Machine Learning Model for the Detection and Prediction of Parkinson's Disease based on Audio Signals

SYNOPSIS

Submitted in partial fulfillment of the requirement of the degree of

BACHELOR IN COMPUTER APPLICATION

In

Computer Science and Engineering

By

Kartikey Raghuvanshi [00718002020]

Saakshi Srivastava [01518002020]

Shubham Tiwari [02118002020]

Sanand Mishra [01718002020]

Under the supervision of

Ms. Ayasha Malik

Assistant Professor

Affiliated to GGSIP University, New Delhi
Approved by AICTE & Council of Architecture

Delhi Technical Campus

GREATER NOIDA

(Affiliated by Guru Gobind Singh Indraprastha University)

DECLARATION BY THE STUDENT

We, Shubham Tiwari, Kartikey Raghuvanshi, Sanand Mishra, and Saakshi Srivastava, students of BCA hereby declare that the project titled “Machine Learning Model for the Detection and Prediction of Parkinson’s Disease based on Audio Signals” which is submitted by us to the Department of Computer Science, DELHI TECHNICAL CAMPUS, Noida, in partial fulfillment of the requirement for the award of the degree of Bachelor in Computer Application has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition.

The Author attests that permission has been obtained for the use of any copyrighted material appearing in the Project report other than brief excerpts requires only proper acknowledgment in scholarly writing and all such use is acknowledged.

Signature

Greater Noida

Date:

Shubham Tiwari, Kartikey Raghuvanshi,
Sanand Mishra, Saakshi Srivastava

CERTIFICATE OF ORIGINALITY

On the basis of the declaration submitted by Shubham Tiwari, Kartikey Raghuvanshi, Sanand Mishra, and Saakshi Srivastava students of BCA. I hereby certify that the project titled “Machine Learning Model for the Detection and Prediction of Parkinson’s Disease based on Audio Signals” which is submitted to, DELHI TECHNICAL CAMPUS, Noida, in partial fulfillment of the requirement for the award of the degree of Bachelor in Computer Application is an original contribution with existing knowledge and faithful record of work carried out by them under my guidance and supervision.

To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Date: Ayasha Malik

Computer Science and Engineering

DELHI TECHNICAL CAMPUS, Greater Noida

ABSTRACT

A neurological ailment that affects millions of individuals worldwide is Parkinson's disease (PD). The ability to quickly intervene and improve patient outcomes depends on the early and precise recognition of PD. By analysing a variety of data sources, including clinical records, genetic markers, imaging scans, and sensor inputs, machine learning (ML) approaches have showed promise in improving the identification and diagnosis of PD. This abstract gives a general overview of how ML is used in PD identification. ML algorithms can efficiently learn patterns and characteristics from PD-related data to identify and predict the existence of the illness. These algorithms use supervised and unsupervised learning approaches. The algorithms can discover different patterns and biomarkers linked with PD by training ML models on large and representative datasets, enabling reliable identification even in the early stages. By gathering real-time information on atypical movement patterns, tremors, and other PD symptoms, the combination of ML with wearable technology and sensor technologies improves PD identification. To give continuous monitoring and individualised insights into illness development, ML models can process and analyse this data. The lack of varied and trustworthy datasets for training and validation is one of the difficulties in PD detection using ML. To create reliable ML models, it is essential to ensure the amount, quality, and representativeness of the data. In addition, resolving privacy issues and adhering to data protection laws are crucial for protecting patient data. The promise of ML in PD diagnosis is highlighted in the abstract's conclusion, including its capacity to increase diagnostic precision, enable early intervention, and support individualised treatment plans. ML-based PD detection systems have the potential to revolutionise the industry, aid medical professionals, and enhance the quality of life for people with Parkinson's disease as ML techniques continue to progress, adding explainable AI and resolving ethical issues.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to my project guide **“AYASHA MALIK”** for giving us the opportunity to work on this topic. It would never be possible for us to take this project to this level without her innovative ideas and her relentless support and encouragement.

We also express our gratitude to our HoD **“DR. SEEMA VERMA”** for her support and guidance.

Shubham Tiwari (02118002020)
Kartikey Raghuvanshi (00718002020)
Sanand Mishra (01718002020)
Saakshi Srivastava (01518002020)

CONSENT FORM

This is to certify that we, Shubham Tiwari, Kartikey Raghuvanshi, Sanand Mishra, and Saakshi Srivastava, students of BCA of 2020 - 2023 presently in the VIth Semester at DELHI TECHNICAL CAMPUS, Greater Noida give our consent to include all our personal details, Shubham Tiwari (02118002020), Kartikey Raghuvanshi (00718002020), Sanand Mishra (01718002020) and Saakshi Srivastava (01518002020) for all accreditation purposes.

Place:

Date:

Shubham Tiwari (02118002020)
Kartikey Raghuvanshi (00718002020)
Sanand Mishra (01718002020)
Saakshi Srivastava (01518002020)

DECLARATION FORM (Health, Safety & Plagiarism)

We, Shubham Tiwari, Kartikey Raghuvanshi, Sanand Mishra, and Saakshi Srivastava students of BCA of Computer Science and Engineering, Enrollment No., batch Shubham Tiwari (02118002020), Kartikey Raghuvanshi (00718002020), Sanand Mishra (01718002020) and Saakshi Srivastava (01518002020), Department of Computer Science and Engineering at Delhi Technical Campus, GGSIP University, New Delhi, hereby declare that utmost care was taken not to harm anyone either physically or emotionally. We have gone through project guidelines including plagiarism.

Date:

Place:

Students Signature

Shubham Tiwari

Kartikey Raghuvanshi

Sanand Mishra

Saakshi Srivastava

DELHI TECHNICAL CAMPUS

(Affiliated Guru Gobind Singh Indraprastha University, New Delhi)

Greater Noida

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LSIT OF ABBRENIATIONS

S. No.	Abbreviations	Definition
1	PD	Parkinson's Disease
2	ML	Machine Learning
3	SVM	Support Vector Machine
4	KNN	K Nearest Neighbors
5	DFD	Data Flow Diagram
6	AI	Artificial Intelligence
7	AUC	Area Under Curve
8	ROC	Receiver Operating Characteristic

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

A neurodegenerative condition that affects the central nervous system is Parkinson's disease. It is characterized by the gradual death of dopamine-producing brain cells, which causes tremors, stiffness, bradykinesia (slowness of movement), and postural instability among motor symptoms. Typically, a medical practitioner will diagnose Parkinson's disease based on a clinical assessment of these symptoms. The use of machine learning (ML) techniques to aid with Parkinson's disease detection and diagnosis is on the rise. Large data sets may be analyzed by ML algorithms, which can also spot patterns that may not be immediately obvious to human observers. It's crucial to remember that ML models are meant to support the diagnosis process rather than to replace the knowledge of healthcare practitioners. They can offer more information and aid in sorting patients according to priority for additional assessment or monitoring. Parkinson's disease diagnosis methods using machine learning (ML) are subject to the correctness and reliability of the data, feature selection, and ML algorithm performance. Therefore, to increase the precision and dependability of these systems, continuing study and validation are required. Parkinson's disease must be found through the application of numerous diagnostic techniques and tests, as well as the recognition of recognizable symptoms. Normally, a healthcare expert will begin by gathering all relevant medical information and doing a comprehensive physical examination. They will assess any accompanying non-motor symptoms as well as the existence and evolution of any motor symptoms, such as tremors, stiffness, and bradykinesia. The severity of Parkinson's disease symptoms is evaluated using a variety of grading systems, and their development over time is tracked. The Unified Parkinson's Disease Rating Scale (UPDRS), which assesses both motor and non-motor symptoms, is the most often used rating scale. hereditary testing may be advised in rare circumstances, notably for those with a family history of early-onset Parkinson's disease or who have certain hereditary. Levodopa, which raises dopamine levels in the brain, is one medicine that has been shown to be effective in treating Parkinson's disease. It's crucial to remember that there is currently no test or biomarker that can clearly identify the presence of Parkinson's disease. The clinical assessment and elimination of other potential explanations of the same symptoms form the foundation of the diagnosis. The competence of medical specialists with knowledge of Parkinson's disease is necessary for an accurate diagnosis. In recent years, researchers have also investigated the use of computational methods and machine learning algorithms to help with Parkinson's disease detection and diagnosis. To find patterns and signs linked to the condition, these tools analyse substantial datasets, including clinical data, neuroimaging data, and sensor data. ML-based methods have the potential to enhance early detection and aid medical personnel in providing precise diagnosis. However, further study is required to certify and improve these techniques for typical clinical application. Artificial intelligence (AI) has a subset known as machine learning (ML), which focuses on creating algorithms and models that let computers learn and make predictions or judgements without having to be explicitly programmed. The performance of ML algorithms is iteratively improved as they are exposed to more data and learn from patterns and data. The ML algorithm is trained on labelled data in supervised learning, where each data point is connected to a predefined target or result. Based on the supplied labels, the algorithm learns to translate input features to the desired output. Support vector machines, neural networks, and decision trees are a few examples of supervised learning techniques. Algorithms for unsupervised learning, in which there are no predetermined labels or results, are trained on unlabeled data. These algorithms examine the data's structures and patterns in order to find any underlying linkages or groups. Unsupervised learning approaches include dimensionality reduction methods like principal component analysis (PCA) and clustering algorithms. Through trial-and-error training, an algorithm may be taught to interact with its surroundings and discover the best course of action. The algorithm learns to maximize rewards over time by receiving feedback in the form of rewards or penalties based on its actions. Applications for reinforcement learning may be found in robotics, video games, and self-driving cars.It's critical to remember that machine learning algorithms are not perfect, and that their effectiveness depends on the accuracy and representativeness of the data they use. Additionally, ML models should be regularly updated and monitored to ensure their continued accuracy and reliability.

1.2 NEED AND SIGNIFICANCE

There are several advantages to using machine learning (ML) to detect Parkinson's disease (PD), and it has great potential for enhancing patient care and diagnosis. Here are a few main justifications for why ML might be useful in PD detection -

- Early Detection: To find patterns and signs connected to PD, ML systems may examine a significant quantity of data, including clinical information, imaging data, and sensor data. Using ML approaches, it could be able to identify PD while symptoms are mild or have not yet become clinically evident. Early identification enables prompt intervention and therapy, which may halt the disease's course and enhance results.
- Assessment that is Objective and Quantitative: ML models can offer assessments that are both objective and quantitative for evaluating PD-related symptoms. Because PD symptoms can be subjective and might differ from patient to patient, this is very helpful. ML algorithms may examine sensor data from gyroscopes or accelerometers to measure bradykinesia, gait irregularities, or tremor severity objectively. These metrics can help

medical practitioners track the development of an illness and assess the efficacy of therapies.

- Integration of Multiple Data Sources: Wearable sensor data, clinical data, imaging data, and genetic data may all be used in PD diagnosis utilising machine learning. ML models can capture a more complete image of the disease by combining these various data kinds, potentially resulting in more precise and thorough diagnostic evaluations.
- Healthcare Professionals' Decision Support Systems: ML-based PD detection techniques may be used as systems that help healthcare professionals make decisions. By analysing patient data, contrasting it with known facts, and producing likelihood estimates or classifications relating to the existence of PD, these technologies can aid in the diagnosing process. These technologies can increase diagnosis accuracy and assist healthcare practitioners in making better informed decisions.
- Personalised medicine: ML models may examine significant datasets to pinpoint PD traits or subtypes. This can aid in grouping individuals according to the characteristics of their diseases and how well they respond to therapy. By assisting in the identification of relevant medicines or interventions catered to each patient's unique needs, ML-based techniques can enable personalised medicine.
- Research and insights: To find new connections and insights into PD, ML algorithms may analyse vast datasets, such as clinical records and genetic data. Indicators of PD development or progression, such as genetic markers or biomarkers, may be found using machine learning techniques. The creation of focused solutions and continuing research can both benefit from this knowledge.
- Correct Diagnosis: PD is a complicated neurodegenerative disorder with symptoms that may be confused with those of other illnesses. Testing aids in separating PD from other illnesses with symptoms that may be similar. In order to receive the proper care and management, a precise diagnosis is essential.
- Patient Education and Support: Patients and their families are better informed about the disorder, how it progresses, and the services that are available when PD is officially diagnosed. It enables access to support networks, instruction on symptom management, and lifestyle changes, all of which can improve the patient's coping skills and general wellbeing.
- Differential diagnosis: Movement disorders including atypical parkinsonism and essential tremor have features in common with PD. When these illnesses are properly diagnosed, it

is possible to differentiate between them and create individualised treatment regimens for each problem.

- Illness monitoring and prognosis: Testing can provide information about the severity, course, and prognosis of the illness. The ability to identify changes in symptoms through routine monitoring and follow-up evaluations enables medical personnel to modify treatment programmes and offer the proper support.
- Family planning and genetic counselling: Rarely, genetic alterations can lead to PD. Individuals who may be at risk of inheriting the illness or who have genetic variants linked to PD can be found through genetic testing. Decisions about family planning and genetic counselling can be aided by this knowledge.

To discuss symptoms, perform necessary testing, and acquire an accurate diagnosis, it is crucial to speak with healthcare specialists with knowledge in neurology and movement disorders. They may help people through the testing process, offer individualised treatment, and support. It's critical to emphasise that while ML exhibits potential in the identification of PD, clinical judgement should still be used as a support mechanism. Medical practitioners should properly evaluate and supervise the integration of ML models into clinical practise once they have been verified using rigorous scientific methodologies. To improve and verify ML-based therapies, more study and collaboration between ML specialists, doctors, and researchers are required.

1.3 OBJECTIVE

- Using machine learning (ML), the main goal of Parkinson's disease (PD) detection is to create accurate and dependable models that can help with the rapid and precise diagnosis of PD. By utilising the power of algorithms, ML-based techniques seek to analyse a variety of datasets, spot trends, and offer unbiased assessments of PD-related symptoms.
- Through the analysis and integration of several data sources, such as clinical data, imaging data, genetic markers, and sensor data, machine learning (ML) algorithms can assist increase the accuracy of PD diagnosis. Large datasets allow ML models to learn patterns and indicators that could be suggestive of PD, enabling more precise and objective diagnostic evaluations.
- By utilising ML algorithms, PD detection seeks to improve diagnosis precision, enable early intervention, and provide individualised treatment for Parkinson's disease patients. Prior to being included into standard clinical practise, ML models should complement clinical experience and be rigorously evaluated using scientific methodologies.
- Using various types of algorithms in order to find the best for the early and accurate detection of Parkinson’s is one of the main objectives.

1.4 PURPOSE

The aim of Parkinson's disease (PD) detection using machine learning (ML) is to enhance PD diagnosis' precision, effectiveness, and accessibility. In order to analyse big datasets and extract patterns and characteristics that might help in the early identification and diagnosis of PD, ML-based methods use algorithms and computational tools. Enhancing diagnostic precision, enabling early intervention, supporting personalised treatment, and advancing research are the goals of PD detection using ML. To ensure the dependability and effectiveness of ML models, it is crucial to thoroughly evaluate them before integrating them into clinical practise. This is done under the proper review and supervision by medical specialists.

1.5 INTENDED USER

The numerous stakeholders engaged in the diagnosis, treatment, and research of Parkinson's disease (PD) might be considered among the intended consumers of machine learning (ML)-based PD detection. These users might be -

- Healthcare Professionals: The main users of ML-based PD detection systems might include neurologists, movement disorder experts, and other healthcare professionals engaged in the diagnosis and management of PD. They may use ML models as decision support tools to help with PD patient monitoring, therapy planning, and correct diagnosis.
- Researchers and Scientists: To analyse sizable datasets, find trends, and pinpoint possible biomarkers or genetic markers connected to PD, researchers and scientists in the fields of neurology and movement disorders can employ ML-based PD detection models. ML can help research efforts and increase our understanding of PD.
- Clinical Trial Investigators: For clinical trial investigators, ML-based PD detection models can be useful in identifying and sifting through suitable candidates for clinical studies including PD. In order to more effectively and precisely recruit participants for trials, machine learning (ML) algorithms can help identify people who have certain illness traits or indicators that satisfy the trial's inclusion requirements.
- Patients and carers can gain from the increased accuracy and early diagnosis offered by these models, while not being the direct users of ML-based PD detection systems. Early diagnosis enables prompt action and access to the best therapies, thereby enhancing patient outcomes and quality of life.
- Public Health Authorities: To learn more about the prevalence, distribution, and effects of PD at the population level, public health authorities and organisations that work in PD research, policy-making, and resource allocation can use ML-based PD detection models. By identifying high-risk groups, directing resource allocation, and influencing public health policy, ML can aid in public health initiatives.

It's crucial to remember that ML-based PD detection systems should be created in conjunction with medical experts, verified using exacting scientific procedures, and implemented into clinical practise after careful assessment and supervision. Healthcare professionals' input is still essential for interpreting ML findings, making clinical judgements, and giving PD patients individualised therapy.

1.6 APPLICATIONS

Machine learning (ML)-based Parkinson's disease (PD) detection has several applications in PD diagnosis, care, and research. The following are some of the main uses of PD detection using ML

- Predictive Modelling: ML methods may be used to create predictive models that estimate the course of the disease or anticipate how PD patients will respond to therapy. Medical professionals may optimise treatment regimens and enhance patient management by using ML models to provide personalised predictions by examining previous data and patient characteristics.
- Identification of Potential Biomarkers for PD: ML-based PD detection models can help in the identification of Potential Biomarkers for PD. Large datasets may be analysed using ML algorithms to find patterns and features that could be used as illness markers. These biomarkers can aid in the creation of new diagnostic procedures and targeted interventions.
- Remote monitoring and telemedicine: ML-based PD detection models may be included into telemedicine platforms. Healthcare practitioners may remotely monitor patients with PD, evaluate symptoms, and decide on medication modifications or interventions by using sensor data and ML algorithms.

To ensure their dependability, accuracy, and ethical usage, ML-based apps in PD detection should be created and verified using rigorous scientific methodologies, and their incorporation into clinical practise should be done in cooperation with healthcare experts.

1.7 COMPONENTS

Machine learning (ML)-based Parkinson's disease (PD) detection relies on a number of interrelated components that work together to provide a reliable and accurate detection method. The following are the main elements of PD detection using ML –

- Data Collection - ML-based PD identification relies heavily on relevant data collection.
- Data Preprocessing - Preprocessing processes are required before ML algorithms are used on input data.
- Feature Selection - In order to distinguish PD from other diseases or forecast PD-related outcomes, the most pertinent and useful elements from the gathered data must be chosen.

- Model Selection - Based on the characteristics of the data, the issue being solved, and the precise objectives of PD detection, ML models are chosen.
- Model Training - In order to train the machine learning model, the preprocessed data must be fed into the chosen method, and its parameters must be optimised so that the system can learn from the data's patterns and correlations.
- Model Evaluation - Utilising validation datasets not utilised during training, the ML model must be assessed after training.
- Testing and Deployment - The ML model is tested on different datasets once it has been trained and reviewed to determine how well it performs in real-world circumstances.
- Monitoring and Iteration - To make sure that ML models for PD detection remain accurate and pertinent, they need to be continuously monitored and periodically evaluated.

These elements cooperate in an iterative process that results in more accurate and reliable PD diagnosis using ML approaches by refining and improving the ML model based on feedback and new data.

1.8 LIMITATIONS

Machine learning (ML)-based Parkinson's disease (PD) diagnosis promises potential improvements, however it's important to take into account the methods drawbacks and difficulties. The following are some of the main drawbacks of PD detection using ML -

- Data Quality and Availability: ML models rely significantly on high-quality datasets that have been carefully vetted. But gathering extensive and varied information for PD detection can be difficult. The performance and generalizability of ML models may be impacted by the lack of widespread access to datasets containing standardised and precisely labelled PD instances.
- Data Bias and Generalizability: ML models may inherit biases existing in the data if they were trained on particular datasets. The model's predictions might not translate well to different groups if the training data is not typical of the overall PD community (e.g., biassed towards a particular demographic or clinical profile). To prevent biassed results, it is essential to make sure that the training data are diverse and representative.
- Interpretability and Explainability: Many ML algorithms, especially complicated deep learning models, can be difficult to understand and comprehend. Because of this lack of interpretability, it may be challenging for medical experts to comprehend and believe the model's predictions. The interpretability of ML models is an important factor to take into account, especially in the context of healthcare where decision-making requires explainability.

- Overfitting and Generalisation: ML models may become too optimised to the training data, which makes it difficult for them to generalise successfully to new, unforeseen data. Overfitting can result in exaggerated training performance measures but worse performance in real-world situations. To solve this difficulty, regularisation approaches and thorough assessment on separate datasets are required.
- Ethical Issues: ML-based PD detection systems bring up ethical issues with regard to data security, privacy, and privacy. To allay these worries and keep patients' trust, it's essential to provide adequate permission, data anonymization, and safe storage and transmission of sensitive medical information.
- Clinical Integration and Validation: The effective integration of machine learning-based PD detection into clinical practise necessitates rigorous validation, regulatory compliance, and significant consultation with healthcare specialists. To make sure ML models deliver useful and applicable information to help clinical decision-making, it is vital to evaluate their performance, safety, and efficacy in real-world contexts. This is done through clinical validation studies.
- Limited Understanding of Disease processes: The fundamental processes and aetiology of PD are still not completely known, despite great breakthroughs. Without a thorough knowledge of the illness causes, there may be limits in the accuracy and reliability of ML models, which mainly rely on patterns and characteristics collected from data.
- Lack of Longitudinal Data: Because PD is a progressive condition, it is essential to have longitudinal data to accurately diagnose and monitor the disease. However, gathering longitudinal data might be difficult, and ML models might not be able to adequately capture and explain the dynamics of illness development.

While ML-based PD detection has a lot of potential, it is crucial to proceed cautiously with its development and implementation, making sure to conduct rigorous validation, attending to ethical issues, and taking into account the ML approach's unique constraints. To create reliable and therapeutically effective ML models for PD diagnosis, collaboration between data scientists, doctors, and researchers is essential.

1.9 FEASIBILITY STUDY

The practicality and viability of establishing an ML-based PD detection system are evaluated in a feasibility study of Parkinson's disease (PD) detection using machine learning (ML). To ascertain if the project is technically, economically, and operationally feasible, numerous factors must be evaluated. Here are some important things to think about before starting a feasibility study -

- Technical Viability: This entails assessing the accessibility of data and machine learning (ML) methods appropriate for PD detection. Analyse the usefulness and accessibility of datasets such as clinical information, genetic markers, imaging scans, or sensor signals. Make sure there are ML algorithms and methods available that can analyse the data efficiently and produce reliable PD detection findings.
- Data Accessibility and Availability: Determine whether there is a sufficient amount of data accessible to train and test ML models. Take into account elements like the dataset's size, data variety, and the demographic and PD stage representation. Consider any potential data security and privacy risks as well, and make sure all applicable laws are followed.
- Determine the effectiveness of ML models for detecting PD by evaluating their performance and accuracy. Analyse the ML models' performance in terms of accuracy, sensitivity, specificity, precision, and other pertinent metrics using the datasets that are readily available. Think about if the ML models are capable of detecting PD with the appropriate degree of accuracy and reliability.
- Resources and Infrastructure: Consider the computing power, set-up, and knowledge needed to put the ML-based PD detection system into practise. Think about things like the necessary hardware, software dependencies, and the availability of qualified staff to create, install, and maintain the ML models.
- Cost-Benefit Analysis: To ascertain whether installing the ML-based PD detection system is economically feasible, perform a cost-benefit analysis. Take into account the expenses related to data collection, preprocessing, model creation, infrastructure setup, and continuous upkeep. Consider the possible advantages, such as improved patient outcomes, reduced healthcare costs, and increased diagnostic accuracy.
- Operational Considerations: Examine the viability of incorporating the ML-based PD detection system into the current workflows and procedures in the healthcare industry. Think about aspects including user acceptability, usability, integration with EHR systems, and the effect on clinical decision-making. Discuss the operational consequences with stakeholders and healthcare experts to get their perspectives.
- Considerations in Terms of Ethics and Law: Examine the moral and legal implications of using ML to identify PD. Think about concerns like patient privacy, data security, informed consent, and adherence to relevant laws like HIPAA or GDPR. Make that the ML-based PD detection method abides by moral standards and protects patient rights.

Organisations may make educated judgements on the deployment of ML-based PD detection systems by completing a comprehensive feasibility assessment. As a result, stakeholders are better able to assess the project's feasibility and make the required modifications to ensure its effective execution. It assists in identifying potential obstacles, risks, and needs.

1.10 ORGANIZATION OF REPORT

Your report must be well organised before writing it. For the audience to understand the information presented, the report must be carefully ordered. This is particularly valid if you discuss how machine learning may be used to detect Parkinson’s disease. It is essential to outline the research design, machine learning techniques used, results obtained, and conclusions in this report. The report's introduction makes an effort to provide a broad overview of the objectives and research methods used. Include some background information on the same to aid readers in understanding the results. The introduction should be followed by a full discussion of the study methodology. This includes details on the data sources utilised, the machine learning techniques used, and the evaluation standards applied. The next part describes the results of using machine learning techniques. A clear and in-depth explanation of every finding should be provided. The report should also contain a comparison of the results with those from other studies on the same topic. A summary of the conclusions and recommendations for further investigation should be included in the report's conclusion.

CHAPTER 2: PROBLEM STATEMENT AND LITERATURE SURVEY OF THE TECHNOLOGIES

2.1 PROBLEM STATEMENT

The following is a definition of the problem statement for the identification of Parkinson's disease (PD) using machine learning (ML) -

The goal is to create a machine learning (ML) model that accurately detects the presence of Parkinson's disease (PD), differentiates it from other movement disorders, or forecasts disease progression in PD patients, given a dataset containing various types of data including clinical information, genetic markers, imaging data, sensor data, and patient-reported outcomes.

The following elements are included in the problem statement –

- Detection: Based on the provided data, the ML model should successfully determine if a person has Parkinson's disease (PD) or not. A binary classification output from the model should be available, indicating whether PD is present or not.
- Differential diagnosis: The ML model should be able to distinguish PD from other illnesses in circumstances when the symptoms may overlap with other movement disorders or ailments. To help medical practitioners make precise and well-informed differential diagnoses, it should include probabilities or classifications.
- Predictive modelling: The ML model should be able to forecast how the disease will develop or how PD patients will respond to therapy. To anticipate how the disease will develop in the future or to determine if a certain therapy will be effective, it should make use of previous data and patient characteristics.

In order to effectively diagnose PD, distinguish it from other disorders, and make predictions about the course of the disease or the effectiveness of therapy, the problem statement calls for the development of an ML model that makes the best use of the data that is currently available. The ultimate objective is to advance PD research, assist personalised medication, improve early detection, and increase diagnostic precision.

2.2 LITERATURE REVIEW

Technology has greatly influenced and will continue to significantly affect how people live today as a result of the increasing use of computers. Almost every industry area now uses technology much more often. The study done previously on this subject by academics and researchers is included here for your benefit and to provide you a more thorough understanding of how to diagnose Parkinson's disease using machine learning.

Table 2.1 – Literature Survey

S.No.	Year	Name	Contribution
1	2011	Heisters D. [1]	Parkinson's disease is an irreversible neurological ailment that causes slowness of movement, tremor, and stiffness of the muscles. The main form of therapy is medication, and continuing research is being done to discover a cure and provide new therapies.
2	2012	A. Ozcift [2]	With up to 97% accuracy in the top-performing classifier, a novel classification model based on support vector machine and rotation forest ensemble classifiers has been created to enhance Parkinson's disease detection.
3	2012	Dr. R. Geetha Ramani et al. [3]	This study employs data mining methods and biological voice measurements to categorise the severity of Parkinson's disease with 100% accuracy using the Random Tree classification algorithm and ReliefF algorithm.
4	2013	Farhad Soleimanian Gharehehopogh et al. [4]	This study classifies Parkinson's disease with great accuracy using two types of artificial neural networks (MBF and MLP), which can help neurologists make better choices.
5	2016	Dragana Miljkovic et al. [5]	The use of machine learning techniques to identify and categorise tremors, gait patterns, and voice dysfunction in Parkinson's disease patients is covered in this research.
6	2016	Arvind kumar tiwari [6]	In this study, random forest with 20 chosen characteristics is used to predict Parkinson's disease with an overall accuracy of 90.3%.
7	2018	Dr. Anupam bhatia et al. [7]	In order to identify the most precise classification method, this research intends to identify Parkinson's Disease by data mining and statistical study of typical symptoms including gait, tremors, and micro-graphia.
8	2018	M. Abdar et al. [8]	This study uses Parkinson's disease data from UCI to evaluate the diagnostic performance of SVM and

			Bayesian networks, and it revealed that SVM with polynomial kernel function and C parameter performed the best, with an average accuracy of 99.18%. Additionally, the SVM algorithm's 10 most crucial components were found.
9	2019	Carlo Ricciardi et al. [9]	Data mining can provide light on the small variations between Parkinson's disease and Progressive Supranuclear Palsy, which can be distinguished via gait analysis.
10	2020	Anila M et al. [10]	The study proposes a unique method for accurately diagnosing Parkinson's disease using artificial neural network models.

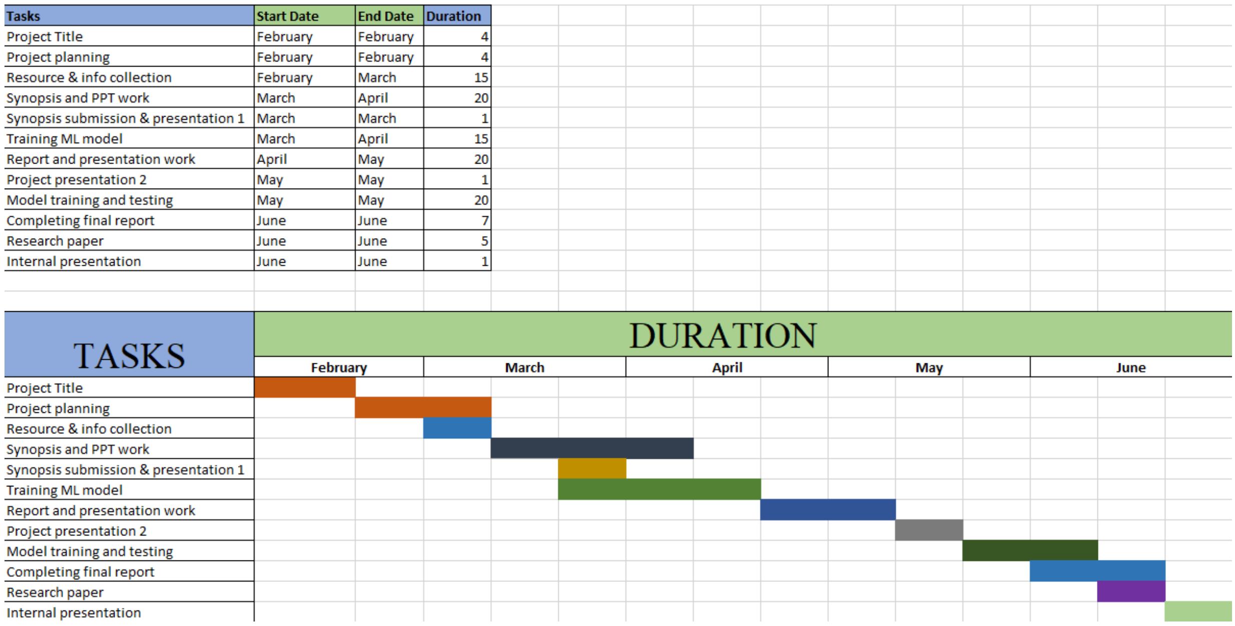
CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1 PLANNING AND SCHEDULING

- WEEK 1
  - Forming a group of 4 members
  - Discussion on the topic
- WEEK 2
  - Deciding the title of the project.
  - Discussion on the languages/technologies to work on.
  - Researching about feasibility study of the project.
- WEEK 3
  - Writing down the objectives.
  - Discussion about further implementation.
  - Preparing and presenting synopsis in presentation 1.
- WEEK 4
  - Study about the Parkinson’s disease.
  - Researching about the awareness among people in today’s era.
  - Studying about the existing models present in this domain.
- WEEK 5
  - Creating the home page with the help of HTML
  - Deciding and creating the elements on the home page.
- WEEK 6
  - Collecting datasets.
  - Installing required python libraries
- WEEK 7
  - Learning Pandas
  - Learning Numpy.
  - Learning sklearn and svm.
- WEEK 8
  - Training model through Logistic regression and Decision Tree.
  - Testing the accuracy of the prediction done.
- WEEK 9
  - Model training through random forest - Information
  - Model training through random forest - Entropy
  - Testing the accuracy of the prediction done by model

- WEEK 10
    - Testing the accuracy of model trained with the help of random forest.
    - Training and testing the model trained through SVM & KNN.
    - Preparation of project presentation 2.
- WEEK 11
    - Updating the synopsis.
    - Project presentation 2.
    - Training and testing the model with the help of Gaussian Naïve Bayes & Bernoulli Naïve Bayes
- WEEK 12
    - Comparison between all model trained through different algorithms.
    - Performance and accuracy are measured.
- WEEK 13
    - Comparing different models on the basis of performance and accuracy.
    - Creating the structure of our website.
- WEEK 14
    - Creating home page.
    - Creating main content page.
    - Creating about section and footer.
- WEEK 15
    - Creating remaining HTML pages.
    - Designing different HTML pages with the help of CSS.
- WEEK 16
    - Completing the thesis.
    - Reading different research articles and papers
- WEEK 17
    - Removing plagiarism from project report.
    - Completing research paper.
    - Reviewing and finalizing the project report and research paper.

3.1.1 Gantt Chart



Tasks	Start Date	End Date	Duration
Project Title	February	February	4
Project planning	February	February	4
Resource & info collection	February	March	15
Synopsis and PPT work	March	April	20
Synopsis submission & presentation 1	March	March	1
Training ML model	March	April	15
Report and presentation work	April	May	20
Project presentation 2	May	May	1
Model training and testing	May	May	20
Completing final report	June	June	7
Research paper	June	June	5
Internal presentation	June	June	1

Figure 3.1: Gantt chart

3.2 SOFTWARE AND HARDWARE REQUIREMENTS

3.2.1 Hardware Requirements

- PC/Laptop
    - Processor – Intel I3 or above
    - 4 GB RAM
    - Processor – Intel I3 or above
    - Display: Dual XGA (1024 x 768) or higher resolution monitors
    - Operating system: Windows7
- Computer Network connection (Ethernet/Wi-Fi)

3.2.2 Software Requirements and Library Requirements

3.2.2.1 Software

- Python - Python is a popular high-level, interpreted programming language because of its clarity, simplicity, and adaptability. It was created by Guido van Rossum and made accessible in 1991. Python supports procedural, object-oriented, and functional programming techniques. Thanks to its big standard library and rich ecosystem of third-party packages, it is suitable for a variety of applications. Python is commonly used in a variety of contexts, including web development, data analysis, scientific computing, artificial intelligence, machine learning, automation, and scripting. It appeals to both novice and professional developers because to its clarity, readability, and wide community support.

3.2.2.2 Libraries

- PIP - Pip is a Python package manager that enables the installation, administration, and updating of Python packages and libraries. It is the default Python package manager and is pre-installed on the majority of Python distributions.
- NumPy - The foundational Python package for scientific computing is called NumPy (Numerical Python). It offers support for large, multidimensional arrays and matrices, as well as a range of mathematical operations for effectively using these arrays. Numerous additional libraries in the scientific Python environment are built on the foundation of NumPy.
- Pandas - Pandas is a powerful library for handling and analysing data. It offers data structures, like as DataFrames, that make handling and modifying structured data simple. Data may be readily read, filtered, transformed, aggregated, and visualised with pandas. It is frequently employed in exploratory data analysis and data preparation.
- Matplotlib - Matplotlib is a Python charting package that makes it possible to build a variety of static, animated, and interactive visualisations. You may generate line plots, scatter plots, bar charts, histograms, and many other types of plots using the variety of customization options and plotting methods that are offered.
- Seaborn - A data visualisation library developed on top of Matplotlib is called Seaborn. It offers a sophisticated interface for producing beautiful statistics visuals. In addition to offering extra statistical features like visualising distributions, plotting regressions, and examining correlations between variables, Seaborn makes it easier to create sophisticated visualisations.
- Sklearn - Python's scikit-learn is a well-liked machine learning package. It offers a broad selection of machine learning tools and algorithms for jobs including model assessment, dimensionality reduction, clustering, regression, and classification. A complete package for machine learning applications, scikit-learn also provides tools for data preparation, feature selection, and model selection.
- XGBoost - The family of gradient boosting algorithms includes XGBoost (eXtreme Gradient Boosting), a potent and popular machine learning technique. By building an ensemble of weak prediction models, such decision trees, then combining their predictions to produce precise and reliable forecasts, it is intended to tackle classification and regression issues. Due to XGBoost's superior predictive performance, adaptability, and efficiency, it has grown in popularity. It has been effectively used in a number of industries, including those where precise forecasts and interpretability are crucial, such as banking, healthcare, retail, and web analytics.

If Python and pip are already installed, the following commands may be put into the command line to install these libraries –

- pip install pandas
- pip install numpy
- pip install matplotlib
- pip install seaborn
- pip install scikit-learn
- pip intall xgboost

Once installed, you may use these libraries' features by importing them into your Python project. For instance -

- import numpy as np
- import pandas as pd
- import matplotlib.pyplot as plt
- import seaborn as sns
- from sklearn import datasets, model_selection
- from xgboost import xgboostclassifier

CHAPTER 4: SYSTEM DESIGN

4.1 CONCEPTUAL MODEL

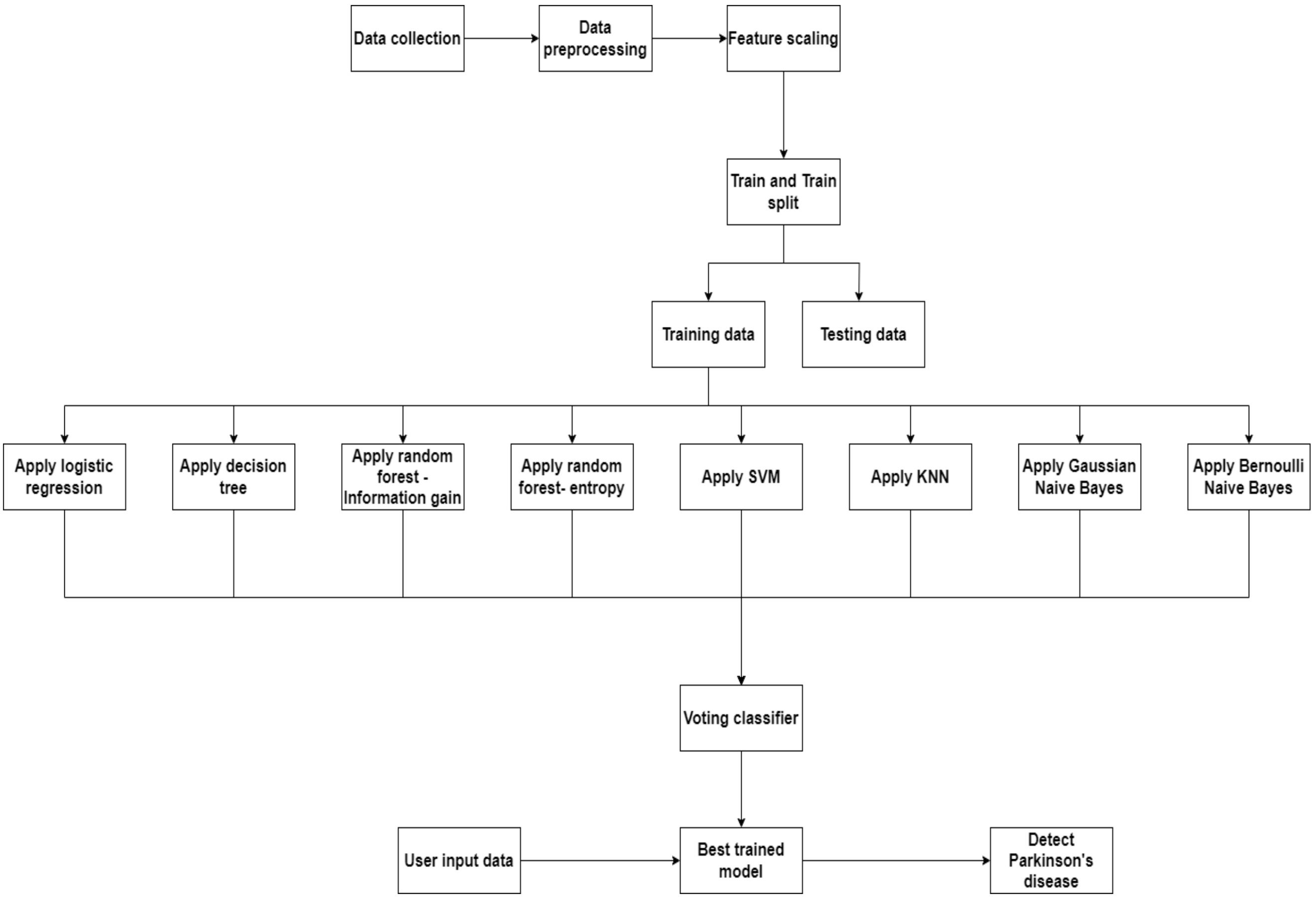


Figure 4.1: Conceptual model

4.2 ABOUT DATASET

In order to follow disease development, identify risk factors, and assess the efficacy of therapies, data collecting is a crucial part of research into Parkinson's disease. Data collection is done from Kaggle with 24 feature/characteristics of 195 people of different age groups. The features are:

- name - ASCII subject na
- me and recording number
- MDVP:Fo(Hz) - Average vocal fundamental frequency
- MDVP:Fhi(Hz) - Maximum vocal fundamental frequency
- MDVP:Flo(Hz) - Minimum vocal fundamental frequency
- MDVP:Jitter(%), MDVP:Jitter(Abs), MDVP:RAP, MDVP:PPQ, Jitter:DDP - Several measures of variation in fundamental frequency
- MDVP:Shimmer,MDVP:Shimmer(dB),Shimmer:APQ3,Shimmer:APQ5,MDVP:APQ,Shimmer:DDA - Several measures of variation in amplitude

- NHR, HNR - Two measures of the ratio of noise to tonal components in the voice
- status - The health status of the subject (one) - Parkinson's, (zero) – healthy
- RPDE, D2 - Two nonlinear dynamical complexity measures
- DFA - Signal fractal scaling exponent
- spread1,spread2,PPE - Three nonlinear measures of fundamental frequency variation

The correlation between all the features is represented in the below mentioned heatmap of the dataset.

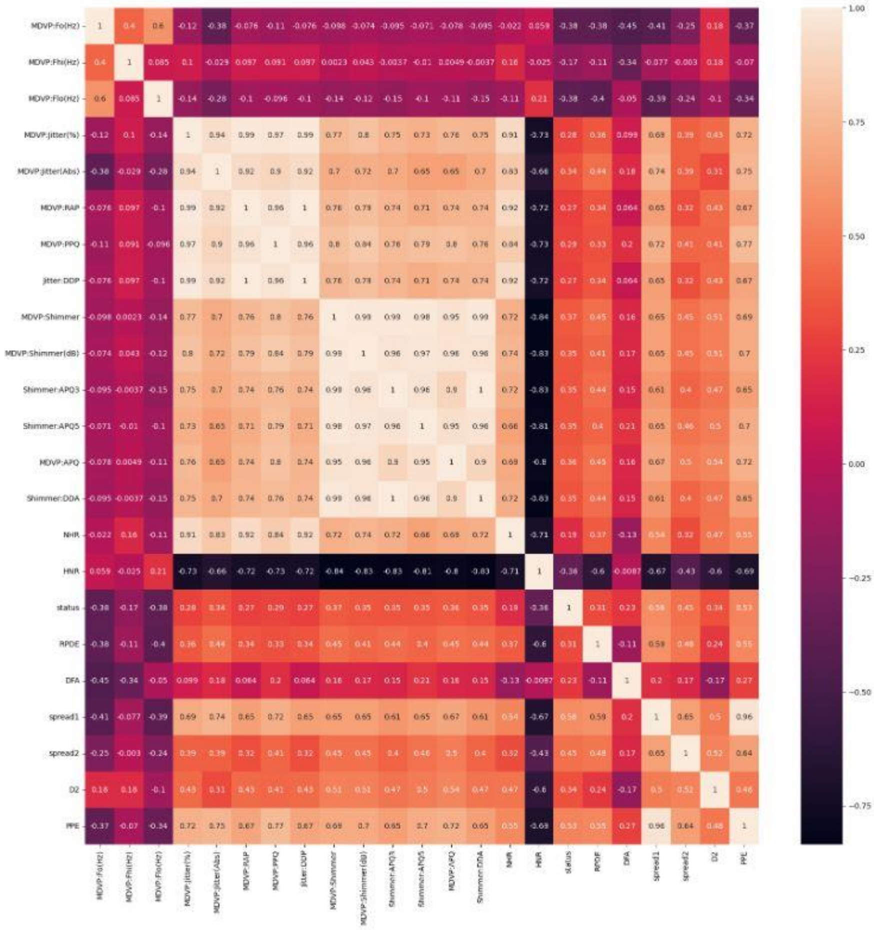


Figure 4.2: Heatmap of the dataset

4.3 OUTLIER DETECTION OF DATA

Machine learning (ML) models rely heavily on outlier identification to find and manage data items that drastically depart from the norm or behave abnormally. Outliers can have a severe influence on the effectiveness and accuracy of ML models, thus spotting and properly handling them is crucial. It's crucial to remember that outlier identification methods should only be used sparingly and in combination with domain expertise. Sometimes outliers can reveal actual abnormalities or brand-new patterns that are pertinent to the current issue. Therefore, deciding whether observed outliers should be viewed as anomalies or important insights requires a thorough comprehension of the data and the context. By treating outliers well and reducing their impact on the model's predictions, suitable outlier detection strategies can help ML models become more resilient, generic, and perform better overall. Potential outliers can be visually identified by visualising the data using techniques like box plots, or histograms. Outliers can be identified by looking at data patterns and identifying them by their extreme values or peculiar distributions. The box plots of the dataset's features are shown below -

- MDVP:Fhi(Hz)

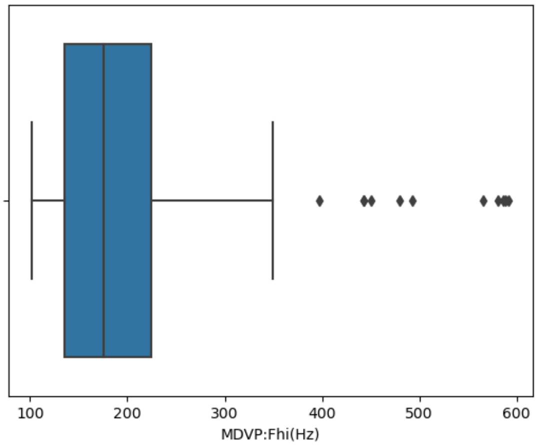


Figure 4.3: MDVP:Fhi(Hz)

- MDVP:Flo(Hz)

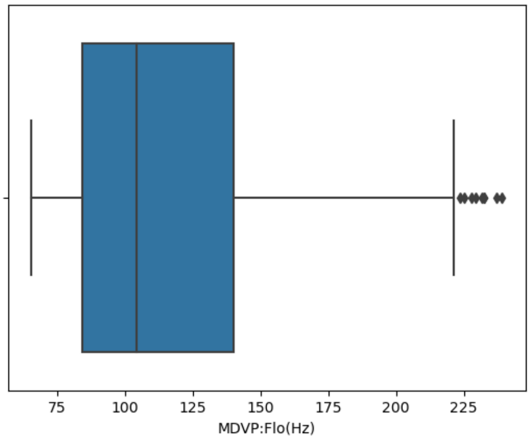


Figure 4.4: MDVP: Flo (Hz)

- MDVP:Jitter(%)

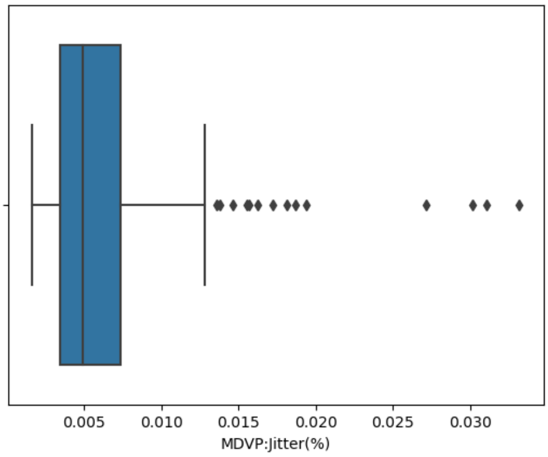


Figure 4.5: MDVP: Jitter(%)

- MDVP:Jitter(Abs)

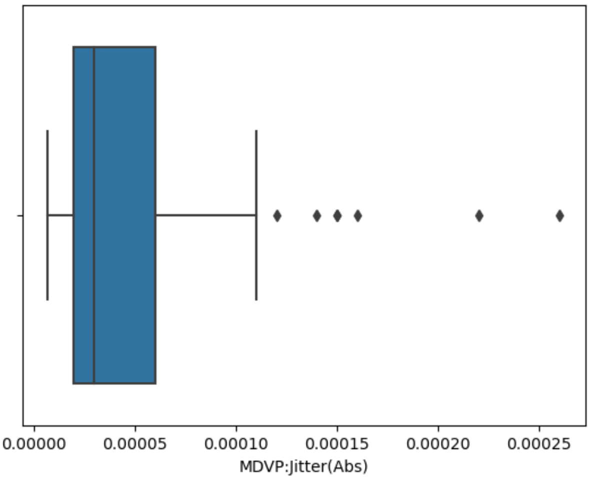


Figure 4.6: MDVP: Jitter(Abs)

- MDVP:RAP

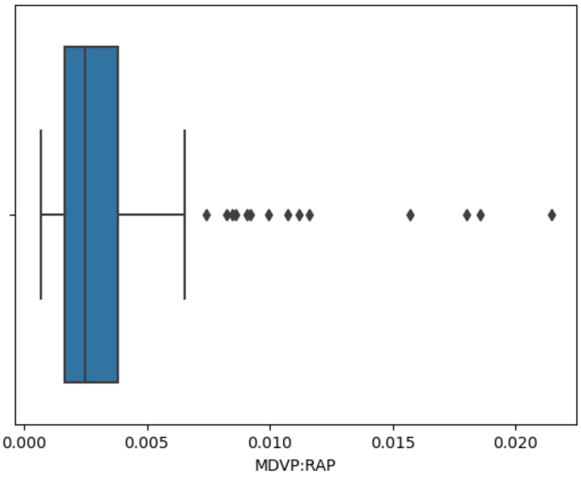


Figure 4.7: MDVP: RAP

- MDVP:PPQ

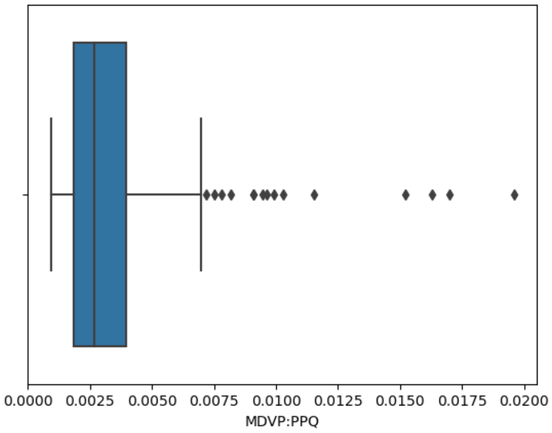


Figure 4.8: MDVP: PPQ

- Jitter:DDP

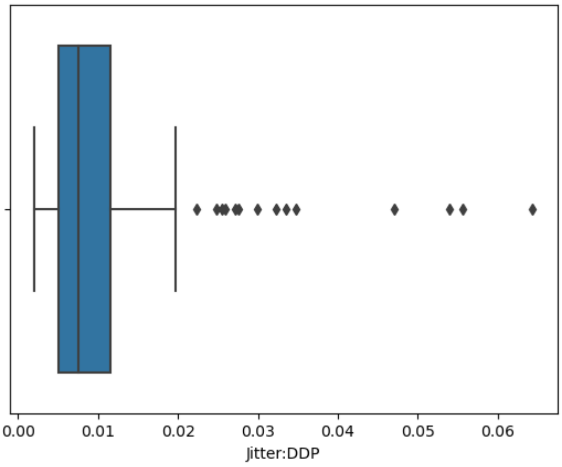


Figure 4.9: Jitter:DDP

- MDVP:Shimmer

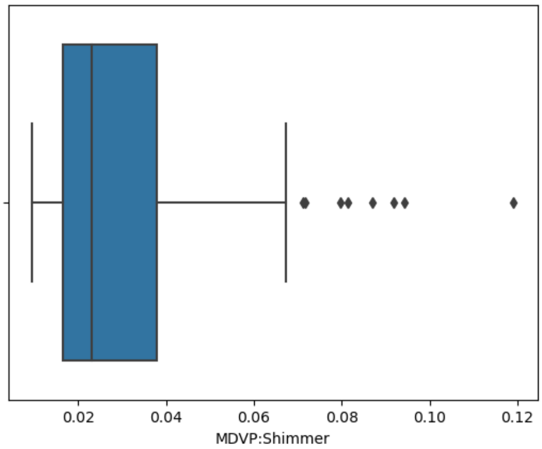


Figure 4.10: MDVP: Shimmer

- MDVP:Shimmer(dB)

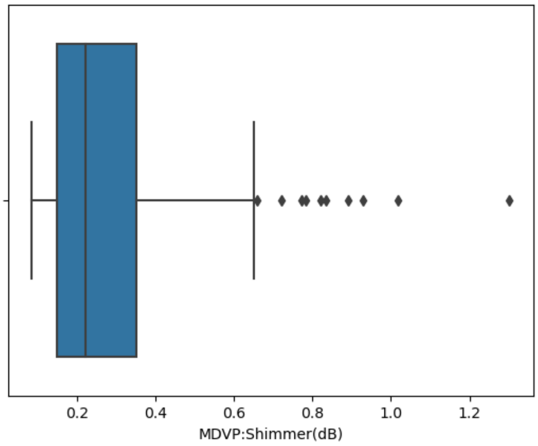


Figure 4.11: MDVP: Shimmer(dB)

- Shimmer:APQ3

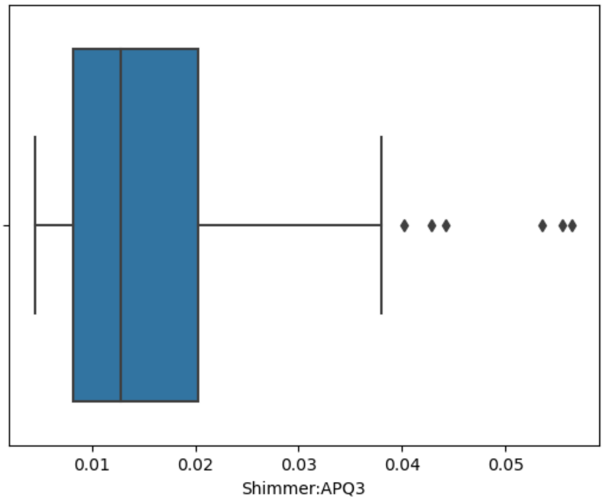


Figure 4.12: Shimmer:APQ3

- Shimmer:APQ5

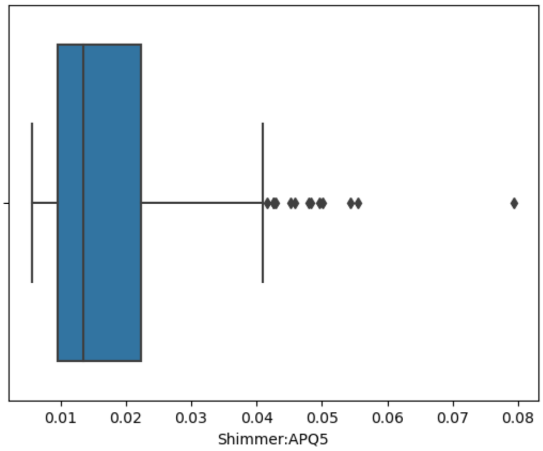


Figure 4.13: Shimmer:APQ5

- MDVP:APQ

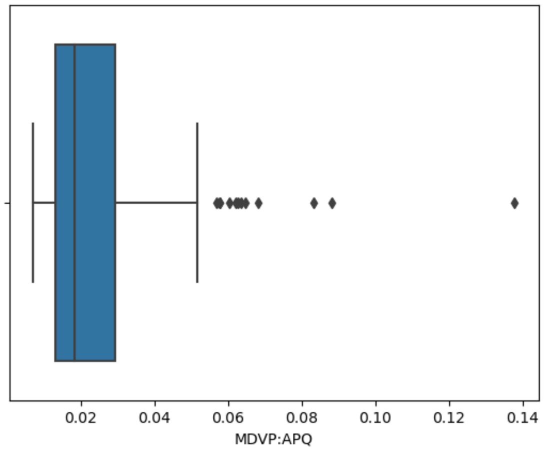


Figure 4.14: MDVP:APQ

- Shimmer:DDA

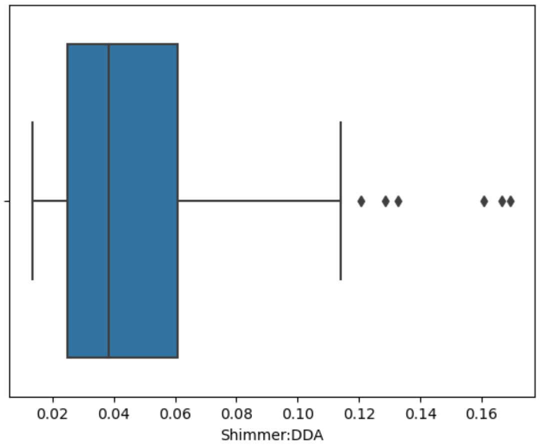


Figure 4.15: Shimmer:DDA

- NHR

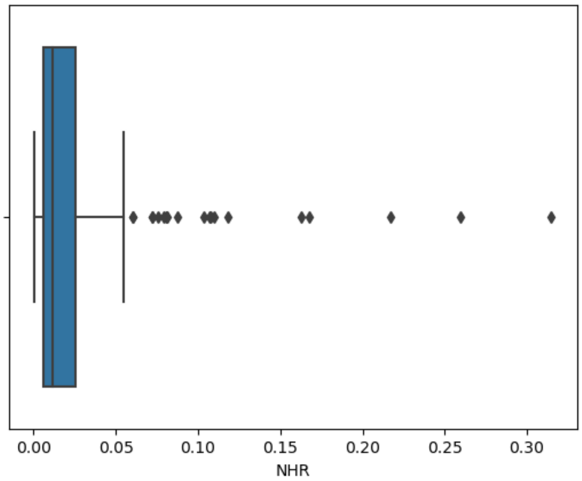


Figure 4.16: NHR

- HNR

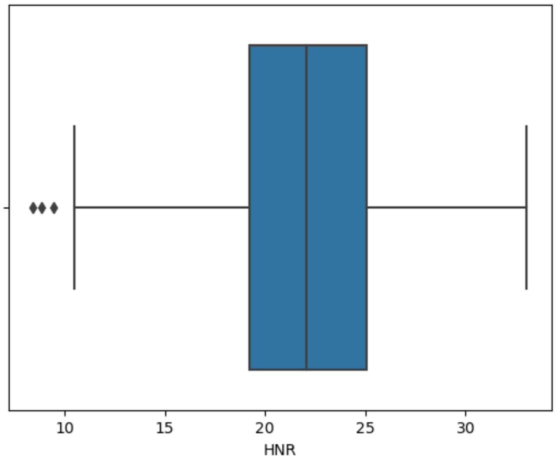


Figure 4.17: HNR

- RPDE

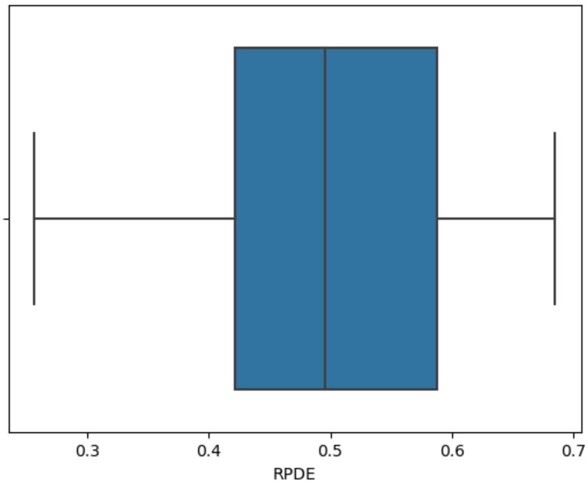


Figure 4.18: RPDE

- DFA

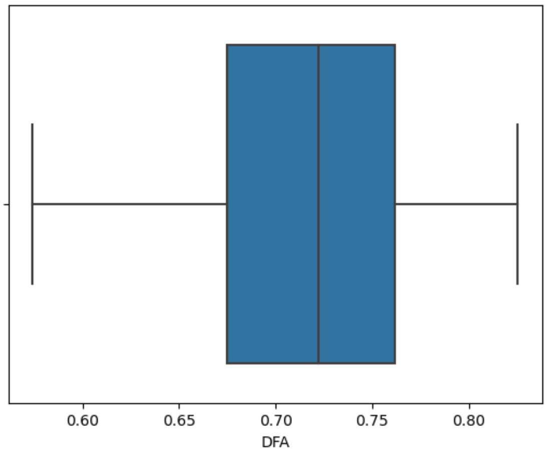


Figure 4.19: DFA

- spread1

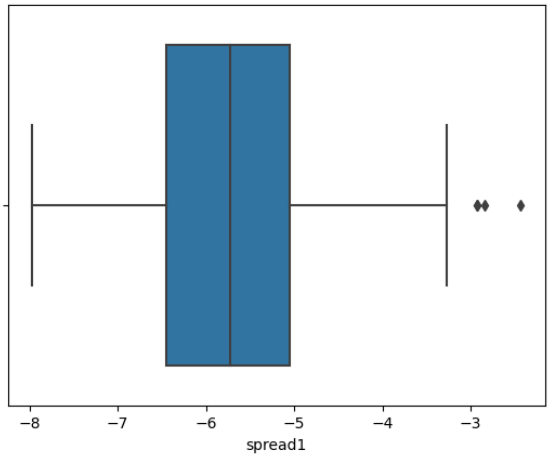


Figure 4.20: Spread1

- spread2

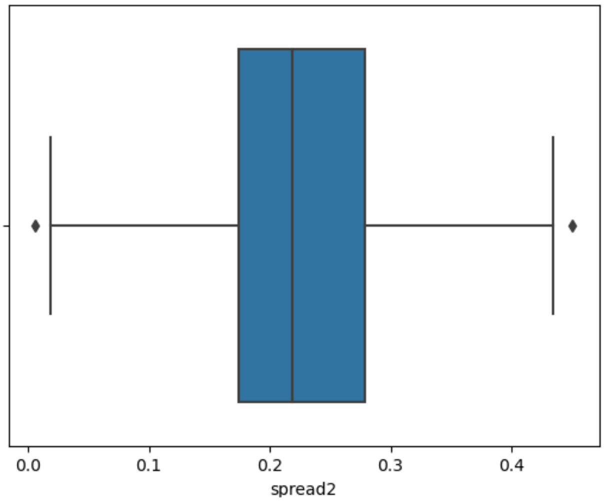


Figure 4.21: Spread2

- PPE

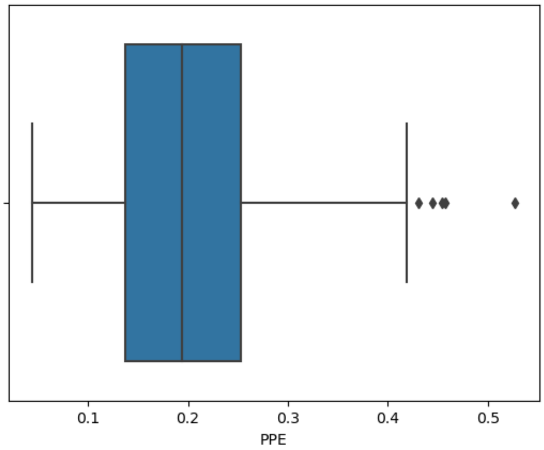


Figure 4.22: PPE

4.4 FUNCTIONAL DESCRIPTION OF MODULES

Machine learning (ML)-based Parkinson's disease (PD) detection relies on a number of interrelated components that work together to provide a reliable and accurate detection method. These crucial elements of PD detection using ML may be used to highlight the function requirements of Parkinson's disease (PD) detection -

- Data Collection: For ML-based PD detection to work, it is essential to collect pertinent data. This entails gathering a variety of information, including clinical details, patient demographics, genetic information, imaging data (MRI, PET scans), sensor data (accelerometers, gyroscopes), and patient-reported results. Performance and dependability of the ML model are significantly influenced by the calibre and volume of data obtained.
- Data preprocessing: Preprocessing operations are required before data is entered into ML algorithms. This entails prepping the data for processing by ML algorithms, addressing missing values, normalising or standardising the data, and cleaning it. Preprocessing guarantees that the data is uniform and suitable for analysis.
- Feature selection and engineering: From the gathered data, the most pertinent and instructive traits that can aid in separating PD from other conditions or forecasting PD-related events are chosen. In order to enhance the performance of the model, new features must be developed or current ones must be modified. To choose and develop important features, this approach needs domain expertise in PD.
- Model Selection: ML models are chosen in accordance with the characteristics of the data, the issue being solved, and the precise objectives of PD detection. Decision trees, random forests, support vector machines (SVM), logistic regression are just a few examples of the many machine learning (ML) methods that may be used. The model of choice need to be capable of handling the peculiarities of the data and producing precise predictions or classifications.
- Model Education: Education The preprocessed data are fed into the chosen algorithm, and its parameters are optimised, allowing the system to learn from the patterns and correlations in the data. PD status or labelled data with established diagnoses are used to train the model. Iteratively modifying the training procedure minimizes errors and improve its predictive performance.
- Model evaluation: The ML model has to be tested using validation datasets that weren't utilised during training after it has been trained. The performance of the model is assessed using evaluation criteria including accuracy, precision, recall, F1-score, and area under the

curve (AUC). The generalizability of the model may be evaluated by using cross-validation methods.

- Classification and Prediction: The ML system ought to let the usage of learned models to generate predictions or categorise fresh, unexplored data. It ought to have features that allow users to enter fresh data and produce probability or forecasts for PD detection. Both binary classification (PD vs. non-PD) and multiclass classification (PD vs. other movement disorders) should be supported by the system.
- Model Interpretability: The machine learning system has to have tools or methods for deciphering and explaining the predictions that the models make. As a result, healthcare practitioners may be better able to trust the model's conclusions and make defensible judgements based on its predictions.
- Testing and deployment: After the ML model has been trained and assessed, it is put to the test on different datasets to determine how well it performs in practical applications. A variety of indicators are used to evaluate the model's performance, and if it satisfies the required standards, it may be used to identify PD in clinical or research contexts. In the deployment phase, the model is integrated into current diagnostic procedures or user-friendly interfaces are created so that healthcare personnel may interact with the model.
- Monitoring and Iteration: In order to maintain accuracy and relevance, ML models for PD detection need to be continuously monitored and periodically evaluated. The model might need to be retrained or updated when new data becomes available in order to include it and retain performance.

These elements cooperate in an iterative process that results in more accurate and reliable PD diagnosis using ML approaches by refining and improving the ML model based on feedback and new data. The creation of an ML-based PD detection system that can process data, train models, make predictions, and integrate into clinical practise or research contexts will be impossible without these function requirements.

4.5 NON-FUNCTIONAL DESCRIPTION OF MODULES

- The qualities or features of a system that specify how it should operate or function are known as non-functional requirements. Some non-functional needs in the context of detecting Parkinson's disease (PD) using machine learning (ML) include -
- Accuracy and Reliability: In order to successfully identify PD, the ML system must exhibit a high level of accuracy and reliability. In order to guarantee that the system produces reliable data that may be utilised for diagnostic or research purposes, it should reduce false positives and false negatives.

- Performance and Efficiency: The machine learning system should be built to handle huge datasets quickly and efficiently. To enable real-time or almost real-time detection, it should be optimised to train models and provide predictions in a fair period of time.
- Scalability: The ML system has to be scalable to manage escalating user demands and data volumes. Larger datasets, more users, and many concurrent requests should all be processed by it without noticeably degrading performance.
- Interoperability: The ML system should provide interoperability with current databases, healthcare systems, or research tools. It should be able to integrate with other programmes or platforms used in the PD detection workflow and import and export data in common formats.
- Security and privacy: The ML system should make sure that patient data is secure and private. It must adhere to all applicable privacy laws and standards, encrypt sensitive data, put access restrictions in place, and guard against unauthorised access or data breaches.
- Usability & User Experience: The ML system has to have an intuitive, user-friendly interface that is simple to use. In order for healthcare practitioners or researchers to properly interact with the system, it must offer clear instructions and feedback.
- Interpretability and Explainability: The ML system should make an effort to provide predictions that can be understood and justified. Users should be able to comprehend the model's logic and make better decisions thanks to the insights it should provide into the characteristics or variables influencing the forecasts.
- Robustness and Error Handling: The machine learning system has to be able to manage erroneous or unexpected input data. It should be equipped with tools for handling errors that can identify and deal with anomalies like missing values or inconsistent data.
- Support and documentation: The ML system must to have thorough documentation outlining its functions, algorithms, and use guidelines. Additionally, it must to offer user assistance or updates to the documentation to help users implement, maintain, and troubleshoot the system.
- Ethical Guidelines and Considerations: The ML system should follow ethical principles. Informed permission, patient privacy, and just and impartial decision-making should all be given top priority in PD detection. The system need to be created and implemented with moral standards in mind.
- These non-functional criteria are crucial to ensuring that the ML-based PD detection system satisfies the specified quality attributes, conforms to user expectations, and responsibly and dependably tackles the unique problems and concerns of PD detection.

4.6 LOGISTIC REGRESSION

For binary classification problems like predicting the existence of Parkinson's disease, machine learning uses the statistical model of logistic regression. Logistic regression is used in this situation to assess the risk of having Parkinson's disease based on patient data. Collecting a dataset containing pertinent characteristics, preprocessing the data, and choosing the most important features are the steps in the process. Healthcare workers may use machine learning to help in the accurate detection of Parkinson's disease by applying logistic regression.

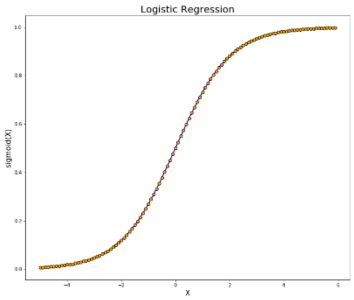


Figure 4.23: Logistic regression

4.7 DECISION TREE

Popular machine learning methods for diagnosing Parkinson's disease include decision trees. Based on input parameters like symptoms and clinical assessments, they construct a model that resembles a tree to predict outcomes. The method creates a tree structure that optimises the division between patients with and without the illness by recursively splitting the data based on useful attributes. Clinicians are able to comprehend the justification for forecasts and recognise crucial traits because to decision trees' interpretability. Decision trees, however, can be vulnerable to overfitting and may have trouble capturing complicated relationships. Nevertheless, when coupled with other methods and bigger datasets, they provide an invaluable tool for Parkinson's disease identification.

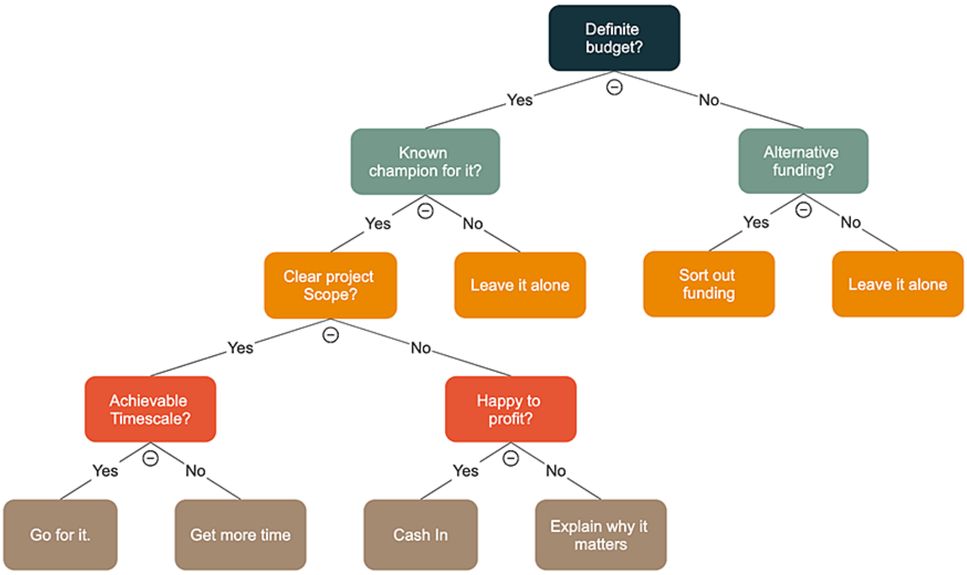


Figure 4.24: Decision tree

4.8 RANDOM FOREST – INFORMATION GAIN

For the identification of Parkinson's disease, the potent machine learning algorithm Random Forest is frequently utilised. In order to create precise forecasts, it makes use of the idea of knowledge gain. Information gain gauges how much uncertainty is reduced as a result of data splitting according to a certain characteristic. The method evaluates the significance of several clinical variables, such as tremor intensity, stiffness, and bradykinesia, in the context of Parkinson's detection by computing their information gain values. Random Forest creates an ensemble of decision trees that collectively forecast the existence or development of Parkinson's disease by choosing the attributes with the largest information gain. This method enhances the precision of the diagnosis and enables efficient monitoring and treatment plans that are personalised to t

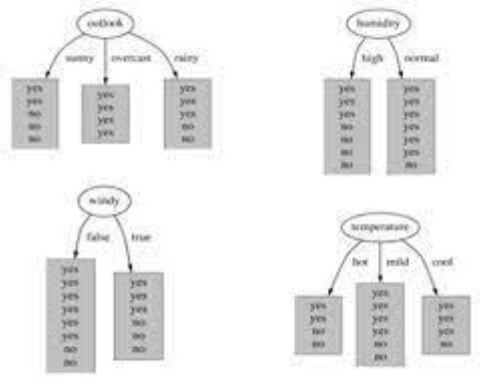


Figure 4.25: Random forest- Information gain

4.9 RANDOM FOREST – ENTROPY

The widely used machine learning algorithm Random Forest makes accurate predictions by using the idea of entropy to diagnose Parkinson's illness. In this application, entropy refers to the measurement of disorder or impurity inside a set of data. The system assesses the entropy of many parameters associated with the illness, including tremor severity, bradykinesia, and stiffness, among others, before building a random forest model for Parkinson's identification. The random forest method can efficiently discover the most useful characteristics for differentiating between healthy persons and those with Parkinson's disease by analysing the entropy of these features. As a result, the model can make wise choices and offer precise predictions for use in diagnosis and therapy.

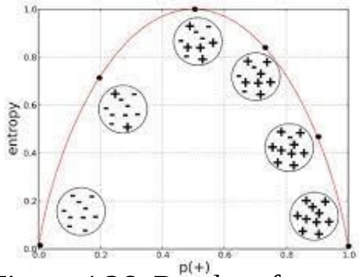


Figure 4.26: Random forest- Entropy

4.10 SUPPORT VECTOR MACHINE

The identification and diagnosis of Parkinson's disease has made substantial use of the potent machine learning technique known as Support Vector Machine (SVM). The SVM supervised learning algorithm separates the data points from several classes with the greatest possible margin by constructing a hyperplane in a high-dimensional feature space. SVM learns to categorise people as either having Parkinson's disease or being healthy in the context of Parkinson's disease detection by using a collection of characteristics derived from patient data, such as speech signals, gait patterns, or motor symptoms. The algorithm employs this model to forecast the illness state of fresh, unseen patients after optimising the hyperplane parameters based on a training dataset.

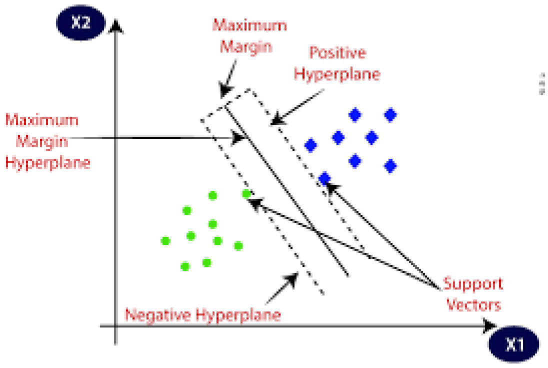


Figure 4.27: Support vector machine

4.11 KNN

Machine learning approaches have been used to identify Parkinson's disease using the K-nearest neighbours (KNN) algorithm. KNN is used in this situation as a classification model to detect and separate people with Parkinson's disease from healthy people. Calculating the distances between an unknown sample and its k closest neighbours in the feature space is how the method operates. Various clinical and demographic information, including age, gender, motor complaints, and neurophysiological measures, are frequently employed in this context as characteristics. The KNN algorithm provides a label to the sample, indicating whether it belongs to the Parkinson's disease or healthy class, by comparing the unknown sample to its closest neighbours.

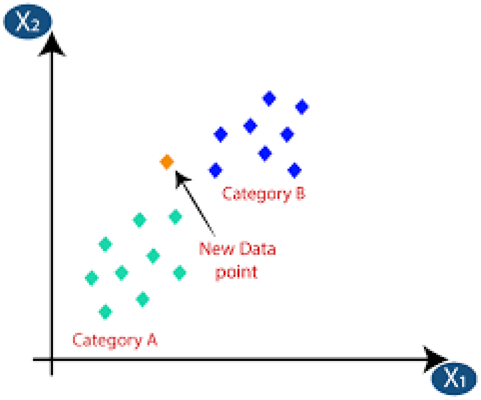


Figure 4.28: KNN

4.12 GAUSSIAN NAÏVE BAYES

An approach for machine learning that is frequently used to identify Parkinson's disease is called Gaussian Naive Bayes. It is predicated on the idea that characteristics are evenly spaced out and independent of one another. Various clinical characteristics, including tremors, bradykinesia, stiffness, and postural instability, are employed as input factors in this situation. By using Bayes' theorem, the method determines the conditional probability that a patient has Parkinson's disease given their feature values. The model learns to estimate the probability distribution of the characteristics for each class by training on a labelled dataset of individuals with and without Parkinson's disease. Then, depending on the feature values of fresh, unknown instances, this trained model may be used to categorise them and determine whether or not a patient has Parkinson's disease.

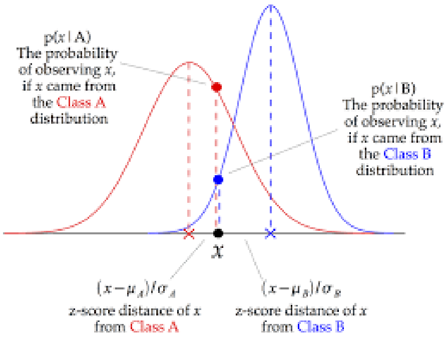

p(x | A)
The probability of observing x, if x came from the Class A distribution
p(x | B)
The probability of observing x, if x came from the Class B distribution
x
$(x-\mu_A)/\sigma_A$
z-score distance of x from Class A
$(x-\mu_B)/\sigma_B$
z-score distance of x from Class B

Figure 4.29: Gaussian naïve bayes

4.13 BERNOULLI NAÏVE BAYES

A popular machine learning method for diagnosing Parkinson's disease is called Bernoulli Naive Bayes. The Naive Bayes method is modified in that it considers features to be binary (boolean) variables. The method uses a series of binary indicators, including tremors, bradykinesia, stiffness, and postural instability, to determine whether Parkinson's disease is present or not. The Bernoulli Naive Bayes algorithm determines the likelihood that a patient has Parkinson's disease based on the observed symptoms by applying Bayes' theorem and assuming independence between the features. By examining binary symptom data, this algorithm has demonstrated potential in properly identifying Parkinson's disease and can offer insightful information for early identification and treatments.

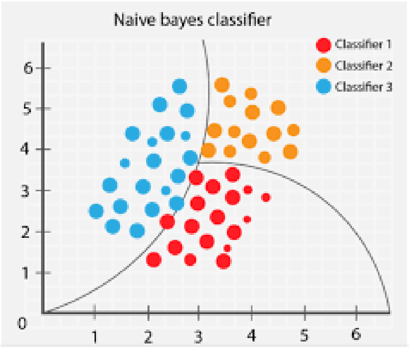


Figure 4.30: Bernoulli naïve bayes

4.14 DATA FLOW DIAGRAMS

4.14.1 DFD Level 0

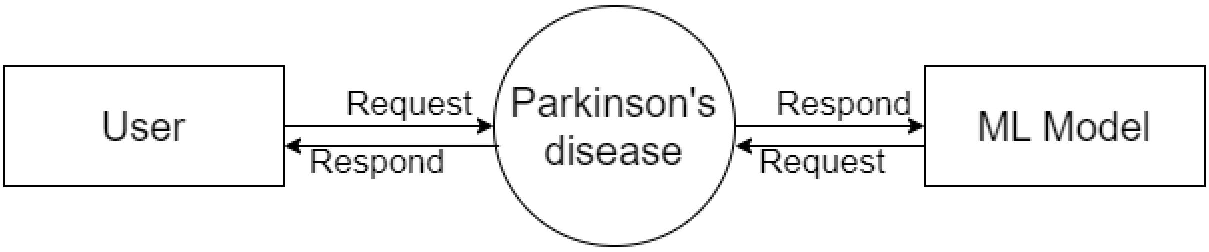


Figure 4.31: DFD level 0

4.14.2 DFD Level 1

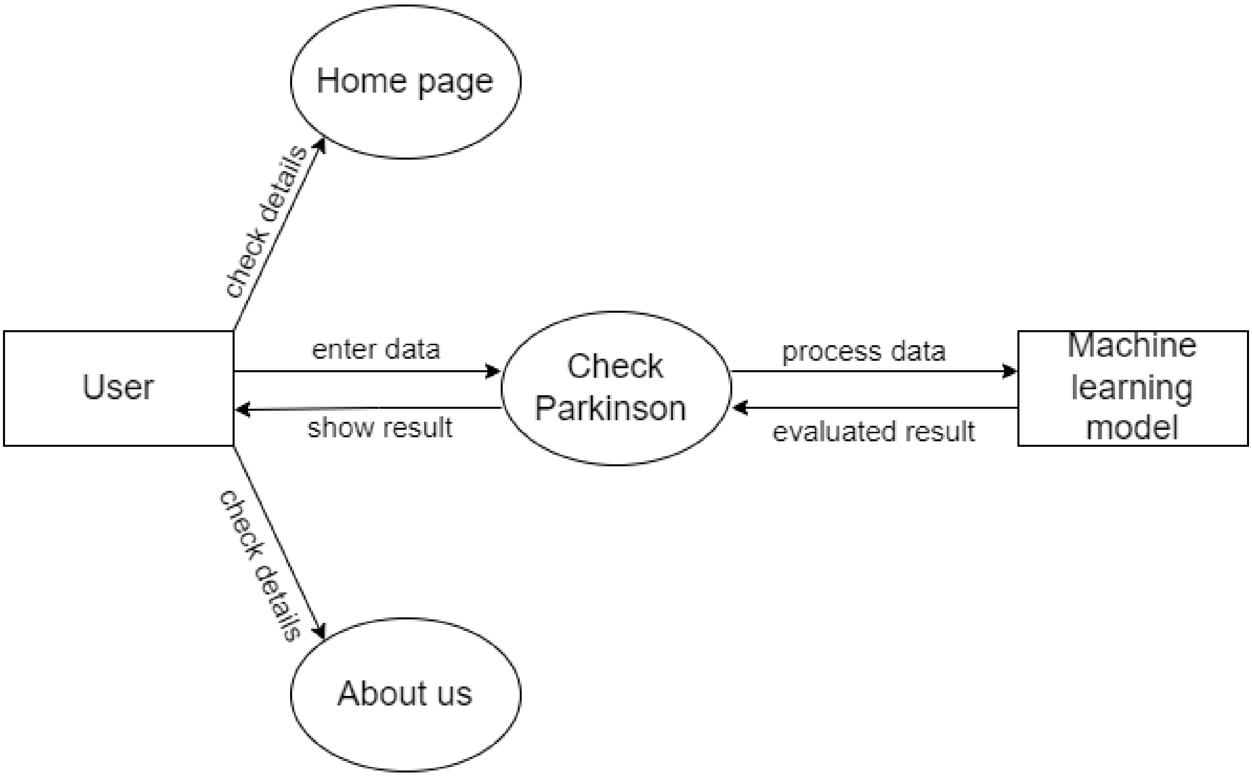


Figure 4.32: DFD level 1

4.14.3 DFD Level 2 of a home page

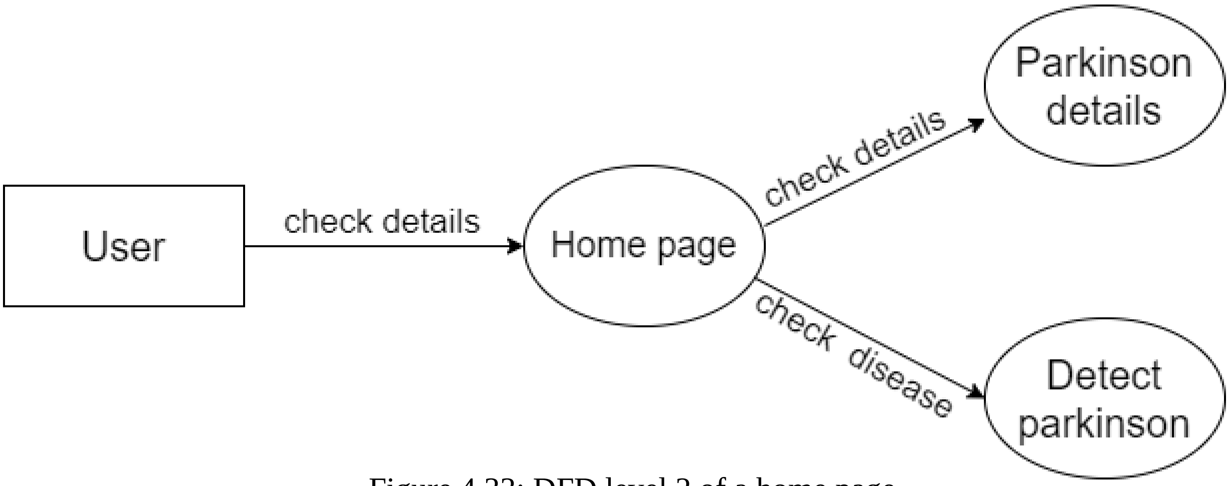


Figure 4.33: DFD level 2 of a home page

4.14.4 DFD Level 2 of about

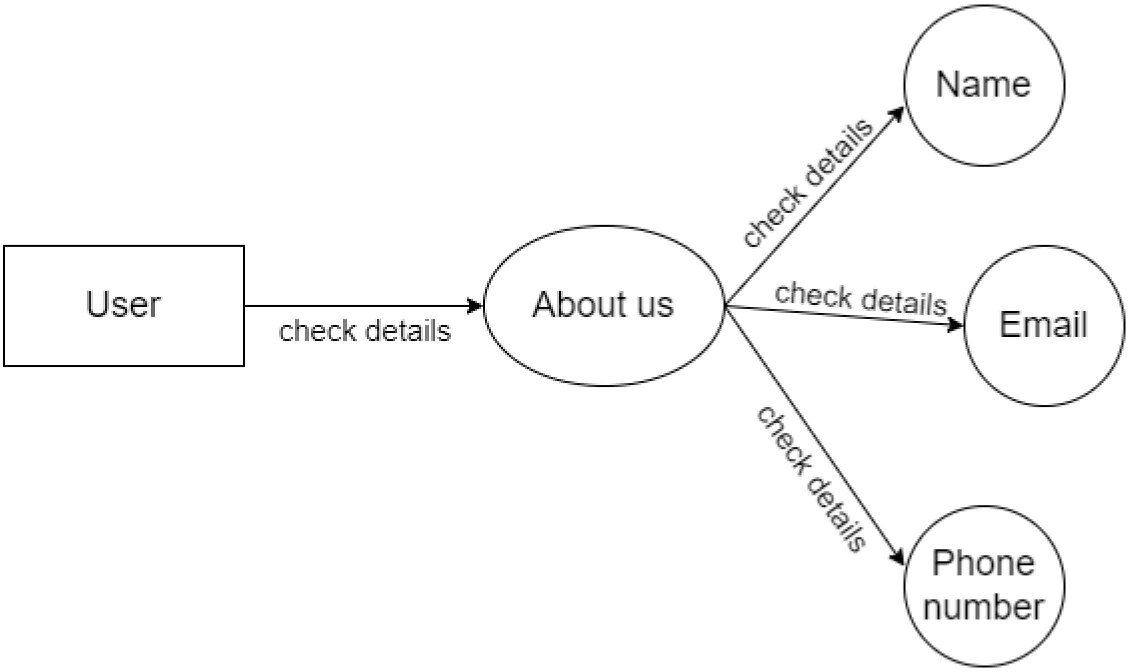


Figure 4.34: DFD level 2 of about

4.15 CLASS DIAGRAM

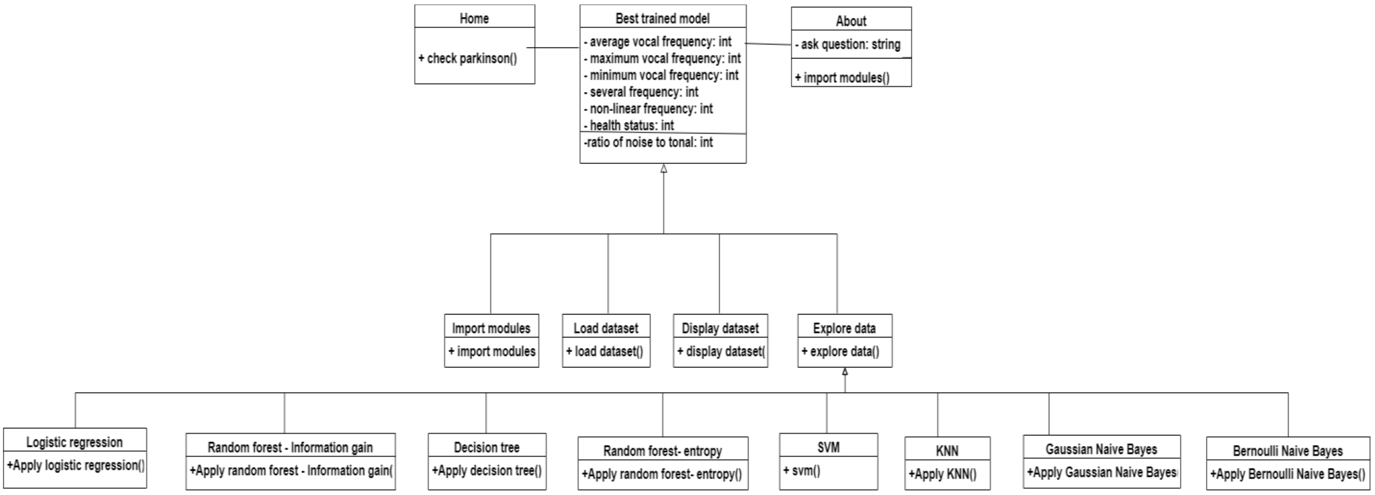


Figure 4.35: Class diagram

4.16 USE CASE DIAGRAM

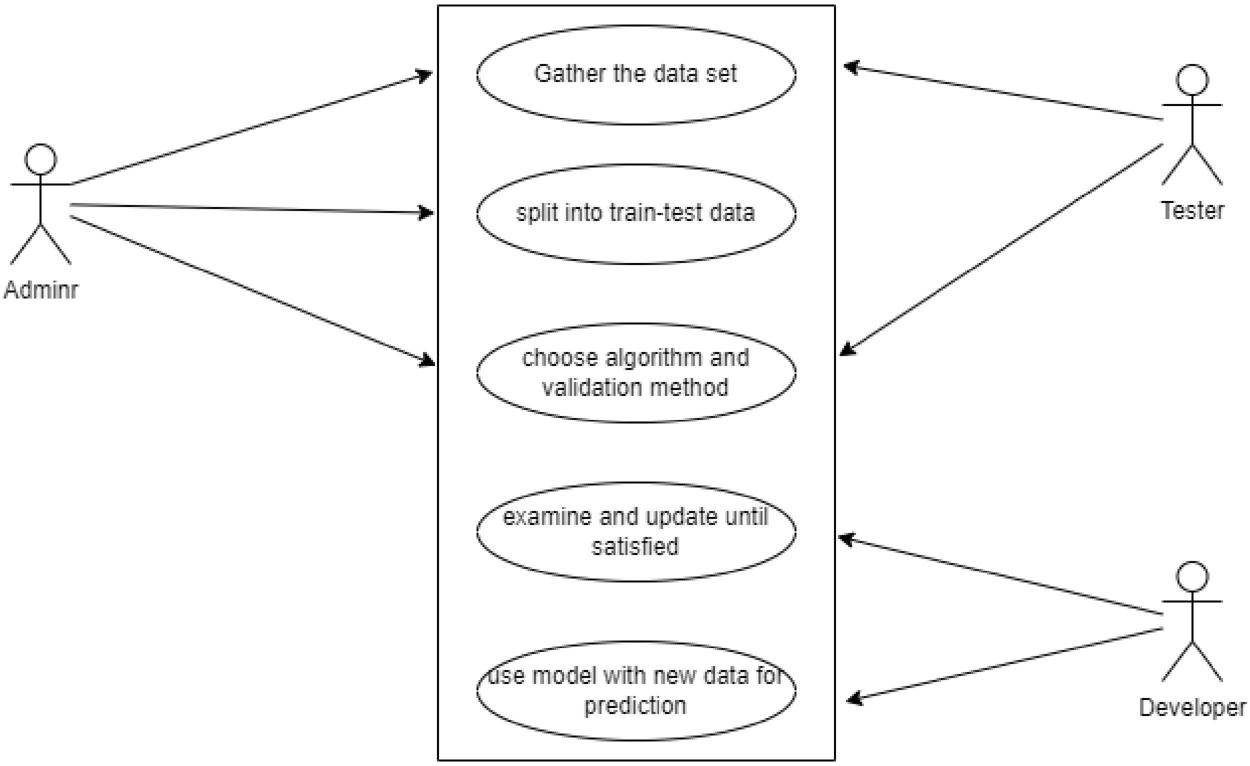

Gather the data set
split into train-test data
choose algorithm and validation method
examine and update until satisfied
use model with new data for prediction
Adminr
Tester
Developer

Figure 4.36: Use case diagram

4.17 PERFORMANCE MEASURES

4.17.1 Precision

Precision is a performance indicator used in machine learning (ML) that assesses how well a model predicts the future. Out of all the positive predictions produced by the model, it calculates the percentage of real positive forecasts. Precision is particularly helpful when the objective is to reduce false positives since it focuses on the accuracy of the model's positive predictions.

The following formula is used to determine the precision:

Precision = True Positives / (True Positives + False Positives)

Depending on the unique application, the costs involved, and the effects of false positives and false negatives, one might choose to prioritise precision over recall. Precision may be more important in some situations, such as medical diagnosis, to prevent false positives that might result in treatments or interventions that are not essential. In other instances, such when determining spam emails, recall may be more important to avoid missing relevant messages.

4.17.2 Recall

Recall is a machine learning (ML) performance indicator that assesses a model's capacity to recognise all pertinent occurrences within a dataset. Out of all real positive examples, it calculates the percentage of true positive forecasts. Recall is very beneficial for reducing false negatives.

The following formula is used to determine the recall:

Recall = True Positives / (True Positives + False Negatives)

When assessing and optimising ML models, recall is a crucial factor to take into account along with other metrics and the particular context of the problem. Precision-recall curves, the F1 score, and other strategies that give a trade-off between the two measures can be used to establish a balance between precision and recall.

4.17.3 F1 Score

The F1 score is a statistic used in machine learning (ML) that combines accuracy and recall into a single indicator of a model's performance. When working with datasets that are unbalanced, where the proportion of positive and negative cases differs noticeably, it is very helpful.

Precision and recall are given equal weight when calculating the F1 score, which is derived as the harmonic mean of both measures. The following equation is used to determine the F1 score:

F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

In terms of accurately recognising positive examples (precision) and recalling all pertinent positive cases (recall), a higher F1 score denotes superior overall performance. The F1 score, which takes both false positives (FP) and false negatives (FN) into account, aids in striking a balance between accuracy and recall. It punishes models with a large number of false positives or false negatives, leading to a lower F1 score.

4.17.4 Support

The number of occurrences or instances of a certain class or category in a dataset is referred to as support in machine learning (ML). The frequency or prevalence of each class within the dataset is typically measured in the context of multi-class classification issues. Support is sometimes calculated as the number or percentage of instances that belong to a certain class. It aids in explaining how facts are distributed across several classes and can serve as a foundation for assessing the relevance or importance of each class. It is important to remember that support alone cannot offer a thorough assessment of a model's success. Usually, it is used with additional measures like F1 score, recall, precision, or recall to gain a more holistic understanding of the model's effectiveness.

4.17.5 Confusion Matrix

A confusion matrix is a table that summarises the effectiveness of a classification model in machine learning (ML) by contrasting predicted labels with actual labels in a dataset. It offers a thorough analysis of the model's forecasts, indicating the proportion of accurate and inaccurate forecasts for each class. The genuine labels of the classes are normally represented by rows, while the anticipated labels of the classes are typically represented by columns, in a confusion matrix. The matrix's components show the number or frequency of events falling into different categories.

The terms TN (True Negative) and FP (False Positive) denote the proportion of instances correctly classified as belonging to the negative class, FN (False Negative) the proportion of instances incorrectly classified as belonging to the negative class, and TP (True Positive) the proportion of instances correctly classified as belonging to the positive class, respectively. Beyond a single assessment indicator, the confusion matrix aids in providing a more thorough insight of a model's performance. It allows for the discovery of particular sorts of model flaws and facilitates additional analysis and parameter optimization for the ML algorithm. Several assessment metrics may be deduced from the confusion matrix like precision, accuracy, recall, and F1 score.

4.17.6 Accuracy

Accuracy is a frequently used performance indicator in machine learning (ML) that assesses the overall correctness of a classification model. It figures out what percentage of a dataset's total occurrences are correctly categorised. The following formula is used to determine accuracy:

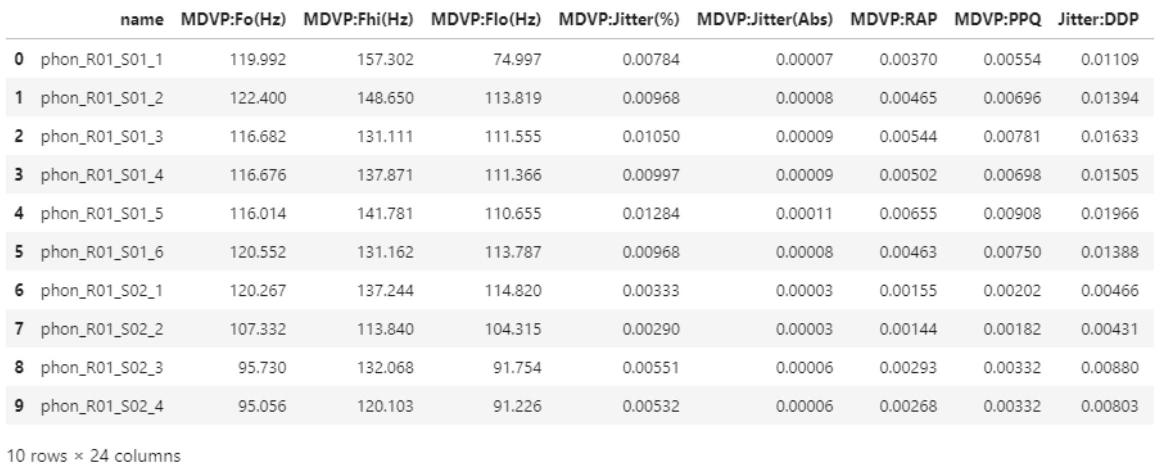
Accuracy = (Number of Correct Predictions) / (Total Number of Predictions)

The accuracy score gives a broad indication of the model's effectiveness. A greater number for accuracy shows that the model is producing more accurate predictions, whereas a lower value for accuracy suggests that the model is producing more inaccurate predictions. It is crucial to remember that accuracy may not always be the best statistic, especially when working with datasets that are unbalanced and have vastly varied numbers of examples in various classes. In certain situations, a high accuracy rating may be deceiving since the model may strongly favour the majority class while underperforming the minority class. A greater number for accuracy shows that the model is producing more accurate predictions, whereas a lower value for accuracy suggests that the model is producing more inaccurate predictions. Precision, recall, F1 score, and area under the receiver operating characteristic curve (ROC-AUC) are important assessment metrics to take into account in order to fully comprehend the performance of the model, especially in skewed or unbalanced datasets.

CHAPTER 5: IMPLEMENTATION AND CODING

5.1 CODING DETAILS

```
#Importing Libraries/Modules
import warnings
warnings.filterwarnings('ignore')
from xgboost import XGBClassifier
import numpy as np
import os , sys
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
#Loading Dataset
parkinsons_data = pd.read_csv(r'C:\Users\sriva\Downloads\parkinsons.csv')
parkinsons_data.head(n=10)
```



	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109
1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01394
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966
5	phon_R01_S01_6	120.552	131.162	113.787	0.00968	0.00008	0.00463	0.00750	0.01388
6	phon_R01_S02_1	120.267	137.244	114.820	0.00333	0.00003	0.00155	0.00202	0.00466
7	phon_R01_S02_2	107.332	113.840	104.315	0.00290	0.00003	0.00144	0.00182	0.00431
8	phon_R01_S02_3	95.730	132.068	91.754	0.00551	0.00006	0.00293	0.00332	0.00880
9	phon_R01_S02_4	95.056	120.103	91.226	0.00532	0.00006	0.00268	0.00332	0.00803

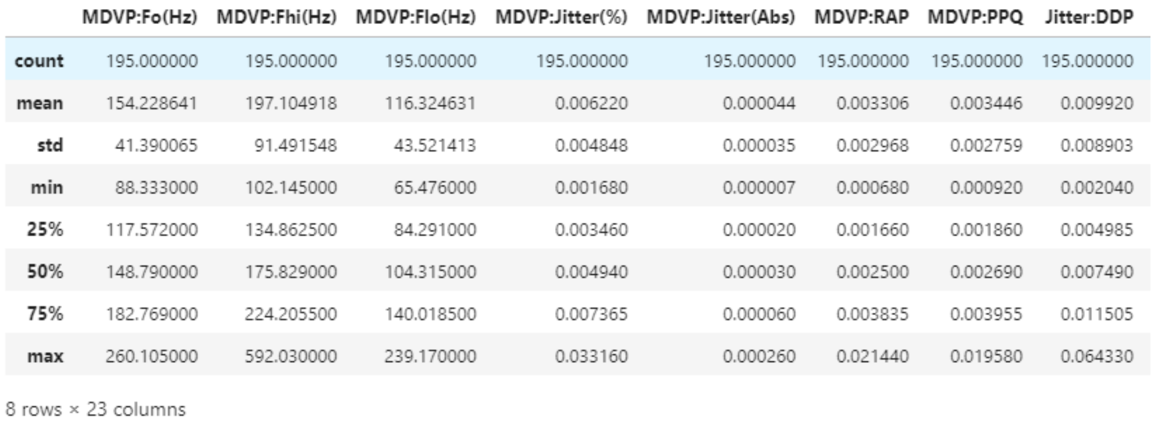
10 rows × 24 columns

Figure 5.1: To know number of rows and columns

```
#To know No of rows and columns
parkinsons_data.shape
(195, 24)
parkinsons_data.info()
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 24 columns):
#   Column           Non-Null Count  Dtype
---  ------           --------------  -----
0   name             195 non-null    object
```

```
1   MDVP:Fo(Hz)       195 non-null    float64
2   MDVP:Fhi(Hz)      195 non-null    float64
3   MDVP:Flo(Hz)      195 non-null    float64
4   MDVP:Jitter(%)    195 non-null    float64
5   MDVP:Jitter(Abs)  195 non-null    float64
6   MDVP:RAP          195 non-null    float64
7   MDVP:PPQ          195 non-null    float64
8   Jitter:DDP        195 non-null    float64
9   MDVP:Shimmer      195 non-null    float64
10  MDVP:Shimmer(dB)  195 non-null    float64
11  Shimmer:APQ3      195 non-null    float64
12  Shimmer:APQ5      195 non-null    float64
13  MDVP:APQ          195 non-null    float64
14  Shimmer:DDA       195 non-null    float64
15  NHR               195 non-null    float64
16  HNR               195 non-null    float64
17  status            195 non-null    int64
18  RPDE              195 non-null    float64
19  DFA               195 non-null    float64
20  spread1           195 non-null    float64
21  spread2           195 non-null    float64
22  D2                195 non-null    float64
23  PPE               195 non-null    float64
dtypes: float64(22), int64(1), object(1)
memory usage: 36.7+ KB
```

```
#Statistical Measures about Data
parkinsons_data.describe()
```



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

8 rows × 23 columns

Figure 5.2: Checking for null values

```
#Checking for null values
parkinsons_data.isnull().sum()
Button(description='Toggle Pandas/Lux', layout=Layout(top='5px', width='140px'),
style=ButtonStyle())
Output()
parkinsons_data.dtypes
name                object
MDVP:Fo(Hz)         float64
MDVP:Fhi(Hz)        float64
MDVP:Flo(Hz)        float64
MDVP:Jitter(%)      float64
MDVP:Jitter(Abs)    float64
MDVP:RAP            float64
MDVP:PPQ            float64
Jitter:DDP          float64
MDVP:Shimmer        float64
MDVP:Shimmer(dB)    float64
Shimmer:APQ3        float64
Shimmer:APQ5        float64
MDVP:APQ            float64
Shimmer:DDA         float64
NHR                 float64
HNR                 float64
status              int64
RPDE                float64
```

```
DFA             float64
spread1         float64
spread2         float64
D2              float64
PPE             float64
dtype: object
#Finding Unique Values
for i in parkinsons_data.columns:
print("*********************************",i,"******************************")
print()
print(set(parkinsons_data[i].tolist()))
print()
#Checking Label Imbalance
import matplotlib.pyplot as plt
import seaborn as sns
temp=parkinsons_data["status"].value_counts()
temp_df=pd.DataFrame({'status' : temp.index, 'values' : temp.values})
print(sns.barplot(x='status', y='values' , data=temp_df))
AxesSubplot(0.125,0.11;0.775x0.77)
```

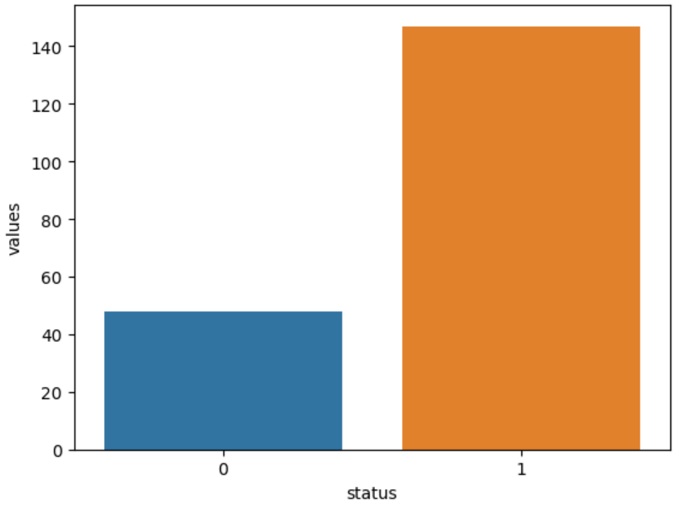


Figure 5.3: Checking Label Imbalance

```
sns.pairplot(parkinsons_data)
<seaborn.axisgrid.PairGrid at 0x1f55d42b520>
```

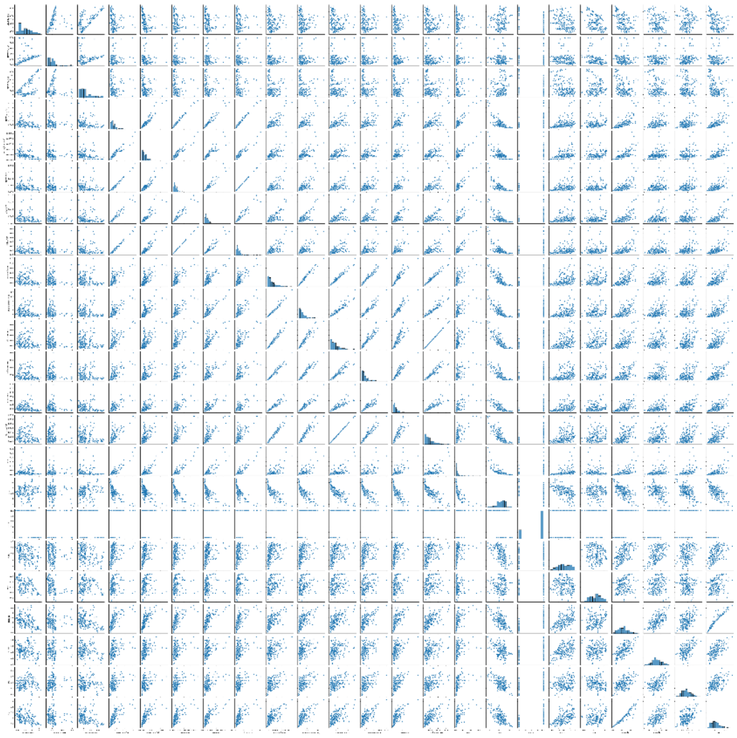


Figure 5.4: Finding distribution of data

```
#Finding distribution of data
def distplots(col):
sns.distplot(parkinsons_data[col])
plt.show()
for i in list(parkinsons_data.columns)[1:]:
distplots(i)
#Finding distribution of data for extreme values
def boxplots(col):
sns.boxplot(parkinsons_data[col])
plt.show()
for i in list(parkinsons_data.select_dtypes(exclude=["object"]).columns)[1:]:
boxplots(i)
#Finding correlation
plt.figure(figsize=(20,20))
corr=parkinsons_data.corr()
sns.heatmap(corr,annot=True)
<AxesSubplot:>
#Seperating Dependent and Independent variables and dropping ID Column
x = parkinsons_data.drop(columns=['name','status'], axis=1) #Drops the mentioned columns
#axis1 for column
```

```
y = parkinsons_data['status']
#Detecting Label Balance
from imblearn.over_sampling import RandomOverSampler
from imblearn.under_sampling import RandomUnderSampler
from collections import Counter
print(Counter(y))
Counter({1: 147, 0: 48})
#Balancing the labels
ros = RandomOverSampler()
X_ros , y_ros = ros.fit_resample(x,y)
print(Counter(y_ros))
Counter({1: 147, 0: 147})
#Scaling - avoids overfit of data
scaler=MinMaxScaler((-1,1))
x=scaler.fit_transform(X_ros)
y=y_ros
#Principle Account Analyser
#Fixed/Retained variance
from sklearn.decomposition import PCA
pca = PCA (.95)
X_PCA=pca.fit_transform(x)
print(x.shape)
print(X_PCA.shape)
(294, 22)
(294, 8)
#Split into test and training data from dataset
x_train, x_test, y_train, y_test = train_test_split(X_PCA, y, test_size=0.2, random_state=7) #0.2 i.e 20%
#20% test data
#80% training data
from sklearn.metrics import confusion_matrix , accuracy_score , f1_score , precision_score , recall_score
list_met=[]
list_accuracy=[]
#Applying Algorithms
```

```
#-------------------------------------(1)*Logistic Regression-----------------------------------------------
--
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(C=0.4,max_iter=1000,solver='liblinear')
lr=classifier.fit(x_train,y_train)
#Prediction
y_pred=classifier.predict(x_test)
#Accuracy
accuracy_LR=accuracy_score(y_test,y_pred)

#----------------------------------------(2)*Decision Tree-------------------------------------------------
from sklearn.tree import DecisionTreeClassifier
classifier2=DecisionTreeClassifier(random_state=14)
dt=classifier2.fit(x_train,y_train)
#Prediction
y_pred2=classifier2.predict(x_test)
#Accuracy
accuracy_DT=accuracy_score(y_test,y_pred2)

#-------------------------------------(3)*Random Forest-Information Gain--------------------------------
from sklearn.ensemble import RandomForestClassifier
classifier3=RandomForestClassifier(random_state=14)
rfi=classifier3.fit(x_train,y_train)
#Prediction
y_pred3=classifier3.predict(x_test)
#Accuracy
accuracy_RFI=accuracy_score(y_test,y_pred3)

#-------------------------------------(4)*Random Forest-Entropy----------------------------------------
from sklearn.ensemble import RandomForestClassifier
classifier4=RandomForestClassifier(criterion='entropy')
rfe=classifier4.fit(x_train,y_train)
#Prediction
y_pred4=classifier4.predict(x_test)
#Accuracy
```

```
accuracy_RFE=accuracy_score(y_test,y_pred4)

#-------------------------------------(5)*Support Vector Machine---------------------------------------
from sklearn.svm import SVC
model_svm=SVC(cache_size=100)
svm=model_svm.fit(x_train,y_train)
#Prediction
y_pred5=model_svm.predict(x_test)
#Accuracy
accuracy_svc=accuracy_score(y_test,y_pred5)

#----------------------------------------(6)*KNN------------------------------------------------
from sklearn.neighbors import KNeighborsClassifier
model_knn3=KNeighborsClassifier(n_neighbors=3)
knn=model_knn3.fit(x_train,y_train)
#Prediction
pred_knn3=model_knn3.predict(x_test)
#Accuracy
accuracy_SVM=accuracy_score(y_test,pred_knn3)

#-------------------------------------(7)*Gaussian Naive Bayes-------------------------------------
from sklearn.naive_bayes import GaussianNB
gnb=GaussianNB()
gnb=gnb.fit(x_train,y_train)
#Prediction
pred_gnb=gnb.predict(x_test)
#Accuracy
accuracy_GNB=accuracy_score(y_test,pred_gnb)




#-------------------------------------(8)*Bernoulli Naive Bayes-------------------------------------
from sklearn.naive_bayes import BernoulliNB
model=BernoulliNB()
bnb=model.fit(x_train,y_train)
```

```
#Prediction
pred_bnb=model.predict(x_test)
#Accuracy
accuracy_BNB=accuracy_score(y_test,pred_bnb)

#----------------------------Combining all using Voting Classifier--------------------------------
from sklearn.ensemble import VotingClassifier
evc=VotingClassifier(estimators=[('lr',lr),('rfi',rfi),('rfe',rfe),('DT',dt),('svm',svm),('knn',knn),('gnb
',gnb),
('bnb',bnb)], voting='hard', flatten_transform=True)
model_evc=evc.fit(x_train,y_train)
#Predicting Test Sets
pred_evc = evc.predict(x_test)
#Accuracy
accuracy_evc = accuracy_score(y_test,pred_gnb)
list1=['Logistic Regression' , 'Decision Tree' , 'Random Forest-Information Gain' , 'Random
Forest-Entropy' ,
'Support Vector Machine' , 'KNN' , 'Gaussian Naive Bayes' , 'Bernoulli Naive Bayes']
list2=[accuracy_LR , accuracy_DT , accuracy_RFI , accuracy_RFE , accuracy_svc
,accuracy_SVM ,  accuracy_GNB ,
accuracy_BNB]
list3=[classifier ,classifier2 , classifier3 , classifier4 , model_svm , model_knn3 , gnb , model]
parkinsons_data_Accuracy=pd.DataFrame({'Algorithm':list1 ,'Accuracy':list2})
print(parkinsons_data_Accuracy)
chart=sns.barplot(x='Algorithm' , y='Accuracy' , data=parkinsons_data_Accuracy)
chart.set_xticklabels(chart.get_xticklabels(), rotation=90)
print(chart)
```

```
Algorithm Accuracy
0 Logistic Regression 0.813559
1 Decision Tree 0.932203
2 Random Forest-Information Gain 1.000000
3 Random Forest-Entropy 1.000000
4 Support Vector Machine 0.932203
5 KNN 0.966102
6 Gaussian Naive Bayes 0.864407
```

```
7 Bernoulli Naive Bayes 0.847458
AxesSubplot(0.125,0.11;0.775x0.77)
#XGBoostClassifier
model_xg=XGBClassifier()
model_xg.fit(x_train,y_train)
XGBClassifier(base_score=None, booster=None, callbacks=None,
colsample_bylevel=None, colsample_bynode=None,
colsample_bytree=None, early_stopping_rounds=None,
enable_categorical=False, eval_metric=None, feature_types=None,
gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
interaction_constraints=None, learning_rate=None, max_bin=None,
max_cat_threshold=None, max_cat_to_onehot=None,
max_delta_step=None, max_depth=None, max_leaves=None,
min_child_weight=None, missing=nan, monotone_constraints=None,
n_estimators=100, n_jobs=None, num_parallel_tree=None,
predictor=None, random_state=None, ...)
#Final Model Accuracy
y_pred=model_xg.predict(x_test)
print(accuracy_score(y_test , y_pred)*100)
100.0
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test , model_xg.predict(x_test))
from sklearn.metrics import f1_score
f1_score(y_test , model_xg.predict(x_test),average='binary')
1.0
from sklearn.metrics import roc_curve , auc , confusion_matrix , classification_report ,
accuracy_score
print(classification_report(y_test , model_xg.predict(x_test)))
print('confusion matrix')
print(cm)
precision    recall  f1-score   support
0       1.00      1.00      1.00        24
1       1.00      1.00      1.00        35
accuracy                    1.00        59
macro avg       1.00      1.00      1.00        59
```

```
weighted avg       1.00      1.00      1.00        59
confusion matrix
[[24  0]
[ 0 35]]
```

```
for i in list3:
print("********" , i , "********")
print(classification_report(y_test , i.predict(x_test)))
print('confusion matrix')
print(cm)
print()
```

CHAPTER 6: SOFTWARE TESTING

6.1 TESTING APPROACH

In machine learning, a model is put to the test to judge how well it performs and how well it can predict outcomes from new data. The following steps are frequently included in the testing process:

- **Data Split:** The given dataset is split into two or three subsets: the training set, the optional validation set, and the test set. The validation set is used to fine-tune hyperparameters and model selection (if necessary), and the test set is saved for the final assessment. The training set is utilised to train the model.
- **Feature pre-processing:** Data in the test set is preprocessed using features in a similar way to the training set. This might entail actions like scaling, addressing missing values, normalisation, or feature engineering. To preserve consistency, it's crucial to employ the same preprocessing procedures as during training.
- **Model Prediction:** The trained ML model receives the preprocessed test data and uses it to make predictions based on the correlations and patterns it has discovered in the training data. The predictions of the model may take the form of continuous values (such as regression) or class labels (such as binary or multi-class classification).
- **Evaluation Metrics**: The model's anticipated outputs are compared to the test set's ground truth labels or values. Depending on the nature of problem, several assessment measures are employed. Metrics including accuracy, precision, recall, F1 score, and area under the ROC curve (AUC-ROC) are frequently employed for classification tasks. Metrics like mean squared error (MSE), mean absolute error (MAE), or R-squared value can be used for regression jobs.
- **Performance Evaluation**: The model's performance is shown by the evaluation metrics derived from comparing the model predictions with the ground truth labels. It aids in assessing the model's ability to generalise to new data. By computing the evaluation metrics, it is possible to do a quantitative analysis of the performance. A qualitative analysis involves visualising the predictions using graphs, charts, or confusion matrices.
- **Iterative Refinement:** Refinement through iteration may be necessary if the model's performance is unsatisfactory. This may entail changing hyperparameters, choosing various features, experimenting with different architectures or methods, or accumulating more training data. After that, the model is iteratively retrained and tested until the target performance is attained.

6.1.1 Scikit-learn

Scikit-learn, sometimes referred to as sklearn, is a well-liked Python machine learning package that is open-source. For a variety of activities, including data preparation, feature selection, model training, and assessment, it offers a comprehensive range of tools and features. The diagnosis and analysis of Parkinson's disease are only two of the many machine learning issues that Sklearn's vast library of algorithms and methodologies may be used for. There are several approaches to apply sklearn in the context of Parkinson's disease:

- Data preparation: Before training a model, data may be handled using a variety of preprocessing techniques offered by Sklearn. This covers normalising data, scaling features, and managing missing values and categorical variables. To ensure data quality and prepare it for use as input in machine learning models, several preprocessing processes are essential.
- Sklearn provides feature selection techniques to find the most pertinent characteristics for Parkinson's disease identification. These methods assist in decreasing dimensionality, enhancing model effectiveness, and maybe increasing prediction accuracy. For instance, two well-liked feature selection techniques offered by sklearn are Recursive Feature Elimination (RFE) and SelectKBest.
- Model Training: A variety of machine learning methods are supported by Sklearn and can be used to diagnose Parkinson's disease. These methods include logistic regression, support vector machines (SVM), decision trees, random forests, and more. Using the provided data, Sklearn offers a simple and consistent interface for training these models.
- Sklearn offers a variety of assessment metrics and methods to judge the effectiveness of machine learning models. Sklearn may be used to compute measures for Parkinson's disease identification, including accuracy, precision, recall, F1 score, and AUC-ROC. Sklearn also provides cross-validation methods, including k-fold cross-validation, to measure the model's performance on several data subsets and lessen the effects of overfitting.
- Finding the ideal mix of hyperparameters for a particular model includes using the tools that Sklearn offers for hyperparameter optimisation. Two well-liked methods in Sklearn for doing an exhaustive or random search through a predetermined set of hyperparameters to find the ideal configuration are GridSearchCV and RandomizedSearchCV.
- Researchers and practitioners may quickly create and assess machine learning models for the identification of Parkinson's disease by utilising the features of sklearn. The library is an invaluable resource in the field of machine learning for medical applications like

Parkinson's disease analysis due to its simplicity of use, thorough documentation, and large community support.

6.1.2 Train_test_split

The train_test_split function in scikit-learn (sklearn) is a helpful tool for dividing a dataset into training and testing groups. When developing models to diagnose Parkinson's disease, this function is frequently used to evaluate the effectiveness and generalizability of the trained model on new data. The use of the train_test_split function is explained in detail below:

You may randomly split a dataset into two or more subgroups depending on a chosen test size or proportion using the train_test_split function in Sklearn. By dividing the data into two subsets, the model is trained on one and assessed on the other, allowing for the estimation of the model's performance on unobserved data. The dataset can be divided up by the function into training and test sets, or even training, validation, and test sets.

Here's an example of how to use the train_test_split function in the context of Parkinson's disease detection:

```
    from sklearn.model_selection import train_test_split
    # X is the feature matrix and y is the target variable
    X, y = load_parkinsons_data()
    # Splitting the dataset into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
  random_state=42)
```

In the above code, X represents the feature matrix, which contains the input variables or features used for Parkinson's disease detection. y represents the corresponding target variable or labels indicating the presence or absence of the disease.

The function train_test_split accepts a number of inputs. The feature matrix and the target variable, respectively, are represented by the first two inputs X and y. The percentage of the dataset that should be allotted to the testing set is specified by the test_size argument. It is set to 0.2 in this example, which indicates that 20% of the data will be utilised for testing. By fixing the random seed, the random_state parameter assures repeatability.

The four subsets that the function returns are X_train, X_test, Y_train, and Y_test. The model is trained using the X_train and y_train subsets, and its performance is assessed using the X_test and y_test subsets. The model is tested on X_test with the associated ground truth labels y_test after being trained on X_train and y_train. This enables us to determine multiple assessment criteria, like accuracy, precision, recall, or any other pertinent statistic, to evaluate the model's effectiveness on unobserved data.

We may replicate real-world circumstances when the model meets fresh, unexplored data during deployment by separating the data into distinct training and testing sets using train_test_split. This aids in evaluating the model's capacity for generalisation and prediction accuracy using previously unreported Parkinson's disease data.

6.2 UNIT TESTING

Table 6.1 – Unit Testing

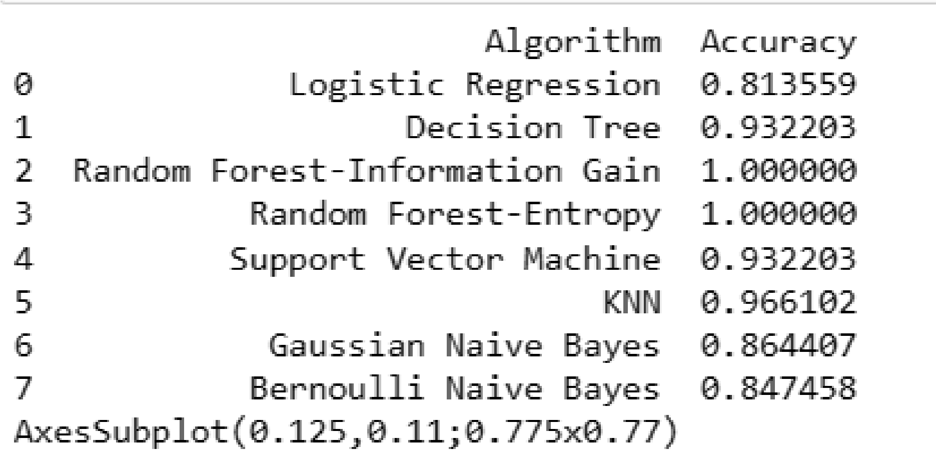
S.NO	Module	Testing
1	Data Loading	Successful!
2	Data Training	Successful!
3	Logistic Regression	Successful!
4	Decision Tree	Successful!
5	Random Forest – Information Gain	Successful!
6	Random Forest – Entropy	Successful!
7	SVM	Successful!
8	KNN	Successful!
9	Gaussian Naïve Bayes	Successful!
10	Bernoulli Naïve Bayes	Successful!
11	Model Training	Successful!
12	Accuracy and Cross Validation	Successful!

6.3 MODIFICATIONS AND IMPROVEMENTS

- Apply a more sophisticated way of detecting Parkinson’s disease
- Increase the quantity of training by incorporating more complicated and diverse datasets.
- To correctly identify and categories the disease, use more algorithms
- To increase the classification's precision, combine various feature extraction techniques and feature engineering techniques.
- To expand the quantity of training instances, use data augmentation approaches.

CHAPTER 7: RESULT AND DISCUSSION

7.1 SNAPSHOTS OF MODEL



```
                           Algorithm  Accuracy
0                Logistic Regression  0.813559
1                      Decision Tree  0.932203
2  Random Forest-Information Gain  1.000000
3               Random Forest-Entropy  1.000000
4             Support Vector Machine  0.932203
5                                KNN  0.966102
6               Gaussian Naive Bayes  0.864407
7              Bernoulli Naive Bayes  0.847458
AxesSubplot(0.125,0.11;0.775x0.77)
```

Figure 7.1: Algorithm and Accuracy

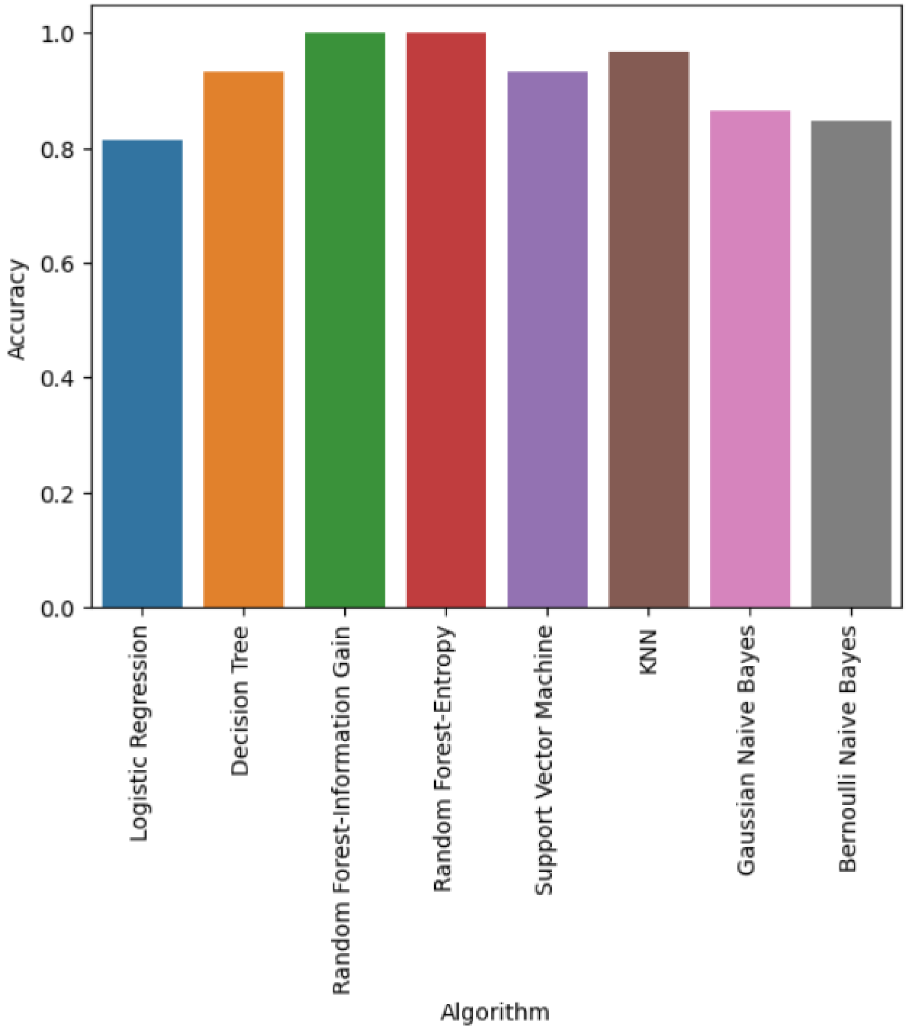
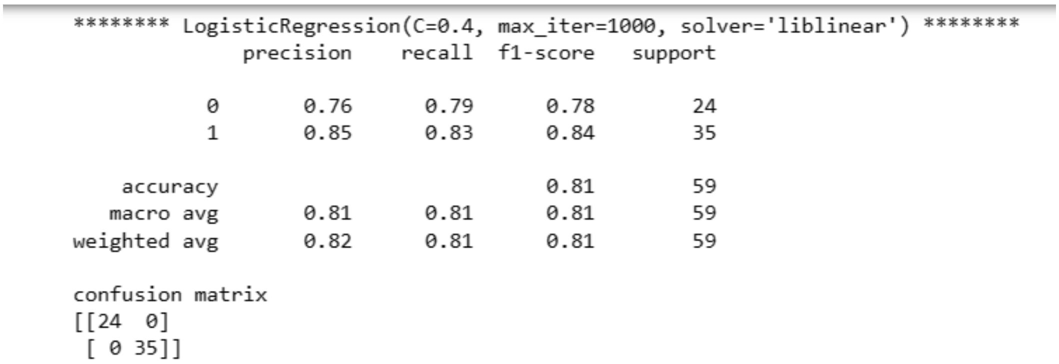


Figure 7.2: Chart of Algorithm and Accuracy

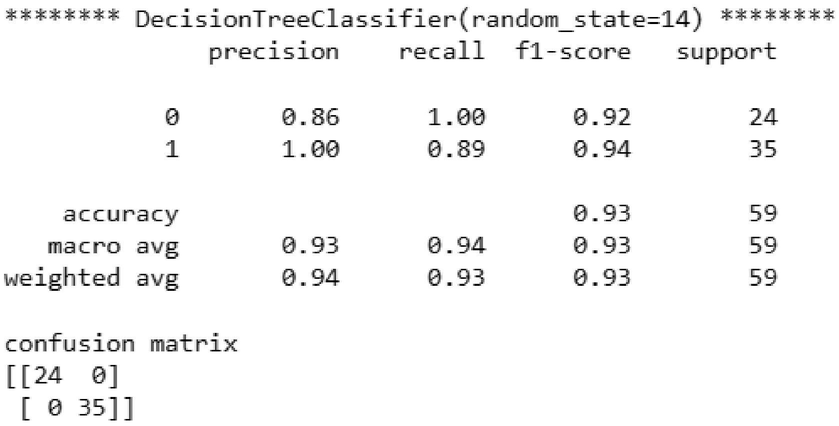
7.1.1 LOGISTIC REGRESSION



```
******** LogisticRegression(C=0.4, max_iter=1000, solver='liblinear') ********
              precision    recall  f1-score   support

           0       0.76      0.79      0.78        24
           1       0.85      0.83      0.84        35

    accuracy                           0.81        59
   macro avg       0.81      0.81      0.81        59
weighted avg       0.82      0.81      0.81        59

confusion matrix
[[24  0]
 [ 0 35]]
```

Figure 7.3: Logistic regression

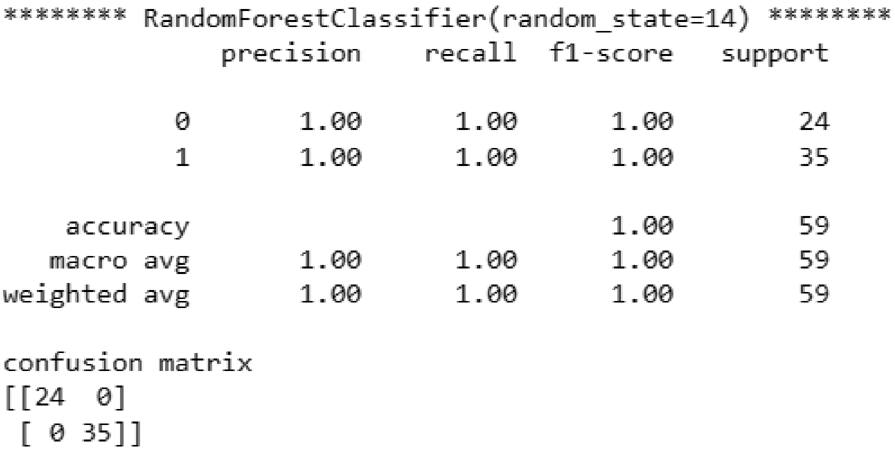
7.1.2 DECISION TREE



```
******** DecisionTreeClassifier(random_state=14) ********
              precision    recall  f1-score   support

           0       0.86      1.00      0.92        24
           1       1.00      0.89      0.94        35

    accuracy                           0.93        59
   macro avg       0.93      0.94      0.93        59
weighted avg       0.94      0.93      0.93        59

confusion matrix
[[24  0]
 [ 0 35]]
```

Figure 7.4: Decision tree

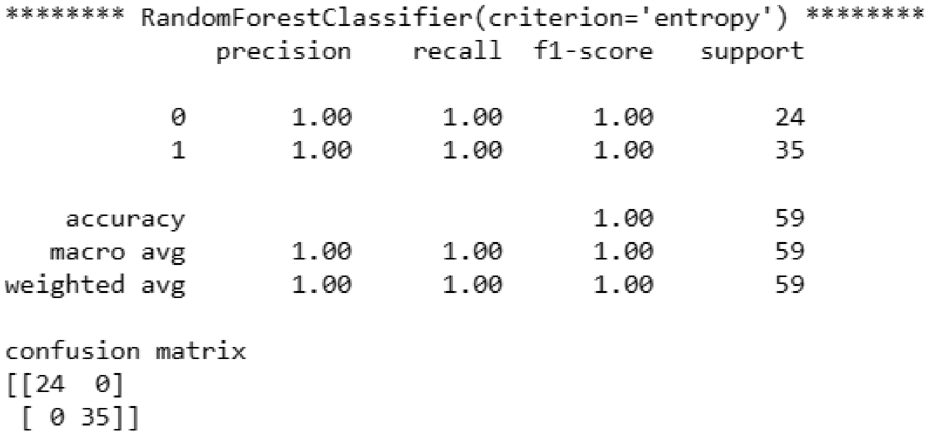
7.1.3 RANDOM FOREST – INFORMATION GAIN



```
******** RandomForestClassifier(random_state=14) ********
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        24
           1       1.00      1.00      1.00        35

    accuracy                           1.00        59
   macro avg       1.00      1.00      1.00        59
weighted avg       1.00      1.00      1.00        59

confusion matrix
[[24  0]
 [ 0 35]]
```

Figure 7.5: Random forest-information gain

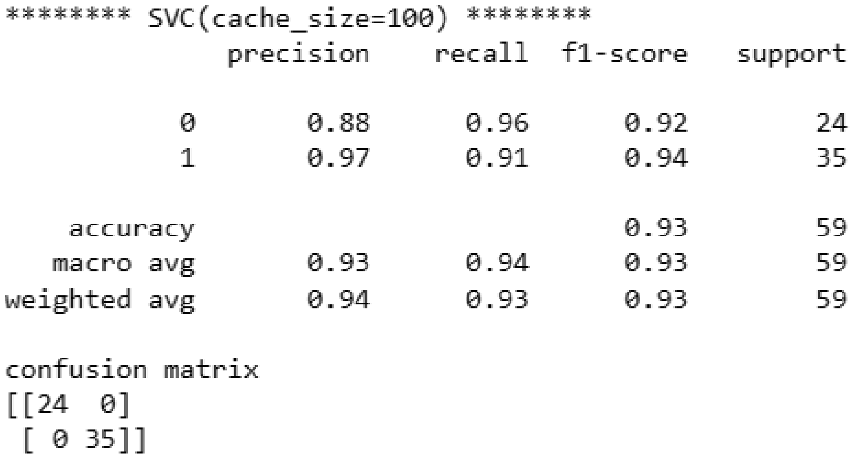
7.1.4 RANDOM FOREST – ENTROPY



```
******** RandomForestClassifier(criterion='entropy') ********
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        24
           1       1.00      1.00      1.00        35

    accuracy                           1.00        59
   macro avg       1.00      1.00      1.00        59
weighted avg       1.00      1.00      1.00        59

confusion matrix
[[24  0]
 [ 0 35]]
```

Figure 7.6: Random forest-entropy

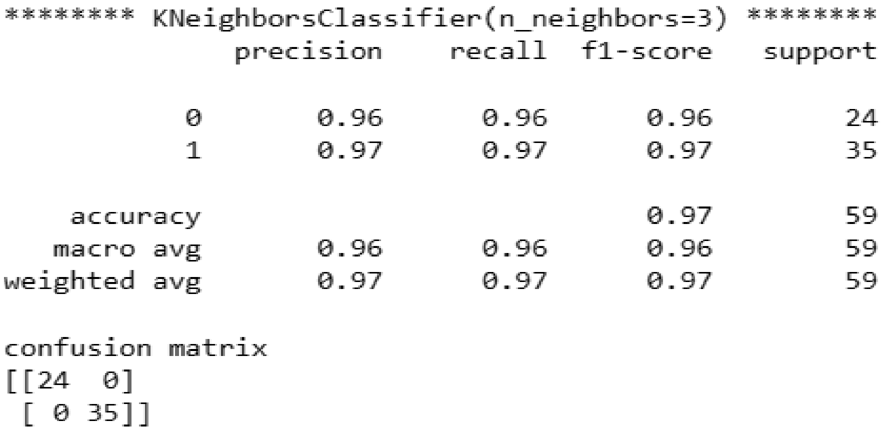
7.1.5 SUPPORT VECTOR MACHINE



```
******** SVC(cache_size=100) ********
              precision    recall  f1-score   support

           0       0.88      0.96      0.92        24
           1       0.97      0.91      0.94        35

    accuracy                           0.93        59
   macro avg       0.93      0.94      0.93        59
weighted avg       0.94      0.93      0.93        59

confusion matrix
[[24  0]
 [ 0 35]]
```

Figure 7.7: Support vector machine

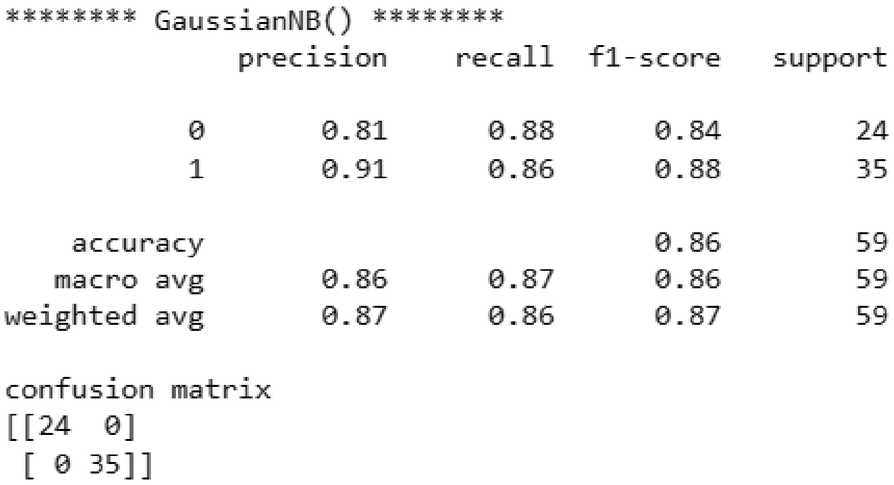
7.1.6 KNN



```
******** KNeighborsClassifier(n_neighbors=3) ********
              precision    recall  f1-score   support

           0       0.96      0.96      0.96        24
           1       0.97      0.97      0.97        35

    accuracy                           0.97        59
   macro avg       0.96      0.96      0.96        59
weighted avg       0.97      0.97      0.97        59

confusion matrix
[[24  0]
 [ 0 35]]
```

Figure 7.8: KNN

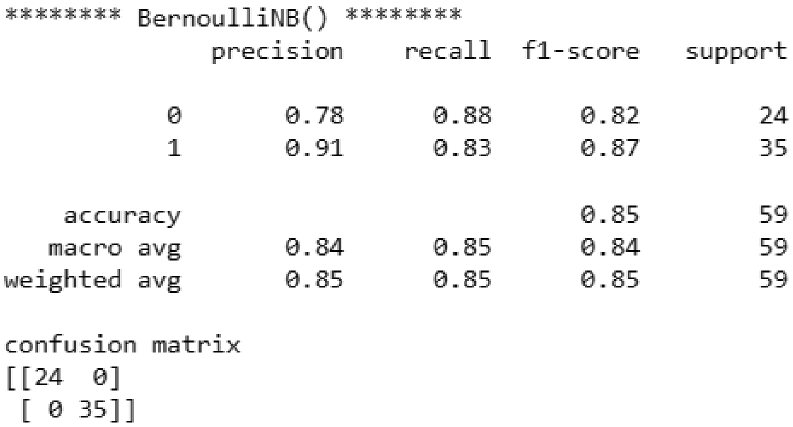
7.1.7 GAUSSIAN NAÏVE BAYES



```
******** GaussianNB() ********
              precision    recall  f1-score   support

           0       0.81      0.88      0.84        24
           1       0.91      0.86      0.88        35

    accuracy                           0.86        59
   macro avg       0.86      0.87      0.86        59
weighted avg       0.87      0.86      0.87        59

confusion matrix
[[24  0]
 [ 0 35]]
```

Figure 7.9: Gaussian naïve bayes

7.1.8 BERNOULLI NAÏVE BAYES



```
******** BernoulliNB() ********
              precision    recall  f1-score   support

           0       0.78      0.88      0.82        24
           1       0.91      0.83      0.87        35

    accuracy                           0.85        59
   macro avg       0.84      0.85      0.84        59
weighted avg       0.85      0.85      0.85        59

confusion matrix
[[24  0]
 [ 0 35]]
```

Figure 7.10: Bernoulli naïve bayes

7.2 TEST REPORTS

Table 7.1 – Test Reports

S.NO	Module	Testing
1	Loading the data	Successful!
2	Reading the data	Successful!
3	Logistic Regression Module	Successful!
4	Decision Tree Module	Successful!
5	Random Forest – Information Gain Module	Successful!
6	Random Forest – Entropy Module	Successful!
7	SVM Module	Successful!
8	KNN Module	Successful!

9	Gaussian Naïve Bayes Module	Successful!
10	Bernoulli Naïve Bayes Module	Successful!
11	Training the model	Successful!
12	Parkinson’s Disease Detection	Successful!
13	Obtaining Accuracy Results	Successful!

An individual's likelihood of having Parkinson's disease (PD) or not is often predicted or classified based on input data as the outcome of Parkinson's disease (PD) detection using machine learning (ML). The ML model is trained using a dataset that includes attributes from people with and without Parkinson's disease, including clinical data, genetic markers, imaging data, and sensor readings. The ML model may produce a binary outcome that divides people into the PD-positive and PD-negative groups. Based on the learnt patterns and decision limits of the ML model, this result suggests whether the person is likely to develop PD or not.The ML model may offer a probability score or confidence level in place of a binary classification to indicate the possibility of PD. The probability score, which ranges from 0 to 1, indicates the ML model's estimated probability that a person has PD. A higher probability score indicates a greater chance of PD. Diagnostic accuracy metrics are another way to quantify the outcome of PD detection using ML. Sensitivity, specificity, accuracy, precision, recall, or F1 score are some examples of these measurements. These measures measure how well the ML model distinguishes between those who have PD and those who don't. Charts or graphical representations can be used to see the outcome. For instance, a receiver operating characteristic (ROC) curve may be used to depict the true positive rate (sensitivity) and false positive rate of the ML model. AUC-ROC, or the area under the ROC curve, is a widely used measurement to evaluate the ML model's discriminative ability. It's crucial to remember that the specific outcome and how it should be interpreted rely on the ML algorithm, feature choice, model assessment metrics, and the unique objectives of the PD detection system. To guarantee the dependability and generalizability of the PD detection output, the performance and results of the ML model should be evaluated using proper evaluation methods and confirmed using separate datasets.

7.3 SECURITY ISSUES

The sensitivity and privacy of patient data, as well as possible weaknesses in the ML system, might cause security difficulties in the identification of Parkinson's disease (PD) using machine learning (ML). Here are a few security issues to think about -

- Patient Data Privacy: Personal and medical data, including as clinical information, genetic information, and imaging scans, are frequently used by PD detection systems. It is essential to safeguard the privacy of this delicate information. To avoid unauthorised access or data breaches, adequate safeguards must be in place, including secure data storage, data encryption, and stringent access restrictions.

- Data Authenticity and Integrity: Since ML models are trained on big datasets, any data manipulation might compromise the system's accuracy and dependability for detecting PD. It is crucial to maintain the data's authenticity and integrity throughout its existence. Data modification or injection attacks can be defended against using methods like digital signatures, secure data transmission protocols, and data validation techniques.
- Adversarial Attacks: Malicious individuals might deliberately alter input data to trick machine learning (ML) models used in PD detection, leading to inaccurate findings. Attacks from the enemy might result in incorrect diagnoses or incorrect forecasts. Adversarial attack risks can be reduced by using strategies such robust model training, input data sanitization, and anomaly detection.
- Model Poisoning: Attackers may add harmful or biassed data into the model training process in order to alter the model's behaviour. ML models are susceptible to model poisoning assaults. This may result in inaccurate or skewed forecasts. Model poisoning assaults may be reduced by regularly checking and monitoring the training data, putting in place outlier detection systems, and using reliable model training methods.
- Model Explanations and Transparency: For ethical and security concerns, it's crucial to guarantee the transparency and interpretability of the ML models employed in PD detection. Building confidence among medical professionals and patients can be facilitated by providing reasons or explanations for the model's predictions. The transparency of ML systems may be improved by using interpretable ML methods, such as decision trees or rule-based models, and by offering feature significance analyses.
- System Vulnerabilities: Software or infrastructural flaws may exist in ML systems used in PD detection. To discover and address any vulnerabilities, it is crucial to constantly update and patch the software components, protect the hosting infrastructure, and carry out security audits. System vulnerabilities can be addressed by using safe software development best practises and adhering to security regulations.
- Regulatory Compliance: PD detection systems must abide by pertinent data protection and privacy legislation, such as the Health Insurance Portability and Accountability Act (HIPAA) or the General Data Protection Regulation (GDPR). Ensuring compliance with these rules helps safeguard patient information, build confidence, and prevents legal repercussions.

A multifaceted strategy, including safe data management, strong model training, secure system architecture, and adherence to privacy legislation, is needed to address these security challenges. The risks of data breaches, adversarial attacks, and system vulnerabilities may be reduced by adopting security measures throughout the PD detection system's design, development, and deployment. This will ensure the privacy and security of patient information.

CHAPTER 8: CONCLUSION

8.1 CONCLUSION

In conclusion, early diagnosis and treatment of Parkinson's disease (PD) might be significantly improved with the use of machine learning (ML) for PD identification. In order to help identify PD, ML algorithms have the capacity to analyse complicated patterns and characteristics in clinical data, genetic markers, imaging scans, and sensor readings. The use of ML in PD detection has a number of advantages. It permits the creation of precise and trustworthy prediction models that can help medical practitioners make defensible choices. In order to improve the precision and effectiveness of PD diagnosis, ML models can reveal hidden links and patterns in data that may not be visible to human observers. Systems for detecting Parkinson's disease (PD) may be able to identify the condition early on by utilising ML approaches, allowing for prompt treatment and better patient outcomes. ML models can support the monitoring of illness development and therapy response, enabling individualised and focused therapeutic approaches. However, there are many factors to take into account and difficulties in PD identification using ML. Robust ML models must be trained using sufficient and varied datasets that cover various illness stages and demographic features. To maintain openness and win the trust of medical professionals and patients, interpretability and explain ability of ML models are still crucial. Furthermore, it is crucial to ensure the privacy and security of patient data. Protecting patient information and preserving the integrity of the PD detection system requires strict adherence to privacy laws, the adoption of secure data processing procedures, and defence against hostile assaults. Overall, Parkinson's disease detection using machine learning has enormous potential to increase diagnostic precision, allow for early intervention, and improve patient care. The creation of efficient and trustworthy PD detection systems will be further aided by ongoing research, collaboration between data scientists and medical practitioners, and improvements in ML algorithms.

8.2 EXPLANATION

- Random Forest Information Gain and Entropy has the best 100% accuracy
- KNN following it has the accuracy of 96%
- Decision Tree and SVM has accuracy score of 93%
- Gaussian Naive Bayes stands with 86% meanwhile Bernoulli Naive Bayes stands with 84%
- The minimum Accuracy score amongst all is of Logistic Regression with 81%

8.3 RECOMMEDNDATION

The order of recommendation is –

Random Forest Information Gain > Random Forest Entropy > KNN > Decision Tree > SVM > Gaussian Naive Bayes > Bernoulli Naive Bayes > Logistic Regression

FUTURE SCOPE OF THE PROJECT

The future potential for detecting Parkinson's disease (PD) using machine learning (ML) is bright and includes a number of new developments. Here are some possible routes for PD detection using ML in the future -

- Learning Transfer and Generalisation: ML models may be used for cross-population and cross-institutional deployment after being trained on huge datasets from varied groups. Transfer learning methods can aid in the generalisation of ML models to fresh patient populations or healthcare environments, facilitating easier accessibility to effective PD detection technologies.
- Explainable AI and Clinical Decision Support: Improving the interpretability and explain ability of machine learning (ML) models employed in Parkinson's disease (PD) diagnosis is essential for winning over the trust and acceptance of medical professionals. Future research might concentrate on creating explainable AI methods that offer concise and intelligible justifications for the predictions made by the ML model. This can aid clinical judgement and raise support for ML-based PD detection methods.
- Integration with Electronic Health Records (EHR): By integrating ML-based PD detection systems with EHRs, smooth data sharing and thorough patient profiling may be made possible. PD diagnosis accuracy may be increased and clinical decision support systems can be supported by ML algorithms by utilising the extensive clinical data from EHRs.

The continued development of algorithms, methods for gathering data, and interaction with healthcare systems are key to the success of PD diagnosis via machine learning. Parkinson's disease (PD) detection using machine learning (ML) has the potential to revolutionize early diagnosis, individualized therapy, and overall management of the condition.

INDIVIDUAL REPORT

Our project is basically being developed for the detection of the best model for the prediction of Parkinson’s disease in humans. The projects work on eight different algorithms and has been developed on Jupyter notebook. The project tells after evaluating and analyzing the dataset about the best model to be used with new data or user entered data for the prediction. It has been developed for the betterment and timely treatment of the patients, which might also decrease the fatality rate.

Team members contribution:

- Shubham Tiwari – Training the model, presentations and research paper
- Saakshi Srivastava - Training the model, presentations and research paper
- Kartikey Raghuvanshi – Front-end, thesis and research paper
- Sanand Mishra - Front-end, thesis and research paper

REFERENCES

[1]. Heisters. D, “Parkinson’s: symptoms, treatments and research”. British Journal of Nursing, 20(9), 548–554. doi:10.12968/bjon.2011.20.9.548, 2011.

[2]. Ozcift, “SVM feature selection-based rotation forest ensemble classifiers to improve computer-aided diagnosis of Parkinson disease” Journal of medical systems, vol-36, no. 4, pp. 2141-2147, 2012.

[3]. Dr. R. Geetha Ramani, G. Sivagami, Shomona Graciajacob “Feature Relevance Analysis and Classification of Parkinson’s Disease Tele Monitoring data Through Data Mining” International Journal of Advanced Research in Computer Science and Software Engineering, vol-2, Issue 3, March 2012.

[4]. Farhad Soleimanian Gharehehopogh, Peymen Mohammadi, “A Case Study of Parkinson’s Disease Diagnosis Using Artificial Neural Networks” International Journal of Computer Applications, Vol-73, No.19, July 2013.

[5]. Dragana Miljkovic et al, “Machine Learning and Data Mining Methods for Managing Parkinson’s Disease” LNAI 9605, pp. 209-220, 2016.

[6]. Arvind Kumar Tiwari, “Machine Learning based Approaches for Prediction of Parkinson’s Disease” Machine Learning and Applications: An International Journal (MLAU) vol. 3, June 2016.

[7]. Dr. Anupam Bhatia and Raunak Sulekh, “Predictive Model for Parkinson’s Disease through Naïve Bayes Classification” International Journal of Computer Science & Communication vol-9, Dec. 2017, pp. 194- 202, Sept 2017 - March 2018

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

[9]. Carlo Ricciardi, et al, “Using gait analysis’ parameters to classify Parkinsonism: A data mining approach” Computer Methods and Programs in Biomedicine vol. 180, Oct. 2019.

[10]. Anila M Department of CS1, Dr G Pradeepini Department of CSE, “DIAGNOSIS OF PARKINSON’S DISEASE USING ARTIFICIAL NEURAL NETWORK”, JCR, 7(19): 7260-7269, 2020.

GLOSSARY

Algorithm - An algorithm is a set of instructions that are executed at run time. Code - Code is a term used to describe the collection of instructions or commands that are used by various programming languages to write, change, or administer computer programmes or applications. Computers employ code to specify parameters, choose the appropriate action, and more.

File Extension- The part of the filename that comes after the final period is frequently used to indicate the kind of file when naming files. Examples include txt, pdf, and jpg.

IDE - A software programme that aids in software development is known as a "Integrated Development Environment," or "IDE." Microsoft Visual Studio, Eclipse, and NetBeans are a few examples of open-source software.

Keyword - Keywords are words with a specific meaning that are only allowed to be used for that meaning in a language. For example, "for" in Processing or C++.

Machine Learning– Machine learning, a branch of artificial intelligence, uses statistical modelling to aid computers in learning from experience.

Programming - Developing a programme, typically in a high-level language, is referred to as programming. the practise of teaching computers how to automatically store, retrieve, and process data.

RAM - RAM, or random access memory, is a technical term. Usually refers to memory that is separate from the CPU and a disc. It is typically volatile, meaning that when the computer is shut off, whatever in it is lost. Among other things, it's used to momentarily hold active instructions and data during execution.

Syntax - The rules that determine whether a statement is a language and not just meaning are known as a language's syntax.

RESEARCH PAPER

A Symphony of Signals: Machine Learning Enabled Parkinson's Disease Detection via Audio Analysis through various Algorithms

Shubham Tiwari[1], Kartikey Raghuvanshi[2], Sanand Mishra[3], Saakshi Srivastava[4], Ayasha Malik[5]

[1,2,3,4] Department of Computer Science and Engineering, Delhi Technical Campus, Greater Noida, UP, India

[5]Assistant Professor, Department of Computer Science and Engineering, Delhi Technical Campus, Greater Noida, UP, India

Abstract - Parkinson's disease is a neurological condition that affects people of all ages worldwide and causes abnormalities in brain activity and motor function. For the early diagnosis of Parkinson's disease symptoms, medical research has recently used computational intelligence tools, notably machine learning and deep learning approaches. These methods make use of numerous medical measurements made using various medical equipment, such as voice volume, handwriting fluctuations, bodily motions, brain signal variations, and protein aggregations. The moderate nature of the earliest indicators, however, makes it difficult to recognise Parkinson's disease in its early stages. The algorithms of machine learning used for Parkinson's disease diagnosis and prediction based on audio data is the main topic of this research study. Particularly, the examination of voice-related symptoms offers a potential route for practical and non-invasive screening methods. Parkinson's disease is diagnosed based on both motor and non-motor symptoms, such as slowness of movement, stiffness, balance issues, tremors, anxiety, breathing issues, depression, loss of smell, and changes in speech. Our work intends to identify the most accurate diagnostic method/algorithm for early Parkinson's disease identification by taking into account speech characteristics and patient data.

Keywords: Parkinson’s disease, Model, Detection, SVM, Machine Learning, Algorithms

1. INTRODUCTION

Parkinson's disease is a common neurological condition that affects people of all ages worldwide. Disruptions in brain activity and bodily motion are its defining characteristics, which cause a variety of motor and non-motor symptoms. For successful care and intervention, Parkinson's disease must be recognised and detected early. By using computational intelligence methods, notably machine learning and deep learning methodologies, to anticipate and diagnose Parkinson's disease symptoms, medical research has made considerable strides recently. Parkinson's disease is categorised based on the specific anomalies seen in those who have it. The disorder mostly affects how the brain functions and how the body moves, and it shows up as symptoms including slowness of movement, stiffness, balance issues, tremors, anxiety, breathing problems, depression, loss of smell, and altered speech. Medical practitioners utilise a variety of medical observations, such as voice volume, handwriting alterations, body movements, brain signal abnormalities, and protein aggregations, assessed using specialised medical equipment, to diagnose Parkinson's disease. The fact that Parkinson's disease is a progressive condition with worse symptoms over time presents a significant diagnostic issue. To detect the illness in its early stages, more sensitive diagnostic methods must be developed. Analysing voice-related symptoms is one of the prospective screening procedures that has promise since it can be easily recorded using non-invasive means, including mobile devices. The intricacy of the earliest symptoms of Parkinson's disease makes early detection extremely difficult. The main goal of this study is to develop various machine learning models for Parkinson's disease diagnosis and prediction using auditory inputs and find the best among them. The project intends to establish an accurate and sensitive diagnostic approach for early Parkinson's disease identification by analysing speech aspects and combining patient data. The planned study emphasises how crucial early diagnosis is for prompt management and better patient care. This work aims to expand the area of Parkinson's disease research and contribute to the creation of efficient screening approaches for early detection through the integration of machine learning techniques with audio signal analysis. The idea of neurodegenerative illnesses and its underlying processes will also be covered in this study. The brain's functional components, or neurons, are essential for sustaining brain function. The increasing shredding and death of neurons in numerous parts of the nervous system, however, is a feature of neurodegenerative illnesses. As neurons are harmed, their capacity for communication declines, which causes a drop in their metabolism. The degeneration of the damaged neurons is further accelerated by the accumulation of cellular debris and the subsequent development of vacuoles. In summary, the goal of this study is to use machine learning and acoustic signal analysis to advance the area of Parkinson's disease identification. Enhancing early detection skills and enabling prompt therapies are the ultimate goals in order to improve the lives of people with Parkinson's disease. In order to better understand the pathophysiology of Parkinson's disease and pave the way for future developments in diagnostic procedures and therapeutic approaches, this project aims to get a deeper knowledge of the underlying processes of neurodegenerative illnesses. The overall format/flow of the paper is as follows: The paper's introduction, which has previously been examined, is the first part. The literature review, or all the papers reviewed to produce this research study, is found in Section 2, which we will now go on to. Reviewing the previous research on the subject here aids in creating a better research report. The technique, which comprises of numerous modules and various functional and non-functional needs, is covered in Section 3 of the article. The necessary hardware and software are listed in the next section. Furthermore, outcomes that commonly displayed the application snapshots are discussed in Section 5 of this article. The study comes to a close with Section 6, which contains the recommendations.

2. LITREATURE REVIEW

A thorough summary of earlier studies on the subject is given in the literature review. We shall synthesise and summarise the results of relevant studies in the following table. The research papers that were read in order to effectively compose this one are shown in the table below:

Table 1 – Literature Survey

S.No.	Year	Name	Contribution
1	2011	Heisters D. [1]	Parkinson's disease is an irreversible neurological ailment that causes slowness of movement, tremor, and stiffness of the muscles. The main form of therapy is medication, and continuing research is being done to discover a cure and provide new therapies.
2	2012	A. Ozcift [2]	With up to 97% accuracy in the top-performing classifier, a novel classification model based on support vector machine and rotation forest ensemble classifiers has been created to enhance Parkinson's disease detection.
3	2012	Dr. R. Geetha Ramani et al. [3]	This study employs data mining methods and biological voice measurements to categorise the severity of Parkinson's disease with 100% accuracy using the Random Tree classification algorithm and ReliefF algorithm.
4	2013	Farhad Soleimanian Gharehehopogh et al. [4]	This study classifies Parkinson's disease with great accuracy using two types of artificial neural networks (MBF and MLP), which can help neurologists make better choices.
5	2016	Dragana Miljkovic et al. [5]	The use of machine learning techniques to identify and categorise tremors, gait patterns, and voice dysfunction

			in Parkinson's disease patients is covered in this research.
6	2016	Arvind kumar tiwari [6]	In this study, random forest with 20 chosen characteristics is used to predict Parkinson's disease with an overall accuracy of 90.3%.
7	2018	Dr. Anupam bhatia et al. [7]	In order to identify the most precise classification method, this research intends to identify Parkinson's Disease by data mining and statistical study of typical symptoms including gait, tremors, and micro-graphia.
8	2018	M. Abdar et al. [8]	This study uses Parkinson's disease data from UCI to evaluate the diagnostic performance of SVM and Bayesian networks, and it revealed that SVM with polynomial kernel function and C parameter performed the best, with an average accuracy of 99.18%. Additionally, the SVM algorithm's 10 most crucial components were found.
9	2019	Carlo Ricciardi et al. [9]	Data mining can provide light on the small variations between Parkinson's disease and Progressive Supranuclear Palsy, which can be distinguished via gait analysis.
10	2020	Anila M et al. [10]	The study proposes a unique method for accurately diagnosing Parkinson's disease using artificial neural network models.

3. METHODOLOGY

This research paper discusses about the Parkinson’s disease and how it’s detection is possible using a machine learning model. In this paper, we are going to discuss about various Machine Learning algorithms through which this disease can be detected and find the best amongst them. We are using different models for prediction and later on a comparative study will be done. For model training along with ML some python libraries such as NumPy, Pandas, scikit-learn will be used.

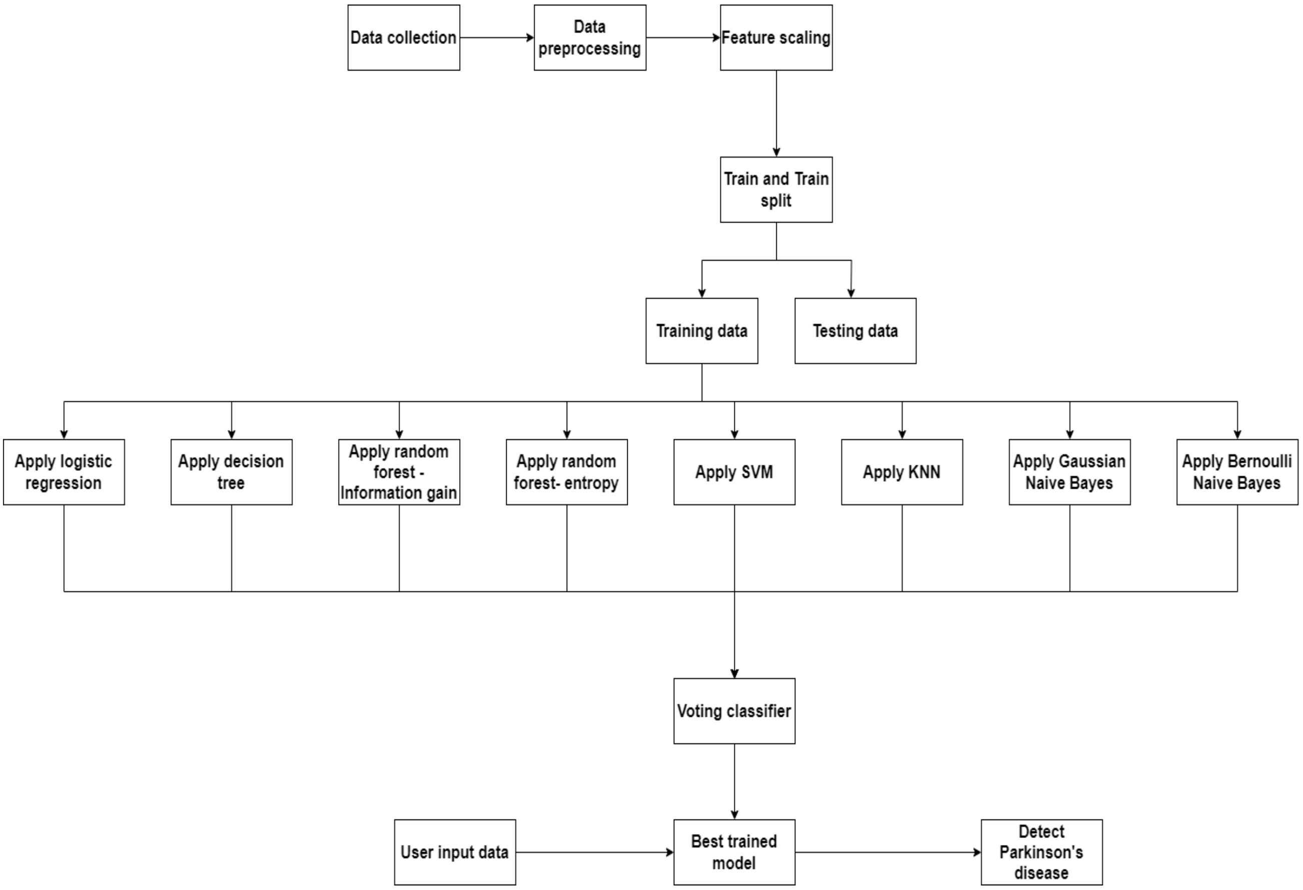


Fig 1: Conceptual Model

3.1 Functional requirements

In Parkinson's disease applications, functional criteria are crucial to ensure that the built system satisfies the demands of the intended users and stakeholders. Several crucial functional needs for Parkinson's disease applications are listed below:

3.1.1. Data collection

In order to follow disease development, identify risk factors, and assess the efficacy of therapies, data collecting is a crucial part of research into Parkinson's disease. Data collection is done from Kaggle with 24 feature/characteristics of 195 people of different age groups. The features are:

- name - ASCII subject name and recording number
- MDVP:Fo(Hz) - Average vocal fundamental frequency
- MDVP:Fhi(Hz) - Maximum vocal fundamental frequency
- MDVP:Flo(Hz) - Minimum vocal fundamental frequency
- MDVP:Jitter(%), MDVP:Jitter(Abs), MDVP:RAP, MDVP:PPQ, Jitter:DDP - Several measures of variation in fundamental frequency
- MDVP:Shimmer,MDVP:Shimmer(dB),Shimmer:APQ3,Shimmer:APQ5,MDVP:APQ,Shimmer:DDA - Several measures of variation in amplitude

- NHR, HNR - Two measures of the ratio of noise to tonal components in the voice
- status - The health status of the subject (one) - Parkinson's, (zero) – healthy
- RPDE, D2 - Two nonlinear dynamical complexity measures
- DFA - Signal fractal scaling exponent
- spread1,spread2,PPE - Three nonlinear measures of fundamental frequency variation

The correlation between all the features is represented in the below mentioned heatmap of the dataset.

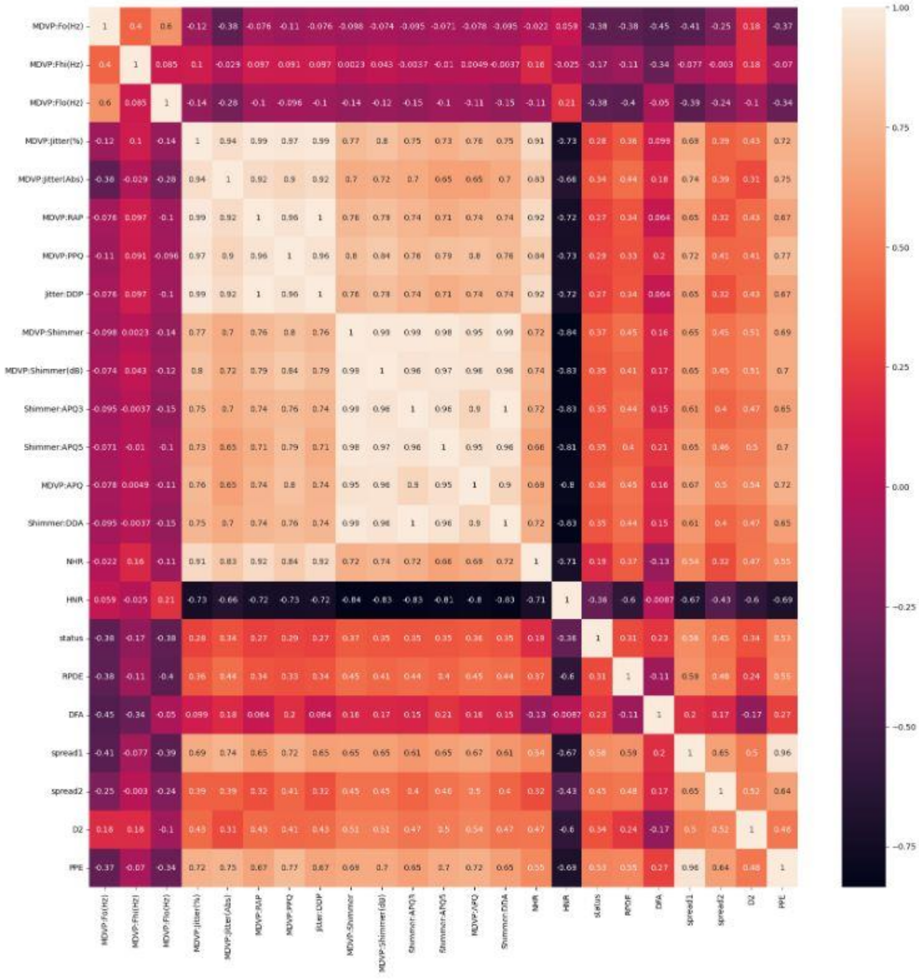
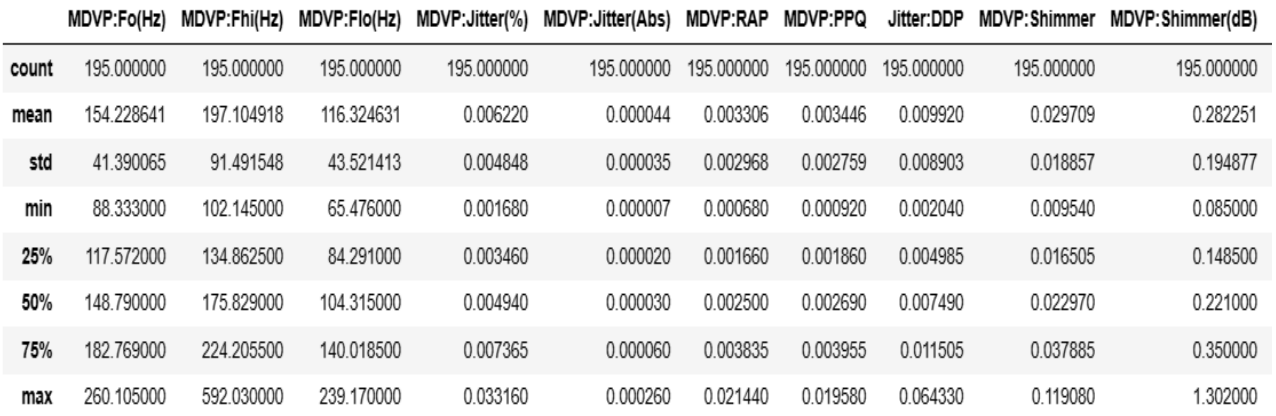


Fig 2: Heatmap

3.1.2. Data processing and analysis

For the purpose of creating machine learning models for Parkinson's disease research, data processing and analysis are crucial processes. The data is statistically analysed for the better understanding of the dataset.



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

Fig 3: Statistical Description of Data

The kind of data and the research issue influence the model and training procedure selection. Research on Parkinson's disease may make use of a variety of machine learning models, such as decision trees, random forests, support vector machines etc. The intricacy of the data and the required level of accuracy will determine which model is used. Once a model has been chosen, it is trained using a dataset to show it how to spot patterns in the data. A training set and a validation/test set are created from the dataset. The training set is used to train the model, while the validation/test set is used to assess the model's performance. The training and test data are split in 80:20 ratio.

Table 2 – Data after being split into test and train sets

No of people’s data for training of model	156
No of people’s data for testing of model	39

Until the model's performance is sufficient, this process is repeated. For training and validation, it is crucial to use high-quality data to guarantee the model's correctness and dependability. For Data Pre-processing it is important to know the pair-plot of the characteristics.

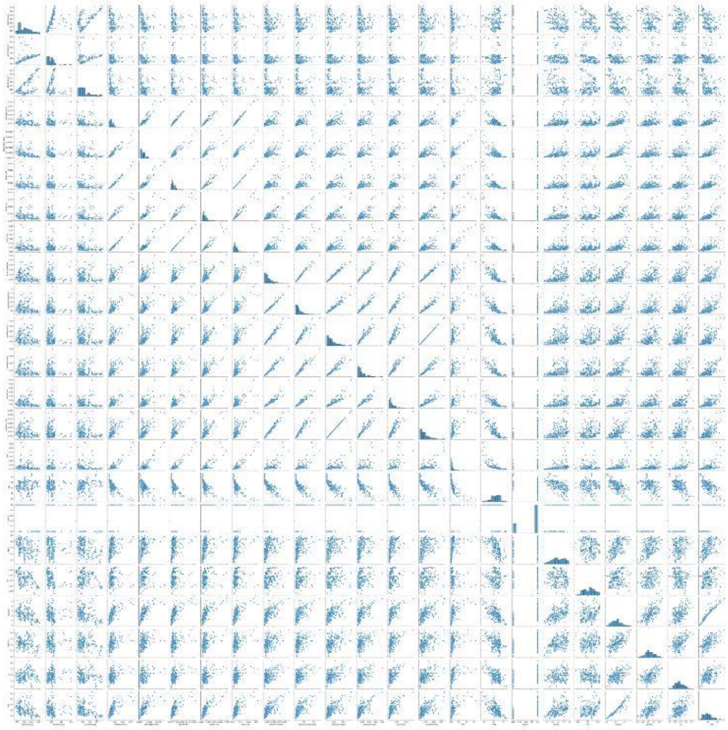
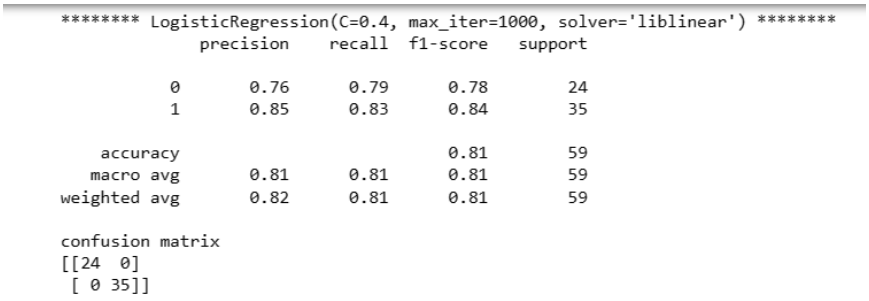


Fig 4: Pair-Plot Graphs of the attributes

3.1.3. Model Training through various algorithms

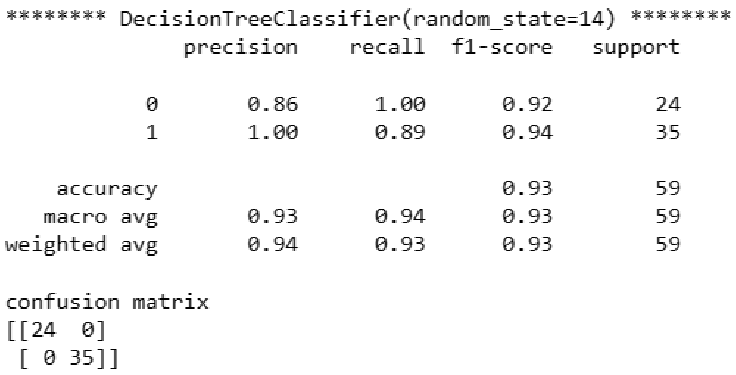
- **Logistic Regression**



```
******** LogisticRegression(C=0.4, max_iter=1000, solver='liblinear') ********
              precision    recall  f1-score   support

           0       0.76      0.79      0.78        24
           1       0.85      0.83      0.84        35

    accuracy                           0.81        59
   macro avg       0.81      0.81      0.81        59
weighted avg       0.82      0.81      0.81        59

confusion matrix
[[24  0]
 [ 0 35]]
```

Fig 5: Model Results of Logistic Regression

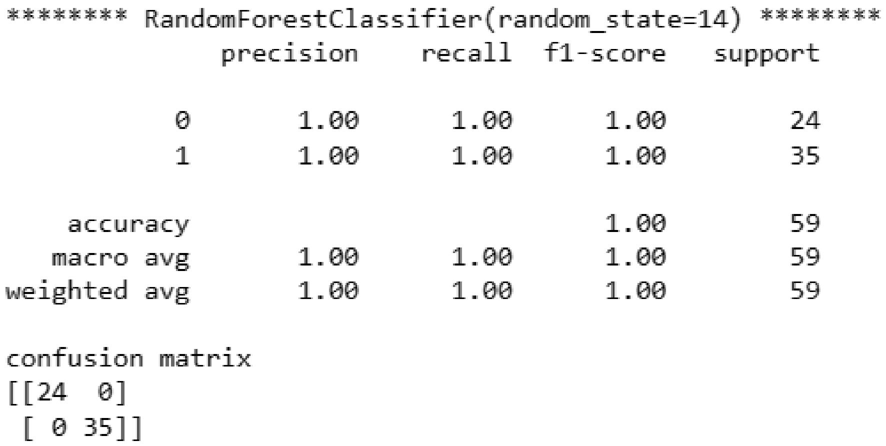
- **Decision Tree**



```
******** DecisionTreeClassifier(random_state=14) ********
              precision    recall  f1-score   support

           0       0.86      1.00      0.92        24
           1       1.00      0.89      0.94        35

    accuracy                           0.93        59
   macro avg       0.93      0.94      0.93        59
weighted avg       0.94      0.93      0.93        59

confusion matrix
[[24  0]
 [ 0 35]]
```

Fig 6: Model Results of Decision Tree

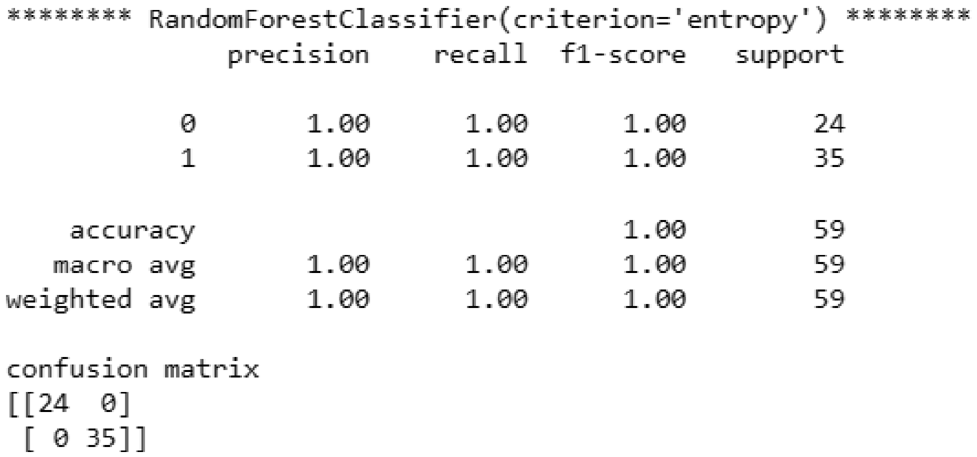
- **Random Forest – Information Gain**



```
******** RandomForestClassifier(random_state=14) ********
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        24
           1       1.00      1.00      1.00        35

    accuracy                           1.00        59
   macro avg       1.00      1.00      1.00        59
weighted avg       1.00      1.00      1.00        59

confusion matrix
[[24  0]
 [ 0 35]]
```

Fig 7: Model Results of Random Forest – Information Gain

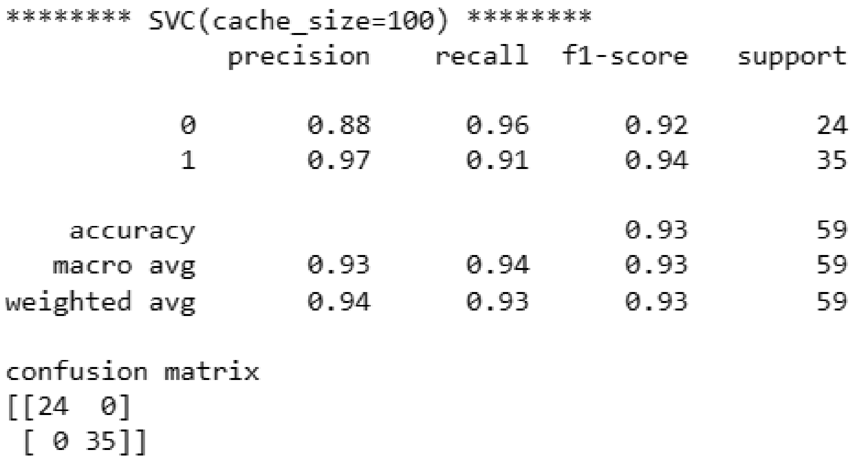
- **Random Forest – Entropy**



```
******** RandomForestClassifier(criterion='entropy') ********
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        24
           1       1.00      1.00      1.00        35

    accuracy                           1.00        59
   macro avg       1.00      1.00      1.00        59
weighted avg       1.00      1.00      1.00        59

confusion matrix
[[24  0]
 [ 0 35]]
```

Fig 8: Model Results of Random Forest – Entropy

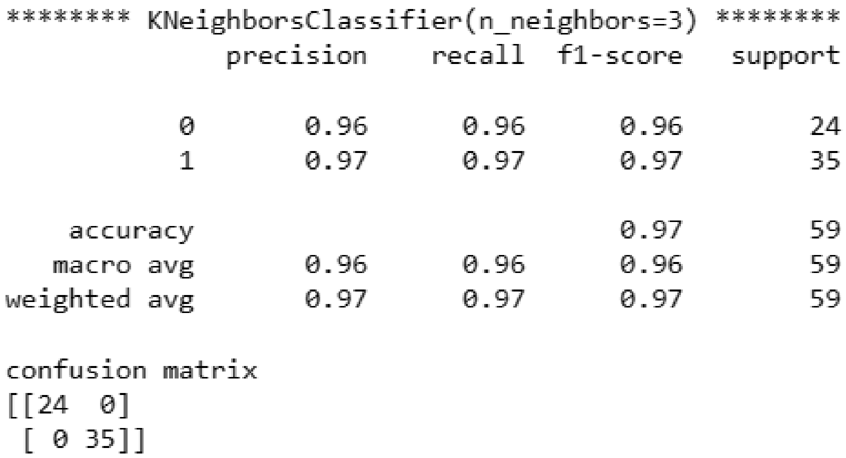
- **Support Vector Machine**



```
******** SVC(cache_size=100) ********
              precision    recall  f1-score   support

           0       0.88      0.96      0.92        24
           1       0.97      0.91      0.94        35

    accuracy                           0.93        59
   macro avg       0.93      0.94      0.93        59
weighted avg       0.94      0.93      0.93        59

confusion matrix
[[24  0]
 [ 0 35]]
```

Fig 9: Model Results of SVM

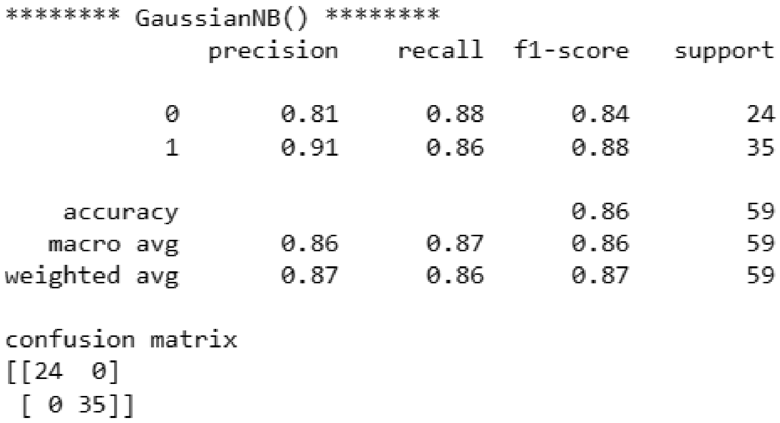
- **KNN**



```
******** KNeighborsClassifier(n_neighbors=3) ********
              precision    recall  f1-score   support

           0       0.96      0.96      0.96        24
           1       0.97      0.97      0.97        35

    accuracy                           0.97        59
   macro avg       0.96      0.96      0.96        59
weighted avg       0.97      0.97      0.97        59

confusion matrix
[[24  0]
 [ 0 35]]
```

Fig 10: Model Results of KNN

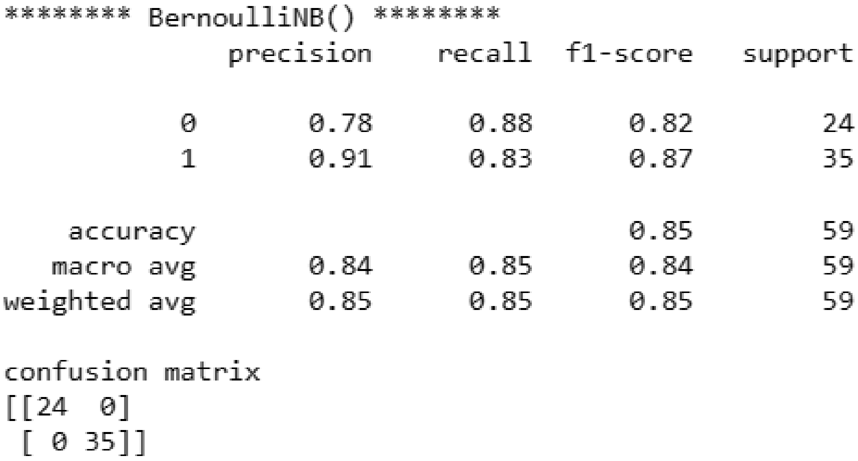
- **Gaussian Naïve Bayes**



```
******** GaussianNB() ********
              precision    recall  f1-score   support

           0       0.81      0.88      0.84        24
           1       0.91      0.86      0.88        35

    accuracy                           0.86        59
   macro avg       0.86      0.87      0.86        59
weighted avg       0.87      0.86      0.87        59

confusion matrix
[[24  0]
 [ 0 35]]
```

Fig 11: Model Results of Gaussian Naïve Bayes

- **Bernoulli Naïve Bayes**



```
******** BernoulliNB() ********
              precision    recall  f1-score   support

           0       0.78      0.88      0.82        24
           1       0.91      0.83      0.87        35

    accuracy                           0.85        59
   macro avg       0.84      0.85      0.84        59
weighted avg       0.85      0.85      0.85        59

confusion matrix
[[24  0]
 [ 0 35]]
```

Fig 12: Model Results of Bernoulli Naïve Bayes

3.2 Non-functional requirements

In Parkinson's disease, non-functional criteria are just as crucial as functional requirements since they guarantee that the system or application operates effectively, securely, and efficiently while delivering a great user experience.

- **Usability**: Even for users who may experience physical or cognitive difficulties as a result of Parkinson's disease, the system or application should be simple to use and navigate.
- **Performance**: Performance is a key consideration in machine learning projects for Parkinson's disease since it affects how well-built models are at correctly forecasting the onset or course of the condition. Depending on the particular issue and the desired result, performance can be assessed using a variety of measures, including accuracy, precision, recall, and F1 score.
- **Security**: The programme or system should make sure that patient information is safe and shielded from unauthorised access.
- **Reliability**: In machine learning initiatives for Parkinson's disease, reliability is essential since it guarantees that the built models can produce reliable findings.
- **Accessibility**: The programme or system must adhere to accessibility guidelines and be usable by people with impairments.
- **Compatibility**: To guarantee widespread use, the system or application should work with a variety of hardware and operating systems.

4. SOFTWARE AND HARDWARE REQUIREMENTS

The minimal software prerequisites for launching our product are described below. Prior to scaling out, it is advised to observe the results of pilot projects as requirements may change depending on utilisation.

4.1 Software requirements

- **Python:** Python is an interpreted, object-oriented, high-level, dynamically semantic programming language. It is particularly appealing for Rapid Application Development as well as for usage as a scripting or glue language to tie existing components together due to its high-level built-in data structures, dynamic typing, and dynamic binding.

Some other python libraries and packages which are being used:

- **PIP:** The software packages given in Python are installed and maintained using the PIP package management system.
- **NumPy:** The all-purpose array processing software is called NumPy. For using these arrays, it offers a high-performance multidimensional array object and tool. It is a essential Python module for scientific computing. It has several characteristics, but the following are the most significant ones. A strong n-dimensional array object. Complex functions

C/C++ and Fortran coding integration tools. Useful liner algebra Capability for the Fourier transform and random numbers.

- **Matplotlib:** An interactive visualisation tool called the Python Library for Matplotlib is used. Matplotlib makes both difficult and basic tasks feasible. Plots should be of publishing quality. Create animated, zoomable, and updating characters.
- **Jupyter Notebook:** The condom package with virtual environment manager is included in the anaconda distribution's 1500 packages, which were chosen from PYPI. In addition to the command line interface (CLI), Hit also has a graphical user interface (GUI), called Anaconda navigator. Typically ending with the ".ipynb" suffix, a Jupiter notebook document is a GSON file that follows a versioned format and has an ordered set of input output cells that can include code, text, arithmetic, graphs, and rich media.
- **Seaborn:** A matplotlib-based Python data visualisation package is called Seaborn. It offers a sophisticated drawing tool for creating eye-catching and educational statistics visuals. You can read the introductory notes or the paper for a quick overview of the concepts underlying the library.
- **Pandas:** Python's Pandas package is used to manipulate data collections. It offers tools for data exploration, cleaning, analysis, and manipulation. Wes McKinney came up with the moniker "Pandas" in 2008, and it refers to both "Panel Data" and "Python Data Analysis."
- **Sklearn:** In the Python environment, Scikit-learn, an open-source data analysis toolkit, is considered to be the pinnacle of machine learning (ML). Important ideas and traits include algorithms for making decisions, such as: Identification and classification, patterns in the data.
- **XGBoost:** The family of gradient boosting algorithms includes XGBoost (eXtreme Gradient Boosting), a potent and popular machine learning technique. By building an ensemble of weak prediction models, such decision trees, then combining their predictions to produce precise and reliable forecasts, it is intended to tackle classification and regression issues.

4.2 Hardware requirements

The minimal hardware specifications for implementing our product are described in this tutorial. Prior to scaling out, it is advised to observe the results of pilot projects as requirements may change depending on utilisation.

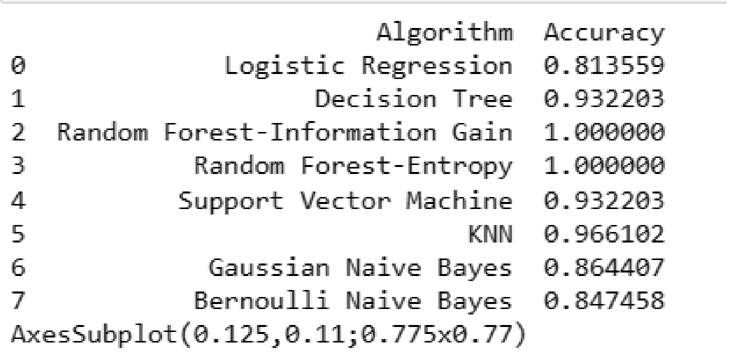
- OS: Window7
- Installed RAM: 4 GB
- Processor: Intel i3 or above

- System type: 64-bit operating system

5. RESULTS AND DISCUSSIONS

Parkinson's disease is extremely serious and currently has no known treatment. The speech is also impacted since it affects how the body's component’s function. Here, the model aims to provide the best method of recognising Parkinson's disease so that one may take swift action to lessen or perhaps delay it from impacting the entire body. It intends to create this way of understanding a Parkinson's case as early as possible via both the patient and scientific professionals. For researchers, engineers, and medical professionals, Parkinson's disease prediction continues to be one of the most challenging jobs. Making use of various machine learning models, lets the user to know which algorithm to use in future for the detection.

5.1 Snapshots



```
                          Algorithm  Accuracy
0               Logistic Regression  0.813559
1                     Decision Tree  0.932203
2  Random Forest-Information Gain  1.000000
3            Random Forest-Entropy  1.000000
4            Support Vector Machine  0.932203
5                               KNN  0.966102
6              Gaussian Naive Bayes  0.864407
7             Bernoulli Naive Bayes  0.847458
AxesSubplot(0.125,0.11;0.775x0.77)
```

Fig 13: ML Models Accuracy

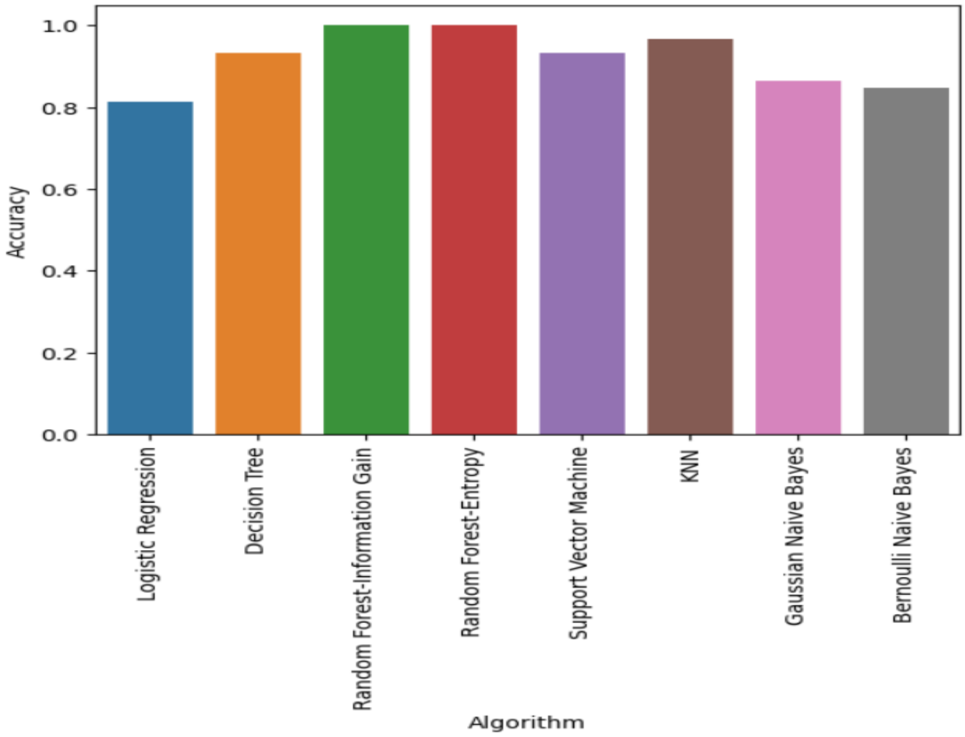


Fig 14: ML Models Accuracy’s Graphical Representation

6. CONCLUSION

- Random Forest Information Gain and Entropy has the best 100% accuracy
- KNN following it has the accuracy of 96%
- Decision Tree and SVM has accuracy score of 93%
- Gaussian Naive Bayes stands with 86% meanwhile Bernoulli Naive Bayes stands with 84%
- The minimum Accuracy score amongst all is of Logistic Regression with 81%

The order of recommendation is –

Random Forest Information Gain > Random Forest Entropy > KNN > Decision Tree > SVM > Gaussian Naive Bayes > Bernoulli Naive Bayes > Logistic Regression

7. FUTURE SCOPE

Machine learning (ML)-based Parkinson's disease diagnosis offers a number of recommendations for the future that will improve the process's precision, effectiveness, and accessibility. Here are some potential directions for future work and ideas for enhancements:

- **Dataset Expansion**: The performance of ML models may be improved by expanding the amount and variety of the dataset used for training.
- **Feature selection and engineering:** The accuracy of ML models may be increased by carefully choosing and honing the characteristics that are utilised to diagnose Parkinson's disease. To identify a wider spectrum of disease-related behaviours, researchers can look at additional biomarkers, such as voice characteristics, gait analyses, and eye movements.
- **Real-Time Monitoring and Mobile Applications:** Continuous assessment of motor symptoms can be achieved by extending the scope of Parkinson's disease detection to real-time monitoring utilising wearable technology and mobile applications. These platforms can incorporate ML algorithms to provide early detection, individualised care, and remote patient management.
- **External Validation and Clinical Trials:** To confirm the efficiency and dependability of ML-based Parkinson's disease detection systems, substantial external validation studies and clinical trials involving a wide range of patients and healthcare environments must be conducted. The results may be strengthened by cooperation with medical experts and participation from several research institutes, which can also raise the system's popularity in clinical settings.
- **User-Friendly Interfaces and User Experience:** Creating user-friendly interfaces and enhancing the user experience of Parkinson's disease detection systems can increase their uptake among medical professionals. The usefulness and adoption of these technologies

may be increased by providing clear results interpretation, simple data entry methods, and intuitive visualisations.

The topic of Parkinson's disease detection using ML needs to be advanced by ongoing study, multidisciplinary cooperation, and close patient and physician participation. We can create more precise, effective, and easily available technologies to assist in the diagnosis and management of Parkinson's disease by addressing these future scopes and suggestions.

8. REFERENCES

[1]. Heisters. D, “Parkinson’s: symptoms, treatments and research”. British Journal of Nursing, 20(9), 548–554. doi:10.12968/bjon.2011.20.9.548, 2011.

[2]. Ozcift, “SVM feature selection-based rotation forest ensemble classifiers to improve computer-aided diagnosis of Parkinson disease” Journal of medical systems, vol-36, no. 4, pp. 2141-2147, 2012.

[3]. Dr. R. Geetha Ramani, G. Sivagami, Shomona Graciajacob “Feature Relevance Analysis and Classification of Parkinson’s Disease Tele Monitoring data Through Data Mining” International Journal of Advanced Research in Computer Science and Software Engineering, vol-2, Issue 3, March 2012.

[4]. Farhad Soleimanian Gharehehopogh, Peymen Mohammadi, “A Case Study of Parkinson’s Disease Diagnosis Using Artificial Neural Networks” International Journal of Computer Applications, Vol-73, No.19, July 2013.

[5]. Dragana Miljkovic et al, “Machine Learning and Data Mining Methods for Managing Parkinson’s Disease” LNAI 9605, pp. 209-220, 2016.

[6]. Arvind Kumar Tiwari, “Machine Learning based Approaches for Prediction of Parkinson’s Disease” Machine Learning and Applications: An International Journal (MLAU) vol. 3, June 2016.

[7]. Dr. Anupam Bhatia and Raunak Sulekh, “Predictive Model for Parkinson’s Disease through Naïve Bayes Classification” International Journal of Computer Science & Communication vol-9, Dec. 2017, pp. 194- 202, Sept 2017 - March 2018

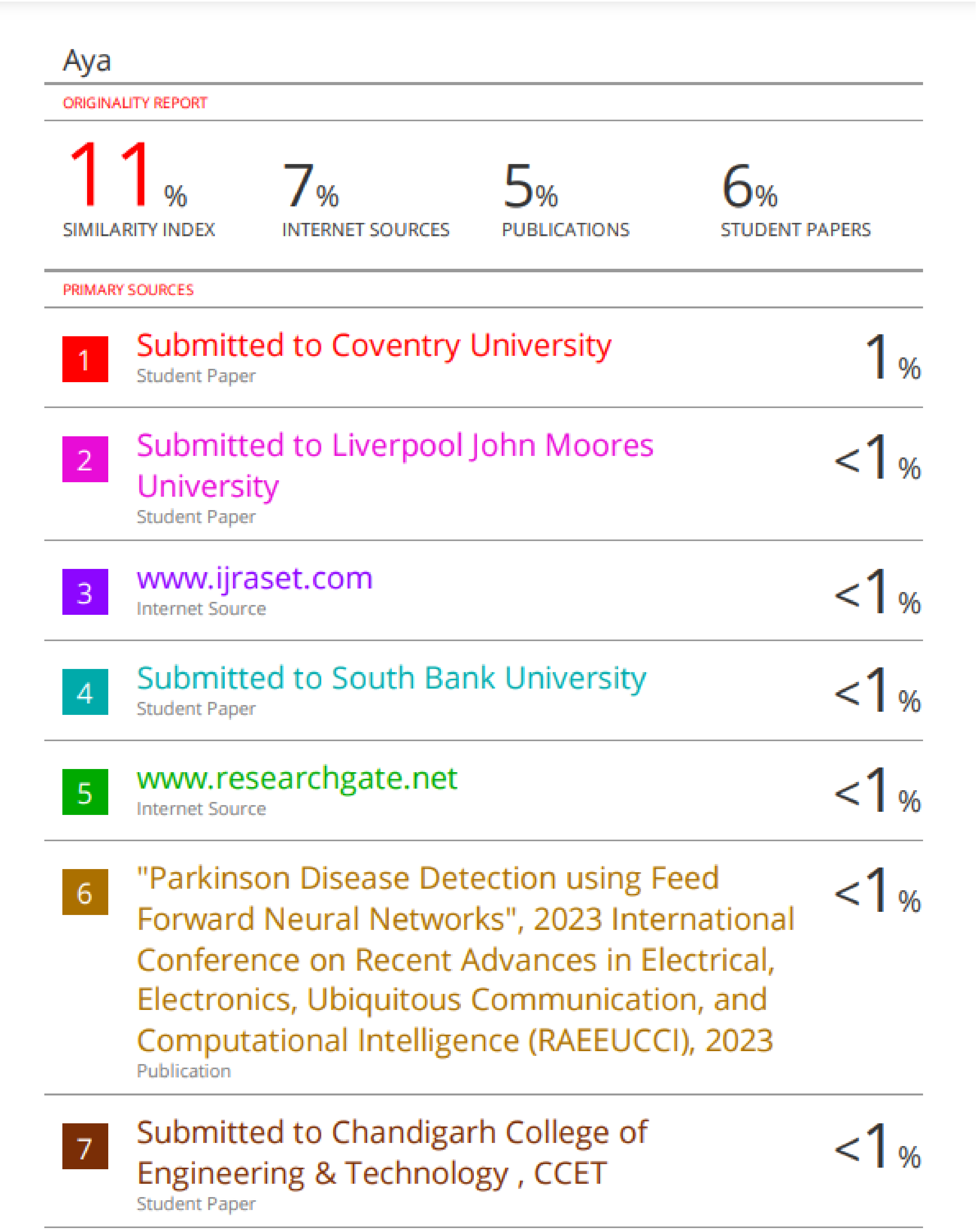
[8]. M. Abdar and M. Zomorodi-Moghadam, “Impact of Patients’ Gender on Parkinson’s disease using Classification Algorithms” Journal of AI and Data Mining, vol-6, 2018.

[9]. Carlo Ricciardi, et al, “Using gait analysis’ parameters to classify Parkinsonism: A data mining approach” Computer Methods and Programs in Biomedicine vol. 180, Oct. 2019.

[10]. Anila M Department of CS1, Dr G Pradeepini Department of CSE, “DIAGNOSIS OF PARKINSON’S DISEASE USING ARTIFICIAL NEURAL NETWORK”, JCR, 7(19): 7260-7269, 2020.

PLAGARISM REPORT OF RESEARCH PAPER

PLAGARISM REPORT OF THESIS



Aya

ORIGINALITY REPORT

11%	7%	5%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Coventry University Student Paper	1%
2	Submitted to Liverpool John Moores University Student Paper	<1%
3	www.ijraset.com Internet Source	<1%
4	Submitted to South Bank University Student Paper	<1%
5	www.researchgate.net Internet Source	<1%
6	"Parkinson Disease Detection using Feed Forward Neural Networks", 2023 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI), 2023 Publication	<1%
7	Submitted to Chandigarh College of Engineering & Technology , CCET Student Paper	<1%