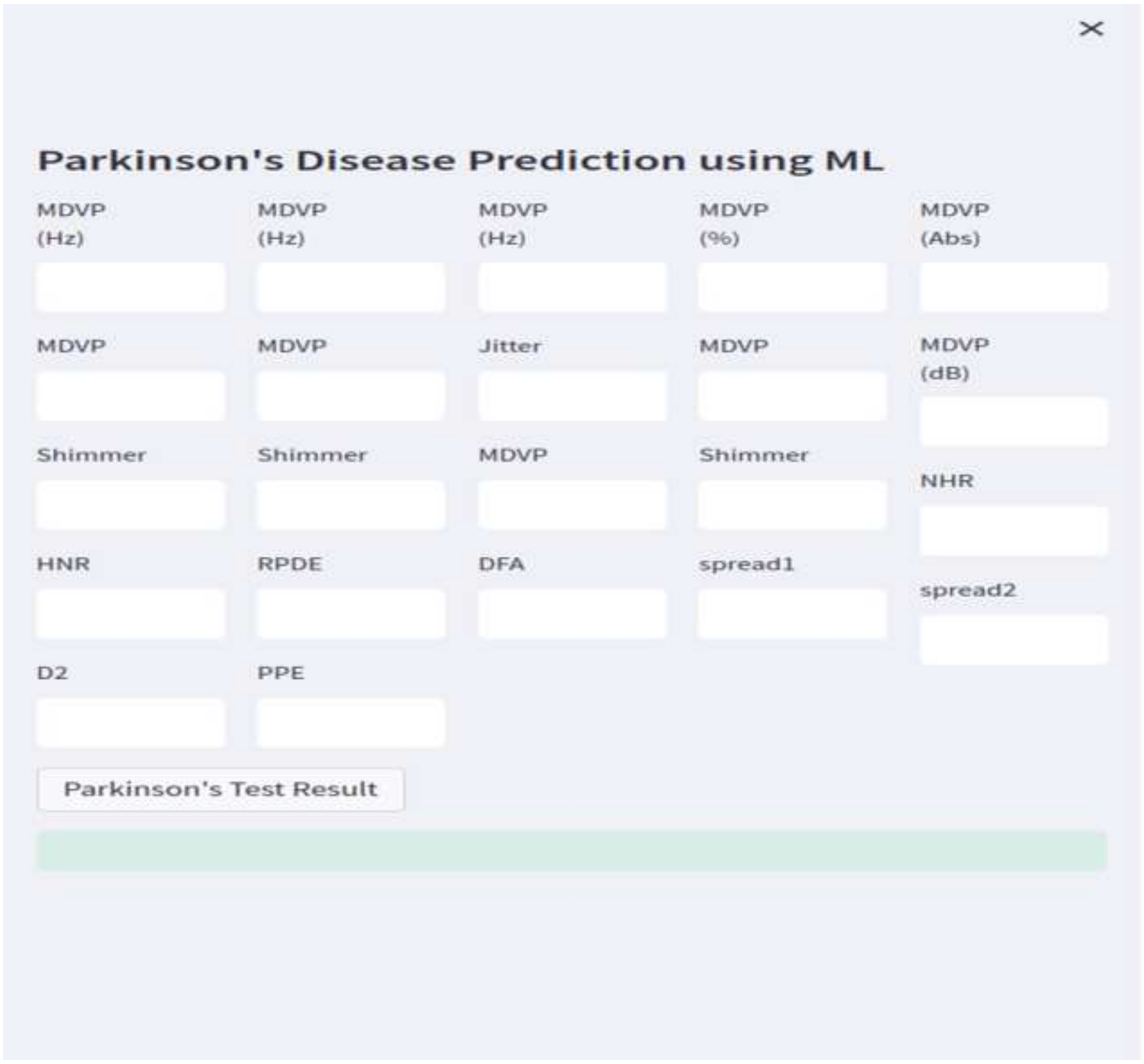


Fig 4.1.1 Dashboard

4.2 System Requirements

A requirement is a feature or a restriction that the system must meet in order for the client to accept it. The goal of requirement engineering is to specify the needs of the system that is being

built. Requirement elicitation, which yields a system specification that the customer can comprehend, and analysis, which yields an analysis model that the developer can clearly explain, are the two key tasks that makeup requirement engineering. An explanation of what the proposed system will accomplish might be a requirement.
Two broad categories can be used to classify requirements:

- Functional Requirements.
- Non-Functional Requirements.

a. Functional Requirements:

A description of the service that the program must provide may be included in a functional requirement. It describes a piece of software or a software system. A function is nothing more than the inputs, behavior, and outputs of the software. The function that a system is likely to accomplish is often defined by a computation, data manipulation, business process, human interaction, or other specified functionality. Independent of its use, functional requirements explain how a system interacts with its surroundings.

- Using the test data as input for the algorithms.
- Show the results together with an explanation of whether or not they indicate Parkinson's.

b. Non-Functional Requirements:

Non-functional requirements outline the software's fundamental characteristics. They assess if the program adhered to non-functional requirements that are essential to the product's success, such as responsiveness, usability, security, portability, and others.
A non-functional requirement would be, "How quickly does the website load?" Systems that fail to satisfy non-functional criteria may not be able to meet user demands.
You can set constraints or limitations on how the system is planned across the various agile backlogs using non-functional requirements.

- Accuracy
- dependability
- flexibility

4.3 System Configuration

- **Software Requirements:**

1) Software
    i) VS Code
    ii) Google Colab
2) Operating System: Windows 10
3) Tools: Web Browser
4) Python Libraries: numpy, pandas, matplotlib, seaborn, sklearn, pickle.

- **Introduction to Python:**

Python is a high-level, all-purpose programming language with an interpreter. Python's straightforward syntax places a strong emphasis on readability, which lowers system maintenance costs. Modules and packages are supported by Python, which encourages system organization and code reuse. Although it takes more time to build its code, it saves space. When coding, indentation must be carefully considered.

Python does the following:

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

For working with machine learning methods, Python includes a wide variety of built-in library functions. Using the "pip" command, all essential Python libraries must be pre-installed.

- **Introduction to Streamlit Framework:**

An internet application developer may design apps without worrying about low-level issues like protocols, thread management, etc. thanks to the Web Application Framework, also known as the Web Framework.
A Python-based open-source app framework is called **Streamlit**. It enables us to quickly develop web applications for data science and machine learning. Major Python libraries like scikit-learn, Keras, PyTorch, SymPy (latex), NumPy, pandas, and Matplotlib are all compatible with it. Callbacks are not required with Streamlit since widgets are regarded as variables. Computation pipelines are made easier and faster by data caching. The application is automatically deployed in the shared link when Streamlit detects modifications to the associated Git repository.

Utilizing Streamlit is easy. A standard Python script is first modified with a few Streamlit instructions, and then it is run using streamlit run:

```
streamlit run your_script.py [-- script args]
```

A local Streamlit server will start up as soon as you run the script as demonstrated above, and your app will launch in a new tab in your default web browser. You can create charts, text, widgets, tables, and more on the app's canvas.

Custom arguments must come after two dashes when being supplied to your script. If not, Streamlit itself will be treated as the object of the parameters. Executing Streamlit as a Python module is an additional method of executing it. When setting up an IDE like VS Code to operate with Streamlit, the following advice might be helpful:

```
# Running
python -m streamlit run your_script.py

# is equivalent to:
streamlit run your_script.py
```

Save the source file every time you wish to update your application. When you do that, Streamlit checks to see if something has changed and then prompts you to relaunch your application. If you want your program to be updated every time you make a change to its source code, choose "Always rerun" in the upper-right portion of your screen.

This enables you to work in a quick interactive loop where you input some code, save it, and then test it out in real time until you're satisfied with the results. One way Streamlit makes your life simpler is the seamless transition between coding and watching outcomes in real time.

- **Python Libraries:**

1. **NumPy:**

   A general-purpose toolkit for handling arrays is called NumPy. It offers a multidimensional array object with outstanding speed as well as capabilities for interacting with these arrays. It is the fundamental Python library for scientific computing. It has a number of characteristics, including the following crucial ones:

   - A strong N-dimensional array object • Complex (broadcasting) operations
   - C/C++ and Fortran code integration tools; practical linear algebra, Fourier transform, and random number functions;

   In addition to its apparent applications in science, NumPy also functions well as a multi-dimensional storage container for general data.

2. **Pandas:**

   Built on top of the NumPy library is the open-source Pandas library. It is a Python library that offers several data structures and actions for working with time series and numerical data. It is quick and offers consumers exceptional performance and productivity. For the Python programming language, it offers high-performance and simple-to-use data structures and data analysis tools. Pandas is used in a variety of academic and professional sectors, including economics, statistics, analytics, and other professions.

3. **Sklearn**

   The most effective and reliable Python library for machine learning is called Scikit-learn (Sklearn). It is an open-source Python library that uses a uniform interface to implement various machine learning, pre-processing, cross-validation, and visualization methods. Through a consistent Python interface, Sklearn offers a variety of effective methods for statistical modeling and machine learning, including classification, regression, clustering, and dimensionality reduction. This library is based on NumPy, SciPy, and Matplotlib and is mostly built in Python.In terms of machine learning libraries for Python, Scikit-learn is without a doubt the most useful. The sklearn toolkit for machine learning and statistical modelling includes a number of helpful features, including classification, regression, clustering, and dimensionality reduction. With sklearn, machine learning models are created.

4. **Pickle:**

   A Python object structure is serialized and deserialized using the Python pickle package. A Python object (list, dict, etc.) can be turned into a character stream by pickling. The concept is that everything needed to recreate the object in another Python script is contained in this character stream. Pickling is advantageous for situations where you want a point of data persistence. The state information for your program is frequently stored on disk, allowing you to continue working on it later.

5. **Matplotlib:**

   It is a very capable plotting package that is beneficial for Python and NumPy users. Visualizing our data is crucial for the creation of statistical interference, and Matplotlib is the tool that will be most useful in this regard. The sole distinction is that it utilizes Python and is open source, providing an interface similar to MATLAB.

6. **Seaborn:**

   On top of Matplotlib, Seaborn may be a data visualization toolkit that is tightly connected with Python's Pandas data structures. The core component of Seaborn that aids in data exploration and comprehension is visualization.
   It provides the following features:

   - Automatic estimation and display of linear regression plots
   - A dataset-oriented API for determining the connection between variables
   - It enables multi-plot grid high-level abstractions
   - Visualizing the distribution of single and multiple variables

4.4 Feasibility Study

The viability of the project and the possibility that the system will be helpful to the organization are both examined in the preliminary inquiry. The major goal of the feasibility study is to determine if it would be technically, operationally, and economically feasible to add new modules and fix existing systems. If a system has infinite resources and time to complete a job, then all systems are feasible.

There are aspects within the feasibility study portion of the preliminary investigation:

- Economical Feasibility

- Technical Feasibility
- Operational Feasibility

Economic Feasibility:

As systems are frequently designed technically, they must nevertheless be an honest investment for the business if they are deployed. The cost of developing the system is weighed against the ultimate benefit obtained from the new system when determining its economic viability. Benefits must match or surpass expenditures in terms of money. The system can be implemented inexpensively. No additional hardware or software is needed. Since the interface for this system was created utilizing already-existing tools and technology, specifically Java 1.6 open source, there is no additional cost, and it is unquestionably economically viable.

Technical Feasibility:

The technological resources that the organization has access to are the main focus of this examination. It aids companies in determining whether the technical resources are enough and whether the technical team has the skills necessary to turn concepts into functional systems. A technical feasibility analysis also looks at the system's software, hardware, and other technological needs. This evaluation is based on an overview of system requirements, which helps determine whether the company has the technical know-how to manage project completion. The following factors need to be taken into account while preparing a feasibility report:

- The component of the business being studied
- The human and economic aspects
- The potential solutions to the problem
- A succinct description of the firm to analyze more potential elements that might affect the study The question at hand at this point (assuming modest cost) is whether the plan is both technically and legally practicable. The technical feasibility evaluation is concerned with assessing the organization's current technological resources and their suitability for the anticipated demands of the proposed system. It's an assessment of the hardware and software and how well it satisfies the requirements of the suggested system.

Operational Feasibility:

Only if they can be developed into an information system are the proposed initiatives valuable. This will satisfy the organization's operational needs. Operational features of the project have to be taken into consideration as a critical component of project implementation. Checking a project's operational viability covers, among other things, the following critical issues:

- Are the users providing the management with enough support?
- If the system is being created and put into use, will it be used and function properly?
- Will the user's opposition to the application destroy any potential benefits? The above-mentioned challenges are what this solution is intended to address. Prior to beginning, management concerns and user needs were taken into account. Therefore, there is no concern about user opposition undermining any potential application advantages. The best possible use of the computer resources would be made possible by the well-planned architecture, which would also contribute to an improvement in performance.

4.5 Code

4.5.1 parkinson_disease_notebook.py

```
# -*- coding: utf-8 -*-

# Importing Libraries

import requests
import pandas as pd
from imblearn.over_sampling import SMOTE
import seaborn as sns
import matplotlib.pyplot as plt


from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn import svm
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB as Naive_Bayes
from sklearn import metrics
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score
from sklearn.datasets import make_classification
from xgboost import XGBClassifier
from sklearn.metrics import ConfusionMatrixDisplay

"""# Data Collection
---

**Dataset Used :** Parkinsons Disease Dataset <br>
**Dataset Source:** UCI Machine Learning Repository <br>
**Dataset Hosting URL:** https://archive.ics.uci.edu/ml/machine-learning-databases/parkinsons/parkinsons.data <br>

"""

# URL For Data Files


url_string = 'https://archive.ics.uci.edu/ml/machine-learning-databases/parkinsons/parkinsons.data'
```

```
# Downloading Content From URL & Storing Into Local File
url_content = requests.get(url_string).content
with open('data.csv', 'wb') as data_file:
  data_file.write(url_content)

# Reading Data Into Pandas Dataframe
df = pd.read_csv('data.csv')

"""# Data Preprocessing


---
 The following steps are performed on the dataset in this section:
 + Dropping Redudant Columns
 + Checking For Duplicated Rows
 + Checking For Missing Values
"""

# Exploring Dataset Content

df.head()

df.tail()

print('Number of Features In Dataset :', df.shape[1])
print('Number of Instances In Dataset : ', df.shape[0])

"""The column **name**, is a *Redundant* column which is not useful for Analysis or Machine
Learning, and will be dropped from the dataframe."""

# Dropping The Name Column

df.drop(['name'], axis=1, inplace=True)

print('Number of Features In Dataset :', df.shape[1])
print('Number of Instances In Dataset : ', df.shape[0])

# Exploring Information About Dataframe

df.info()

df.describe()

"""It can be observed that the column **Status** is stored as *int64* datatype. However, since the
column contains only two numeric values **0 & 1**, we will be changing the datatype to *uint8*,
to save Memory Space. """

df['status'] = df['status'].astype('uint8')

df.info()
```

```
# Checking For Duplicate Rows In Dataset

print('Number of Duplicated Rows :',df.duplicated().sum())

"""As observed in the above step, the dataset does **NOT** contain any Duplicated Rows."""

# Checking For Missing Values In Dataset

df.isna().sum()

"""As seen in the above step, **No Columns** of the dataset contains any Missing Values.

# Exploratry Data Analysis
"""

#Balance of Data

sns.countplot(x='status',data=df)

fig, ax = plt.subplots(figsize=(20,20))
sns.heatmap(df.corr(),annot=True,ax=ax)

"""In this correlation heatmap, we can see that many independent features are highly correlated with eachother."""

#Box Plot

fig,axes=plt.subplots(5,5,figsize=(15,15))
axes=axes.flatten()

for i in range(1,len(df.columns)-1):
    sns.boxplot(x='status',y=df.iloc[:,i],data=df,orient='v',ax=axes[i])
plt.tight_layout()
plt.show()

"""From the boxplot shown above it is very evident that if a patient has a lower rate of 'HNR','MDVP:Flo(Hz)','MDVP:Fhi(Hz)','MDVP:Fo(Hz)' ,then he/she is affected by parkinsons disease."""

plt.rcParams['figure.figsize'] = (15, 4)
sns.pairplot(df,hue = 'status', vars = ['MDVP:Jitter(%)','MDVP:Jitter(Abs)','MDVP:RAP','MDVP:PPQ', 'Jitter:DDP'] )
plt.show()


"""From the above pair plot we can understand that all these fundamental frequencies are highly correlated with eachother."""
```

```
plt.rcParams['figure.figsize'] = (15, 4)
sns.pairplot(df,hue = 'status', vars = ['MDVP:Shimmer','MDVP:Shimmer(dB)','Shimmer:APQ3','Shimmer:APQ5','MDVP:APQ','Shimmer:DDA'] )
plt.show()

"""From the above pair plot we can understand that all these measures variation in amplitude are highly correlated with eachother.

# Balancing Dataset
---
In this section, as it is observed that the Dataset Is Heavily Imbalanced, with Number of Samples of Parkinson Disease Samples being 147, and Non-Parkinson Being only 48. Hence, in this section, we make use of **SMOTE** to **Oversample** and Balance the dataset.
"""

# Exploring Imabalance In Dataset


df['status'].value_counts()

# Extracting Features Into Features & Target

X = df.drop(['status'], axis=1)
y = df['status']

print('Feature (X) Shape Before Balancing :', X.shape)
print('Target (y) Shape Before Balancing :', y.shape)

# Intialising SMOTE Object

sm = SMOTE(random_state=300)


# Resampling Data


X, y = sm.fit_resample(X, y)

print('Feature (X) Shape After Balancing :', X.shape)
print('Target (y) Shape After Balancing :', y.shape)

# Scaling features between -1 and 1  for normalization

scaler = MinMaxScaler((-1,1))


# define X_features , Y_labels

X_features = scaler.fit_transform(X)
```

```
Y_labels = y


# splitting the dataset into traning and testing sets into 80 - 20


from sklearn.model_selection import train_test_split
X_train , X_test , y_train , y_test = train_test_split(X_features, Y_labels , test_size=0.20,
random_state=20)

"""# Machine Learning Model Training
In this section, we have trained the following Machine Learning Models:
+ Decision Tree Classifier
+ Random Forest Classifier
+ Logistic Regression
+ SVM
+ Naive Bayes
+ KNN Classifier
+ XGBoost Classifier


## Decision Tree Classifier"""


clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)
predDT = clf.predict(X_test)

print(classification_report(y_test, predDT))

param_grid = {
    'max_features': ['sqrt', 'log2'],
    'max_depth' :range(1,10),
    'random_state':range(30,210,30),
    'criterion' :['gini', 'entropy']
}
CV_dt = GridSearchCV(estimator=clf, param_grid=param_grid, cv= 5)
CV_dt.fit(X_train, y_train)

CV_dt.best_params_

dt1=DecisionTreeClassifier(random_state=120, max_features='sqrt', max_depth=6,
criterion='entropy')
dt1.fit(X_train, y_train)
predDT = dt1.predict(X_test)
print(classification_report(y_test, predDT))


"""**Decision tree confusion matrix**"""
```

```
ConfusionMatrixDisplay.from_estimator(clf, X_test, y_test, cmap= "Blues")
plt.show()

y_pred_proba = dt1.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred_proba)
auc = metrics.roc_auc_score(y_test, y_pred_proba)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()


"""## Random Forest Classifier"""


rfc = RandomForestClassifier()
rfc.fit(X_train, y_train)
predRF = rfc.predict(X_test)

print(classification_report(y_test, predRF))

param_grid = {
    'n_estimators': range(100,300,25),
    'max_features': ['sqrt', 'log2'],
    'max_depth' :range(1,10),
    'random_state':range(100,250,50),
    'criterion' :['gini', 'entropy']
}
CV_rfc = GridSearchCV(estimator=rfc, param_grid=param_grid, cv= 5)
CV_rfc.fit(X_train, y_train)

CV_rfc.best_params_

rfc1=RandomForestClassifier(random_state=200, max_features='auto', n_estimators= 125, 
max_depth=7, criterion='entropy')
rfc1.fit(X_train, y_train)
predRFC = rfc1.predict(X_test)
print(classification_report(y_test, predRFC))


"""**Random forest confusion matrix**"""


ConfusionMatrixDisplay.from_estimator(rfc1, X_test, y_test, cmap= "Reds")
plt.title('Confusion matrix for Random Forest', y=1.1)
plt.show()

y_pred_proba = rfc1.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred_proba)
```

```
auc = metrics.roc_auc_score(y_test, y_pred_proba)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()

"""## Logistic Regression"""

logmodel = LogisticRegression()
logmodel.fit(X_train, y_train)
predlog = logmodel.predict(X_test)

print(classification_report(y_test, predlog))
print("Confusion Matrix:")
confusion_matrix(y_test, predlog)

ConfusionMatrixDisplay.from_estimator(logmodel, X_test, y_test, cmap="Greens")
plt.title('Confusion matrix for Logistic Regression', y=1.1)
plt.show()

y_pred_proba = logmodel.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred_proba)
auc = metrics.roc_auc_score(y_test, y_pred_proba)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()

"""##SVM"""

#Create a svm Classifier


clf = svm.SVC(kernel='linear') # Linear Kernel


#Train the model using the training sets


clf.fit(X_train, y_train)

#Predict the response for test dataset

y_pred = clf.predict(X_test)

# Model Accuracy: how often is the classifier correct?

print("Test Set Accuracy:",metrics.accuracy_score(y_test, y_pred))

X_pred = clf.predict(X_train)
print("Train Set Accuracy:",metrics.accuracy_score(y_train, X_pred))
```

```
param_grid = {'kernel':['linear','rbf','poly'],'C': [0.5, 1, 10, 100],
             'gamma': [1, 0.1, 0.01, 0.001, 0.0001]}

grid_SVC = GridSearchCV(svm.SVC(), param_grid, scoring='f1', verbose = 3)
grid_SVC.fit(X_train, y_train)

# print best parameter after tuning

print("\nBest Parameters: ", grid_SVC.best_params_)

# print how our model looks after hyper-parameter tuning

print("\n", grid_SVC.best_estimator_)

predSVC = grid_SVC.predict(X_test)

# print classification report

print("\n", classification_report(y_test, predSVC))

ConfusionMatrixDisplay.from_estimator(grid_SVC, X_test, y_test, cmap="Purples")
plt.title('Confusion matrix for SVM', y=1.1)
plt.show()

fpr, tpr, _ = metrics.roc_curve(y_test,  predSVC)
auc = metrics.roc_auc_score(y_test, predSVC)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()


"""## Naive Bayes """

# Naive Bayes


gnb = Naive_Bayes()
gnb.fit(X_train, y_train)
predgnb = gnb.predict(X_test)

print(classification_report(y_test, predgnb))

print("Confusion Matrix:")
confusion_matrix(y_test, predgnb)

# scores -check how efficiently labels are predicted
accuracy_testing = accuracy_score(y_test, predgnb)
print("Accuracy % :",accuracy_testing*100)

ConfusionMatrixDisplay.from_estimator(gnb, X_test, y_test, cmap="Oranges")
```

```
plt.title('Confusion matrix for Naive Byes', y=1.1)
plt.show()

y_pred_proba = gnb.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred_proba)
auc = metrics.roc_auc_score(y_test, y_pred_proba)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()


"""## KNN Classifier"""


import numpy as np

Ks = 10
mean_acc = []
ConfustionMx = [];
for n in range(2,Ks):

#Train Model and Predict

    neigh = KNeighborsClassifier(n_neighbors = n).fit(X_train,y_train)
    yhat=neigh.predict(X_test)
    mean_acc.append(metrics.accuracy_score(y_test, yhat))
print('Neighbor Accuracy List')
print(mean_acc)

plt.plot(range(2,Ks),mean_acc,'g')
plt.ylabel('Accuracy ')
plt.xlabel('Number of Neighbours (K)')
plt.tight_layout()
plt.show()

knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
predKNN = knn.predict(X_test)

ConfusionMatrixDisplay.from_estimator(knn, X_test, y_test, cmap="Blues")
plt.title('Confusion matrix KNN', y=1.1)
plt.show()

y_pred_proba = knn.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred_proba)
auc = metrics.roc_auc_score(y_test, y_pred_proba)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()
```

```
"""## XGBoost Classifer
In this section, we have trained a XGBoost Classifier, for classification of Instances to be
Parkinsons or Not. The following parameters of the XGBoost Classifier have been optimized in this
section:
+ **Max Depth**: This value is used to determine the Maximum Depth of the Tree.
+ **ETA** : This is also known as Learning Rate.
+ **Reg_Lambda** : This is the L2 Regularization for the weights.
+ **Random State** : This is used to evaluate and determine the performance of the model based
on different random states.

The *Parameter Optimization* has been performed using **GridSearchCV** with the following
parameters:
+ **Scoring Parameter**: F1 Score
+ **Cross Validation**: 3
"""


# Defining Parameter Dictionary


param_dict = {'max_depth': range(4,8), 'eta' : [0.1, 0.2, 0.3, 0.4, 0.5],
              'reg_lambda' : [0.8, 0.9, 1, 1.1, 1.2],
              'random_state': [300, 600, 900]}

clf = GridSearchCV(XGBClassifier(), param_grid = param_dict,
                   scoring = 'f1', cv = 3, verbose = 1)
clf.fit(X_train, y_train)

# Extracting Best Classifier From GridSearchCV

xgb_clf = clf.best_estimator_


# Evaluating Performance on Train Set


pred = xgb_clf.predict(X_train)
print('For Train Set')
print('Accuracy :', metrics.accuracy_score(y_train, pred))
print('Precision :', metrics.precision_score(y_train, pred))
print('Recall :', metrics.recall_score(y_train, pred))
print('R2 Score :', metrics.r2_score(y_train, pred))


# Evaluating Performance on Train Set


predXGB = xgb_clf.predict(X_test)
print('For Test Set')
print('Accuracy :', metrics.accuracy_score(y_test, predXGB))
print('Precision :', metrics.precision_score(y_test, predXGB))
print('Recall :', metrics.recall_score(y_test, predXGB))
print('R2 Score :', metrics.r2_score(y_test, predXGB))

ConfusionMatrixDisplay.from_estimator(xgb_clf, X_test, y_test, cmap="Blues")
plt.title('Confusion matrix for XGBoost', y=1.1)
plt.show()


"""# Comparision Table"""


from sklearn.metrics import precision_score,recall_score ,accuracy_score, f1_score, r2_score, log_loss

chart = {
        'Metric':["Accuracy", "F1-Score", "Recall", "Precision", "R2-Score"],
        'DT':[accuracy_score(y_test, predDT), f1_score(y_test, predDT), recall_score(y_test, predDT), precision_score(y_test, predDT), r2_score(y_test, predDT)],
        'RF':[accuracy_score(y_test, predRFC), f1_score(y_test, predRFC), recall_score(y_test, predRFC), precision_score(y_test, predRFC), r2_score(y_test, predRFC)],
        'SVM':[accuracy_score(y_test, predSVC), f1_score(y_test, predSVC), recall_score(y_test, predSVC), precision_score(y_test, predSVC), r2_score(y_test, predSVC)],
        'NB':[accuracy_score(y_test, predgnb), f1_score(y_test, predgnb), recall_score(y_test, predgnb), precision_score(y_test, predgnb), r2_score(y_test, predgnb)],
        'KNN':[accuracy_score(y_test, predKNN), f1_score(y_test, predKNN), recall_score(y_test, predKNN), precision_score(y_test, predKNN), r2_score(y_test, predKNN)],
        'XGB':[accuracy_score(y_test, predXGB), f1_score(y_test, predXGB), recall_score(y_test, predXGB), precision_score(y_test, predXGB), r2_score(y_test, predXGB)]
}
chart = pd.DataFrame(chart)

display(chart)

import pickle

filename = "SVM_model.sav"
pickle.dump(clf, open(filename, "wb"))
```

4.5.2 dashboard.py

```
import pickle
import streamlit as st
from streamlit_option_menu import option_menu
from pathlib import Path


# loading the saved models
parkinsons_model = pickle.load(open('/mnt/DATA/College_project/Adoption-of-best-Machine-Learning-Algorithm-for-Parkinson-s-Disease/parkinsons_model.sav', 'rb'))



with st.sidebar:
#    selected = option_menu('Parkinsons Prediction', options: )



# Parkinson's Prediction Page


    # page title
    st.title("Parkinson's Disease Prediction using ML")

    col1, col2, col3, col4, col5 = st.columns(5)

    with col1:
        fo = st.text_input('MDVP:Fo(Hz)')

    with col2:
        fhi = st.text_input('MDVP:Fhi(Hz)')

    with col3:
        flo = st.text_input('MDVP:Flo(Hz)')

    with col4:
        Jitter_percent = st.text_input('MDVP:Jitter(%)')

    with col5:
        Jitter_Abs = st.text_input('MDVP:Jitter(Abs)')

    with col1:
        RAP = st.text_input('MDVP:RAP')

    with col2:
        PPQ = st.text_input('MDVP:PPQ')

    with col3:
        DDP = st.text_input('Jitter:DDP')
```

```
with col4:
    Shimmer = st.text_input('MDVP:Shimmer')

with col5:
    Shimmer_dB = st.text_input('MDVP:Shimmer(dB)')

with col1:
    APQ3 = st.text_input('Shimmer:APQ3')

with col2:
    APQ5 = st.text_input('Shimmer:APQ5')

with col3:
    APQ = st.text_input('MDVP:APQ')

with col4:
    DDA = st.text_input('Shimmer:DDA')

with col5:
    NHR = st.text_input('NHR')

with col1:
    HNR = st.text_input('HNR')

with col2:
    RPDE = st.text_input('RPDE')

with col3:
    DFA = st.text_input('DFA')

with col4:
    spread1 = st.text_input('spread1')

with col5:
    spread2 = st.text_input('spread2')

with col1:
    D2 = st.text_input('D2')

with col2:
    PPE = st.text_input('PPE')



# code for Prediction
parkinsons_diagnosis = ''

# creating a button for Prediction
if st.button("Parkinson's Test Result"):
    parkinsons_prediction = parkinsons_model.predict([[fo, fhi, flo, Jitter_percent, Jitter_Abs,
RAP,
PPQ,DDP,Shimmer,Shimmer_dB,APQ3,APQ5,APQ,DDA,NHR,HNR,RPDE,DFA,spread1,spread
2,D2,PPE]])

        if (parkinsons_prediction[0] == 1):
          parkinsons_diagnosis = "The person has Parkinson's disease"
        else:
          parkinsons_diagnosis = "The person does not have Parkinson's disease"

    st.success(parkinsons_diagnosis)
```

4.6 Experimental Analysis and Performance Measures

Performance assessment metrics are the factors that aid in the comparative study of various machine learning approaches, i.e., they reveal the best algorithm among all other algorithms or methods that medical research may employ for the early diagnosis of Parkinson's disorders. To assess the accuracy of the predictions, we have employed a variety of metrics. Confusion Matrix, Accuracy, Precision, Recall or Sensitivity, Specificity, F1-score, LR-, LR+, odd, and Youden score are some examples of these metrics.

- **Confusion Matrix:**

  The error matrix is another name for the confusion matrix. It is a table that is frequently used to summarize how well a classification algorithm performs on a set of test data when the real values are known. Instances belonging to a predicted class are represented in each column of the matrix. The correlation matrix is shown as follows:

Table 4.6.1 Confusion Matrix

	Positive	**Negative**
Positive	**TP**	**TN**
Negative	**FP**	**FN**

Where, TP: True Positive

FP: False Positive

FN: False Negative

TN: True Negative

- **Accuracy:**

  Accuracy is the proportion of the total number of predictions that were correct. It can be obtained by the sum of true positive and true negative instances divided by the total number of Samples.

  It is expressed as :

$$\text{Accuracy} = (TP+TN) / (TP+FP+FN+TN)$$

- **Precision:**

  Precision is a fraction of true positive and predicted yes instances. It is also known as the ratio of correct positive results to the total number of positive results predicted by the system.

  It is expressed as :

$$\text{Precision}(P) = TP / (TP + FP)$$

- **Recall (or) Sensitivity:**

  Recall is defined as the fraction between True Positive instances and Actual yes instances or it is the ratio of correct positive results to the number of all relevant samples.

  It is expressed as:

$$\text{Recall}(R) = TP / (TP + FN)$$

- **F1-score:**

  F1-score is the fraction between product of the recall and precision to the summation of recall and precision parameter of classification. It is the harmonic mean of Precision and Recall. It measures the test accuracy. The range of this metric is 0 to 1.

  It is expressed as:-

$$\text{F1 score} = 2 * 1/((1/\text{Precision})+(1/\text{recall})) = 2PR / (P+R)$$

- **Specificity:**

Specificity is a measure of how well a test can identify true negatives. Specificity is also referred to as selectivity or true negative rate, and it is the percentage, or proportion, of the true negatives out of all the samples that do not have the condition (true negatives and false positives).

It is expressed as:

$$\text{Specificity} = TN / (TN+FP)$$

- **Comparative Analysis:**

  The model that suits our system has been found out through mentioned graph comparison. In this graph we showed the comparative analysis of the three models based on some of the above mentioned evaluation metrics.

Table 4.6.2 Comparative Analysis

Metric	**DT**	**RF**	**SVM**	**KNN**
Accuracy	0.932203	0.966102	0.966102	0.966102
F1-Score	0.920000	0.961538	0.960000	0.960000
Recall	0.884615	0.961538	0.923077	0.923077
Precision	0.958333	0.961538	1.000000	1.000000
R2-Score	0.724942	0.862471	0.862471	0.862471

CHAPTER 5

CONCLUSION & FUTURE WORK

5.1 CONCLUSION

Parkinson's disease is a degenerative neurological condition that impairs movement and can result in stiffness, tremors, and issues with balance and coordination. Parkinson's disease does not yet have a cure, but there are a number of treatments and therapies that can help control the symptoms and enhance quality of life for those who have the condition. Levodopa and dopamine agonists are some of the drugs used to treat Parkinson's disease, along with physical therapy, occupational therapy, and speech therapy. For people with advanced Parkinson's disease who have not reacted well to alternative treatments, deep brain stimulation surgery may possibly be a possibility.

Parkinson's disease patients should collaborate closely with their medical professionals to create a specialised treatment regimen that takes into account their unique requirements and symptoms. A nutritious diet, frequent exercise, and stress-reduction tactics can all help control Parkinson's symptoms and enhance general health. In this study, in order to directly characterize the two main biophysical factors of disordered voices: increased nonlinear, complex aperiodicity and non-Gaussian, aero acoustic breath noise, we have introduced recurrence and scaling analysis methods. We introduced a new, combinednonlinear/stochastic signal model of speech production that is capable, in principle, of producing the wide variation in behavior of normal and disordered voice examples. To exploitthe output of this model in practice, and hence all types of normal and disordered voices, we explored the use of two nonlinear measures: the recurrence period density entropy and detruded fluctuation analysis.

5.2 FUTURE WORK

A comparison of various machine learning models has been carried out for Parkinson's disease Prediction. However, using deep learning methods has not been explored for PD prediction to a great extent. In the future, the work can be extended by using the deep learning framework with auto-encoders to reduce the number of features and to extract the mostimportant from them. Also, the UCI dataset used in this work is small and not so complex, therefore, the auto-encoder may not learn well. The auto-encoders learn well with a large and complex dataset to give better results. Future work needs to be done in the area of developingcompact, light-weight models with higher processing speed to ensure their suitability for mobile applications in the ubiquitous computing environment.

In future, these models can be trained with different datasets that have best features and can be predicted more accurately. If the accuracy rate increases, it can be used by the laboratories and hospitals so that it is easy to predict in early stages. This models can be also used with different medical and disease datasets. In future the work can be extended by building a hybrid model that can find more than one disease with an accurate dataset and that dataset has common features of two diseases. In future the work can extended to build a model that may extract more important features among all features in the dataset so that it produce more accuracy.

REFERENCES

[1] B K Varghese, G. B. Amali D*, Uma Devi K S “Parkinson's Disease Prediction Using Machine Learning Approaches” 2013 Fifth International Conference on Advanced Computing(ICoAC)

[2] Parkinson's Disease Diagnosis Using Machine Learning and Voice" by Timothy J. Wroge, Yasin Ozkanca, C. Demiroglu, Dong Si, David C. Atkins, and R. Ghomi.

[3] Gokul.S, Sivachitra.M, “Parkinson's Disease Prediction Using Machine Learning Approaches” 2013 Fifth International Conference on Advanced Computing (ICoAC)

[4] Early Prediction of Parkinson Disease Using Machine Learning and Deep Learning Approaches Harshvardhan Tiwari, Shiji K Shridhar, Preeti V Patil, K R Sinchana and G Aishwarya.

[5] A Comparative Analysis Of Parkinson Disease Prediction Using Machine Learning Approaches F.M. Javed Mehedi Shamrat, Md. Asaduzzaman, A.K.M. Sazzadur Rahman, Raja Tariqul Hasan Tusher, Zarrin Tasnim.

[6] Comparative Analysis Of The Early Detection Of Parkinson's Disease J. Olivia Capitola Assistant professor, Computer Application Department & Affiliated to Bharathidasan University, Bishop Heber College, Tiruchirappalli

[7] Support Vector Regression and its Mathematical Implementation, Rahul Rastogi, Medium,2020.

[8] Saloni, R. K. Sharma, and A. K. Gupta, “Detection of Parkinson Disease Using ClinicalVoice Data Mining,” vol. 9, pp. 320–326, 2015.

[9] Tarigoppula V.S Sriram , M. Venkateswara Rao , G V Satya Narayana3 , DSVGK Kaladhar4 , T Pandu Ranga Vital, “Intelligent Parkinson Disease Prediction Using Machine Learning Algorithms” International Journal of Engineering and Innovative Technology (IJEIT) Volume 3, Issue 3, September 2013.

[10] Braak, H. a. B. E., 2000. Pathoanatomy of Parkinson’s disease. Journal of neurology, 247(2), pp. II3--II10.

[11] Matthews, V., 2016. An Analysis of Nonlinear and Fractal Measures in Vocal Frequencies forDeveloping Models to Predict Parkinson’s *Disease.* [Online] Available at: https://rstudio-pubs-static.s3.amazonaws.com/197257_4ecb7314bc1f4619b59ab44a1374ce7f.html

[12] De Lau, L. M. a. B. M. M., 2006. Epidemiology of Parkinson's disease. The Lancet N5(6), pp.

[13] Timothy J. Wroge, Yasin Özkanca, C. Demiroğlu, Dong Si, David C. Atkins and R. Ghomi (2018), “Parkinson’s Disease Diagnosis Using Machine Learning and Voice”, Computer Science, IEEE Signal Processing in Medicine and Biology Symposium (SPMB).

[14] Saloni, R. K. Sharma, and A. K. Gupta, “Detection of Parkinson Disease Using Clinical Voice Data Mining,” vol. 9, pp. 320–326, 2015.

[15] Davie, C. A., 2008. A review of Parkinson's disease. British medical bulletin, 86(1), pp. 109- 127.

[16] Sveinbjornsdottir, S., 2016. The clinical symptoms of Parkinson's disease. Journal of neurochemistry, Volume 139, pp. 318-324.

[17] Tarigoppula V.S Sriram , M. Venkateswara Rao , G V Satya Narayana3 , DSVGK Kaladhar4 , T Pandu Ranga Vital, “Intelligent Parkinson Disease Prediction Using Machine Learning Algorithms” International Journal of Engineering and Innovative Technology (IJEIT) Volume 3, Issue 3, September 2013.

[18] Satish Srinivasan, Michael Martin & Abhishek Tripathi, “ANN based Data Mining Analysis of Parkinson’s Disease” International Journal of Computer Applications, vol-168,June 2017.

[19] 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.

[20] T. Swapna, Y. Sravani Devi, “Performance Analysis of Classification algorithms on Parkinson’s Dataset with Voice Attributes”. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 2 pp. 452-458, 2019.

[21] 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.