

A MINOR PROJECT REPORT ON

# **ANALYSIS OF PARKINSON'S DISEASE USING MACHINE LEARNING TECHNIQUES**

A dissertation submitted in partial fulfilment of the  
Requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**  
in  
**INFORMATION TECHNOLOGY**

*Submitted by*

**N. Deepika (20B81A1211)**

**M. Harshini (20B81A1217)**

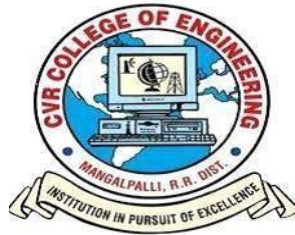
**K. Jeeva Bindhu (20B81A1219)**

*Under the esteemed guidance of*

**Mrs A. Srichandana**

Sr. Assistant Professor, IT Department

CVR College of Engineering



**DEPARTMENT OF INFORMATION TECHNOLOGY**  
**CVR COLLEGE OF ENGINEERING**

ACCREDITED BY NBA, AICTE & Affiliated to JNTU-H

Vastunagar, Mangalpally (V), Ibrahimpatnam (M), R.R. District, PIN-501 510

2023-2024



Cherabuddi Education Society's  
**CVR COLLEGE OF ENGINEERING**

(An Autonomous Institution)

**ACCREDITED BY NATIONAL BOARD OF ACCREDITATION, AICTE**

(Approved by AICTE & Govt. of Telangana and Affiliated to JNT University)

**Vastunagar, Mangalpalli (V), Ibrahimpatan (M), R.R. District, PIN - 501 510**

Web : <http://cvr.ac.in>, email : [info@cvr.ac.in](mailto:info@cvr.ac.in)

Ph : 08414 - 252222, 252369, Office Telefax : 252396, Principal : 252396 (O)

## **DEPARTMENT OF INFORMATION TECHNOLOGY**

### **CERTIFICATE**

This is to certify that the Project Report entitled “**Analysis Of Parkinson’s disease using Machine Learning Techniques**” a bonafide work done and submitted by **N. Deepika (20B81A1211), M. Harshini (20B81A1217), K. Jeeva Bindhu (20B81A1219)** during the academic year 2023-2024, in partial fulfilment of requirement for the award of Bachelor of Technology degree in Information Technology from Jawaharlal Nehru Technological University Hyderabad, is a bonafide record of work carried out by them under my guidance and supervision.

Certified further that to my best of the knowledge, the work in this dissertation has not been submitted to any other institution for the award of any degree or diploma.

#### **INTERNAL GUIDE**

**A. Srichandana**

Sr. Assistant Professor, IT Department

#### **HEAD OF THE DEPARTMENT**

**Dr Bipin Bihari Jayasingh**

HOD & Professor, IT Department

#### **PROJECT COORDINATOR**

**G. Sunitha Rekha**

Sr. Assistant Professor, IT Department

#### **EXTERNAL EXAMINER**

---

City Office : # 201 & 202, Ashoka Scintilla, Opp. KFC, Himayatnagar, Hyderabad - 500 029, Telangana.  
Phone : 040 - 42204001, 42204002, 9391000791, 9177887273

## ACKNOWLEDGEMENT

The satisfaction of completing this project would be incomplete without mentioning our gratitude towards all the people who have supported us. Constant guidance and encouragement have been instrumental in the completion of this project.

First and foremost, we thank the Chairman, Principal, Vice Principal for availing infrastructural facilities to complete the major project in time.

We offer our sincere gratitude to our internal guide **A. Srichandana**, Sr. Assistant Professor, IT Department, CVR College of Engineering for her immense support, timely cooperation and valuable advice throughout the course of our project work.

We would like to thank the Professor In-Charge of Projects, **Dr. Seetharamaiah**, Professor, Information Technology for his valuable suggestions in implementing the project.

We would like to thank the Head of Department, Professor **Dr. Bipin Bihari Jayasingh**, for his meticulous care and cooperation throughout the project work.

We are thankful to **G. Sunitha Rekha**, Project Coordinator, Sr. Assistant Professor, IT Department, CVR College of Engineering for her supportive guidelines and for having provided the necessary help for carrying forward this project without any obstacles and hindrances.

We also thank the **Project Review Committee Members** for their valuable suggestions.

## DECLARATION

We hereby declare that the project report entitled “**Analysis Of Parkinson’s disease using Machine Learning Techniques**” is an original work done and submitted to IT Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad in partial fulfilment of the requirement for the award of Bachelor of Technology in **Information Technology** and it is a record of bonafide project work carried out by us under the guidance of **A.Srichandana, Sr. Assistant Professor, Department of Information Technology.**

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other Institute or University.

---

Signature of the Student

(N.Deepika)

(20B81A1211)

---

Signature of the Student

(M.Harshini)

(20B81A1217)

---

Signature of the Student

(K.Jeeva Bindhu)

(20B81A1219)

## **ABSTRACT**

Parkinson's Disease (PD) is a neurodegenerative movement disease where the symptoms gradually start with a slight tremor in one hand, a feeling of stiffness in the body and it becomes worse over time. At present there is no conclusive result for this disease by non-specialist clinicians, particularly in the early stage of the disease where identification of the symptoms are very difficult. By using machine learning techniques, the problem can be solved with minimal error rate. Since there is no standard test to detect Parkinsonism, we propose a statistical approach using the most common symptoms of PD which are gait and tremors. This includes analyzing the correlation between the symptoms and classifying the achieved data using different classification algorithms, to find the highest accuracy in diagnosing PD patients. Our output will showcase the early detection of the disease and can help in increasing the lifespan of the patient with proper treatment and medication.

## LIST OF FIGURES

Fig-1: Architecture Diagram .....	20
Fig-2: Flow of Process .....	21
Fig-3: Class Diagram .....	22
Fig-4: Sequence Diagram .....	23
Fig-5: Use Case Diagram .....	24
Fig-6: Activity Diagram .....	25
Fig-7: KNN algorithm .....	27
Fig-8: Naive Bayes algorithm .....	28
Fig-9: Decision tree algorithm .....	29
Fig-10: Confusion Matrix Diagram .....	37
Fig-11: KNN Confusion Matrix .....	37
Fig-12: Naive Bayes Confusion Matrix .....	38
Fig-13: Decision Tree Confusion Matrix .....	38

## LIST OF TABLES

Table-1: Performance Comparison .....	40
---------------------------------------	----

# TABLE OF CONTENTS

1. Introduction	10
1.1 Literature Survey	13
1.2 Existing Works	15
1.3 Limitations to Existing Works	16
2. System Requirement Specifications	17
2.1 Functional & Non-Functional Requirements	17
2.2 Hardware Requirements	19
2.3 Software Requirements	19
2.4 Technical Description	20
3. Design	21
4. Implementation	26
4.1 K-Nearest Neighbor	27
4.2 Naïve Bayes	28
4.3 Decision Tree	29
4.4 About the dataset	30
4.5 Training and Testing Code Snippets	32
5. Evaluate Model on Test Data	33
5.1 Confusion Matrix	36
5.2 Accuracies	40
5.3 Line graph	41
5.4 Bar graph	42
5.5 Tools	43
5.6 Technology Used	44



Conclusion .....	45
Future Enhancements .....	46
References .....	47
Appendix .....	48

# 1. INTRODUCTION

The realm of medical diagnostics stands at the precipice of a transformative era, propelled by the synergy of medical expertise and cutting-edge technological advancements. Amid this paradigm shift, the early and accurate detection of Parkinson's Disease (PD) emerges as a pressing challenge. PD, a neurodegenerative disorder with debilitating consequences, necessitates timely diagnosis to optimize patient outcomes. However, the elusive nature of early symptoms, coupled with the complexity of their interplay, has posed formidable barriers to non-specialist clinicians.

In response, this study pioneers a novel approach that leverages the formidable capabilities of machine learning algorithms to revolutionize PD diagnosis. Drawing inspiration from the hallmark symptoms of the disease—gait abnormalities and tremors—the study delves into the intricate web of symptom correlations, seeking to decode the underlying patterns that signify the onset of PD.

The algorithms of choice for this inquiry—K-Nearest Neighbors (KNN), Naive Bayes, and Decision Tree—epitomize the fusion of computational prowess and medical insight. KNN harnesses proximity analysis to classify patients based on symptom correlations, while Naive Bayes thrives on probabilistic independence to discern intricate relationships. The Decision Tree algorithm navigates the symptom landscape with a systematic flowchart approach, uncovering crucial diagnostic pathways.

This exploration of algorithms poised at the frontier of machine learning is not merely a technological endeavour—it embodies a profound quest to enhance human health. By synthesizing the intricate dance between data and diagnosis, this study endeavours to unlock a new chapter in PD diagnostics. Through the seamless integration of algorithmic sophistication and clinical acumen, the path towards early, accurate, and impactful PD diagnosis stands illuminated, offering a glimmer of hope to patients and clinicians alike.

## MOTIVATION

The motivation behind this project stems from the urgent need for early and accurate diagnosis of Parkinson's Disease (PD). PD is a progressive neurodegenerative disorder that often begins with subtle symptoms, making early detection challenging, especially for non-specialist clinicians. The initial manifestations, such as tremors and gait abnormalities, can be easily overlooked or misdiagnosed, leading to delayed intervention and compromised patient outcomes.

The potential of machine learning techniques to revolutionize medical diagnostics has inspired this endeavour. Leveraging the power of advanced algorithms, we aim to bridge the diagnostic gap and provide a reliable tool for identifying PD at an early stage. By developing a statistical model that correlates common symptoms like gait and tremors, we strive to enhance the accuracy of PD diagnosis.

Through our project, we seek to minimize diagnostic errors and enable non-specialist clinicians to confidently identify PD cases even in its nascent phases. Early detection holds immense promise in extending patients' lifespans and improving their quality of life through timely treatment and medication. By combining medical expertise with cutting-edge technology, we aspire to contribute to the advancement of PD diagnostics, aligning with the overarching goal of enhancing healthcare outcomes and fostering a healthier society.

## **PROBLEM STATEMENT**

The problem statement for this project revolves around the need for accurate and early diagnosis of Parkinson's Disease (PD) using machine learning techniques. Despite its debilitating impact, PD often presents with subtle symptoms that are difficult to detect, particularly in its initial stages. This challenge is amplified when non-specialist clinicians attempt to identify PD cases, as they may lack the expertise to discern these early indicators.

Current diagnostic methods fall short in reliably differentiating PD from other conditions, leading to delayed interventions and compromised patient well-being. There is a pressing need for a comprehensive and accessible solution that empowers non-specialist clinicians with a tool capable of accurately diagnosing PD at an early phase.

The proposed project seeks to address this challenge by harnessing the potential of machine learning. By analyzing and correlating common PD symptoms such as gait abnormalities and tremors, the project aims to create a statistical model capable of effectively classifying PD cases. The primary objective is to design an accurate and efficient diagnostic tool that can assist nonspecialist clinicians in identifying PD cases, enabling timely intervention, proper treatment, and improved patient outcomes.

In summary, the problem statement involves creating a machine learning-based solution to enhance the accuracy of PD diagnosis by enabling non-specialist clinicians to recognize early symptoms, thereby contributing to better patient care and overall healthcare effectiveness.

## PROBLEM OBJECTIVE

The objectives of this project are to:

**Enhance Early Detection:** Develop a machine learning-driven diagnostic system that can identify subtle and early symptoms of Parkinson's Disease (PD), enabling timely intervention and treatment.

**Minimize Diagnostic Errors:** Create a reliable and accurate classification model that reduces the risk of misdiagnosis by non-specialist clinicians, thereby improving the overall accuracy of PD detection.

**Utilize Symptom Correlation:** Analyze the correlation between common PD symptoms, such as gait abnormalities and tremors, to establish robust patterns and features for effective classification.

**Evaluate Classification Algorithms:** Implement and compare various classification algorithms to identify the most suitable approach for accurately diagnosing PD cases.

**Optimize Model Performance:** Fine-tune the machine learning model parameters and features to achieve the highest possible accuracy in PD classification.

**User-Friendly Interface:** Design an intuitive and user-friendly interface that allows non-specialist clinicians to easily input patient data and receive accurate PD diagnostic results.

**Contribute to Healthcare:** Provide a valuable tool that empowers non-specialist clinicians with the ability to confidently diagnose PD at an early stage, ultimately leading to improved patient care and outcomes.

**Ethical Considerations:** Address ethical implications related to patient data privacy, informed consent, and responsible deployment of the diagnostic system.

**Documentation and Sharing:** Document the development process, methodology, and findings for future reference and potential integration into healthcare systems.

**Contribute to Medical Research:** Generate insights into the correlation between PD symptoms and their significance in early diagnosis, potentially advancing medical understanding of the disease.

## 1.1 LITERATURE SURVEY

Arvind Kumar Tiwari Proposed the paper titled “Machine Learning-based Approaches for Prediction of Parkinson’s Disease”[1], In this paper, minimum redundancy maximum relevance feature selection algorithms were used to select the most important feature among all the features to predict Parkinson diseases. Here, it was observed that the random forest with 20 number of features selected by minimum redundancy maximum relevance feature selection algorithms provide the overall accuracy 90.3%, precision 90.2%, Mathews correlation coefficient values of 0.73 and ROC values 0.96 which is better in comparison to all other machine learning based approaches such as bagging, boosting, random forest, rotation forest, 10 random subspace, support vector machine, multilayer perceptron, and decision tree based methods.

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

T. J. Wroge, Y. Özkanca, C. Demiroglu, D. Si, D. C. Atkins and R. H. Ghomi, proposed the paper titled “Parkinson’s Disease Diagnosis Using Machine Learning and Voice” [3] is that it explores the effectiveness of using supervised classification algorithms, such as deep neural networks, to accurately diagnose individuals with the disease. Historically, PD has been difficult to quantify and doctors have tended to focus on some symptoms while ignoring others, relying primarily on subjective rating scales. The analysis of this paper provides a comparison of the effectiveness of various machine learning classifiers in disease diagnosis with noisy and high dimensional data. Their peak accuracy of 85% provided by the machine learning models exceeds the average clinical diagnosis accuracy of non-experts (73.8%) and average accuracy of movement disorder specialists (79.6% without follow-up, 83.9% after follow-up) with pathological post-mortem examination as ground truth.

Siva Sankara Reddy Donthi Reddy and Udaya Kumar Ramanadham proposed the paper “Prediction of Parkinson’s Disease at Early Stage using Big Data Analytics” [4]. This paper describes mainly various Big Data Analytical techniques that may be used in diagnosing of right disease in the right time. The main intention is to verify the accuracy of prediction algorithms.

Their future study aims to propose an efficient method to diagnose this type of neurological disorder by some symptoms at the early stage with better accuracy.

Daiga Heisters proposed the paper titled “Parkinson’s: symptoms, treatments and research” [5]. This paper initially says that Current treatments can help to ease the symptoms but none can repair the damage in the brain or slow the progress of the condition; now, Parkinson’s UK researchers are working to develop new treatments that can and finally worked together to build on existing discoveries and explore these innovative areas of research, it is hoped that a cure for Parkinson’s will be found. Parkinson’s UK offers support for everyone affected,, including people with the condition, their family, friends and careers, researchers and professionals working in this area.

T. Swapna, Y.Sravani Devi proposed a paper and titled “Performance Analysis of Classification algorithms on Parkinson’s Dataset with Voice Attributes” [6]. This paper deals with the application of seven classification algorithms on the acquired data set and then drawing out a comparison of the results to one another and also predicting the outcome whether the person is healthy or Parkinson disease effected from the given data. The results of the selected algorithms namely Naïve Bayes, Random Forest, Neural Networks, Decision Trees, AdaBoost, SVM, KNN were compared and tabulated. According to the outputs derived with the help of python, implementing Scikit Libraries. Final accuracy was calculated using these parameters. Random Forest algorithm gives with optimum accuracy of 78.56% which is closely followed by Decision Tree Algorithm with the optimal accuracy of 77.63%. Following the Decision Tree Algorithm is the MLP Classifier with an optimal accuracy of 76.72%, and lastly the Naïve Bayes Algorithm which has the optimal accuracy of 70.82%. Finally, these algorithms can help in classifying whether a person get effected with Parkinson’s disease or not.

Sriram,T.V., proposed a paper “Intelligent Parkinson Disease Prediction Using Machine Learning Algorithms” [7]. In this paper, the author used voice measures of the patients to check whether the patient has Parkinson’s or not. The author applied the dataset to various machine learning algorithms and find the maximum accuracy. To analyse the models the author used the ROC curve and sieve graph. The random forest results with more accuracy i.e. are 90.26%.

Dragana Miljkovic, et al, proposed a paper “Machine Learning and Data Mining Methods for Managing Parkinson’s Disease” [8]. In this paper, the author concluded that based on the medical tests taken by the patients the Predictor part was able to predict the 15 different Parkinson’s symptoms separately. The machine learning and data mining techniques are applied on different symptoms separately and gives an accuracy range between 57.1% and 77.4% where tremor detection has the highest accuracy.

Dr. R.GeethaRamani, G.Sivagami, and ShomonaGraciajacob proposed a paper “Feature Relevance Analysis and Classification of Parkinson’s Disease TeleMonitoring data Through Data Mining” [9]. In this paper, the author used thirteen classification algorithms to diagnose the disease. The author used the Tele-monitoring dataset which contains 16 biomedical voice features for evaluating the system. The aim of this paper is to predict motor UPDRS and total UPDRS from the voice measures.

A. Ozcift, proposed a paper “SVM feature selection based rotation forest ensemble classifiers to improve computer-aided diagnosis of Parkinson 12 disease” [10]. In this paper, the author summarizes that improve the PD diagnosis accuracy with the use of support vector machine feature selection. To evaluate the performances the author used accuracy, kappa statistics, and area under the curve of the classification algorithms. The rotation Forest ensemble of these classifiers used to increase the performance of the system.

## **1.2 EXISTING WORKS**

In the realm of Parkinson's Disease diagnosis using machine learning, existing research has made significant strides in advancing our understanding and capabilities. Several noteworthy studies have delved into the application of machine learning techniques to enhance the early detection and accurate classification of Parkinson's Disease.

One such study titled "Automated Diagnosis of Parkinson’s Disease Using Genetic Programming" by Costa Abecasis and colleagues (2016) harnesses the power of genetic programming to automate the diagnosis process. Their work underscores the potential of machine learning algorithms in effectively discerning Parkinson's Disease patients based on motor symptom data.

Another notable contribution, "Parkinson’s Disease Detection Using Envelope Analysis of Tremor" by Roy et al. (2016), investigates the viability of envelope analysis of tremor signals for diagnosis. Employing machine learning methods, they extract pertinent features and successfully classify Parkinson's Disease patients based on distinctive tremor characteristics.

### 1.3 LIMITATIONS

Selecting pertinent features from intricate datasets, such as those containing speech or movement patterns, can be difficult due to their complexity. Expertise in the specific domain is often necessary to discern which features hold critical information, ensuring that the machine learning model focuses on the most relevant aspects of the data.

With a shortage of data, machine learning models can become overly tailored to the training dataset, leading to overfitting. This phenomenon results in the model fitting noise in the training data, causing it to struggle when confronted with new, unseen data, and potentially compromising its ability to make accurate predictions.

Engaging with sensitive medical data requires addressing ethical considerations related to patient privacy, data security, and informed consent. Ensuring responsible handling of such data is imperative to protect patients' rights and maintain ethical integrity in the use of machine learning for medical analysis.

Models developed using a specific dataset may struggle to accurately apply their insights to diverse populations or different medical contexts. This limitation underscores the importance of creating adaptable models that can effectively address variations in patient characteristics and healthcare environments. Imbalanced datasets, where one class is underrepresented, can lead to biased predictions and inaccurate results.



## **2. SYSTEM REQUIREMENT SPECIFICATIONS**

### **Hardware requirements:**

- CPU: A quad-core CPU or higher is preferred
- Processor: 11<sup>th</sup> Gen, Intel core i5 or higher / AMD Ryzen 5 or higher
- Architecture: 64-bit
- RAM: 16 threads,16GB memory or higher

### **Software requirements:**

- Operating System: Windows 8.1-1.
- Packages: Pandas, Numpy, Matplotlib
- Programming Language: Python-3.10.1
- Framework: Jupyter Notebook (Offline), Google Colab (Online)

## **2.1 FUNCTIONAL & NON-FUNCTIONAL REQUIREMENTS**

### **Functional Requirements:**

#### **Data Collection and Storage:**

The system shall collect and store patient data, including symptoms and medical history. It shall support the integration of diverse data sources, such as tremor measurements and gait analysis.

#### **Feature Extraction:**

The system shall extract relevant features from the collected data, including tremor patterns and gait characteristics. It shall employ signal processing techniques to preprocess raw data and derive meaningful features.

**Machine Learning Model Development:**

The system shall develop and train machine learning models for Parkinson's Disease prediction. It shall implement algorithms such as KNN, Naive Bayes, and Decision Trees for classification.

**Model Evaluation and Validation:**

The system shall assess model accuracy using appropriate evaluation metrics (e.g., accuracy, precision, recall). It shall validate the model using cross-validation and external datasets to ensure generalizability.

**Real-time Prediction:**

The system shall provide real-time predictions based on input symptoms and data. It shall offer immediate feedback to healthcare professionals and patients.

**User Interface:**

The system shall have an intuitive user interface for data input and visualization of predictions. It shall support easy navigation and accessibility for both clinicians and patients.

**Non-Functional Requirements:****Performance:**

The system shall provide accurate predictions with minimal latency, even for large datasets. It shall handle concurrent user requests efficiently without degradation in response time.

**Accuracy:**

The prediction models shall achieve a minimum accuracy of 90% on benchmark datasets. The system shall continuously monitor and improve prediction accuracy over time.

**Security and Privacy:**

Patient data shall be encrypted during storage and transmission to ensure confidentiality. The system shall comply with relevant data protection regulations (e.g., HIPAA, GDPR).

**Scalability:**

The system architecture shall be designed to accommodate increasing data volume and user demand. It shall scale horizontally to handle additional patient data and future enhancement.

## 2.2 HARDWARE REQUIREMENTS

### **Computer System:**

A computer system with a minimum of quad-core processor (e.g., Intel Core i5 or equivalent). Sufficient RAM (8 GB or more) to accommodate data processing and machine learning tasks.

### **Storage Space:**

Adequate storage space (at least 100 GB) for storing datasets, preprocessed data, and trained models.

### **Input Devices:**

Keyboard and mouse for data input and interaction with the user interface.

### **Internet Connectivity:**

Stable internet connection for accessing external resources, data, and potential updates.

## 2.3 SOFTWARE REQUIREMENTS

### **Operating System:**

Windows 10 for hosting the development environment.

### **Development Environment:**

Python (version 3.7 or later) as the primary programming language for data processing and model development. Integrated Development Environment (IDE) such as Anaconda, Jupyter Notebook, or Visual Studio Code for coding and experimentation.

### **Libraries and Frameworks:**

Essential Python libraries for data manipulation (e.g., pandas), machine learning (e.g., scikitlearn), and signal processing (e.g., scipy).

### **Data Visualization:**

Matplotlib, Seaborn, or other visualization libraries for generating graphs and plots.

## Documentation and Reporting:

LaTeX or Microsoft Word for creating project documentation, reports, and research papers.

## 2.4 SOFTWARE ARCHITECTURE

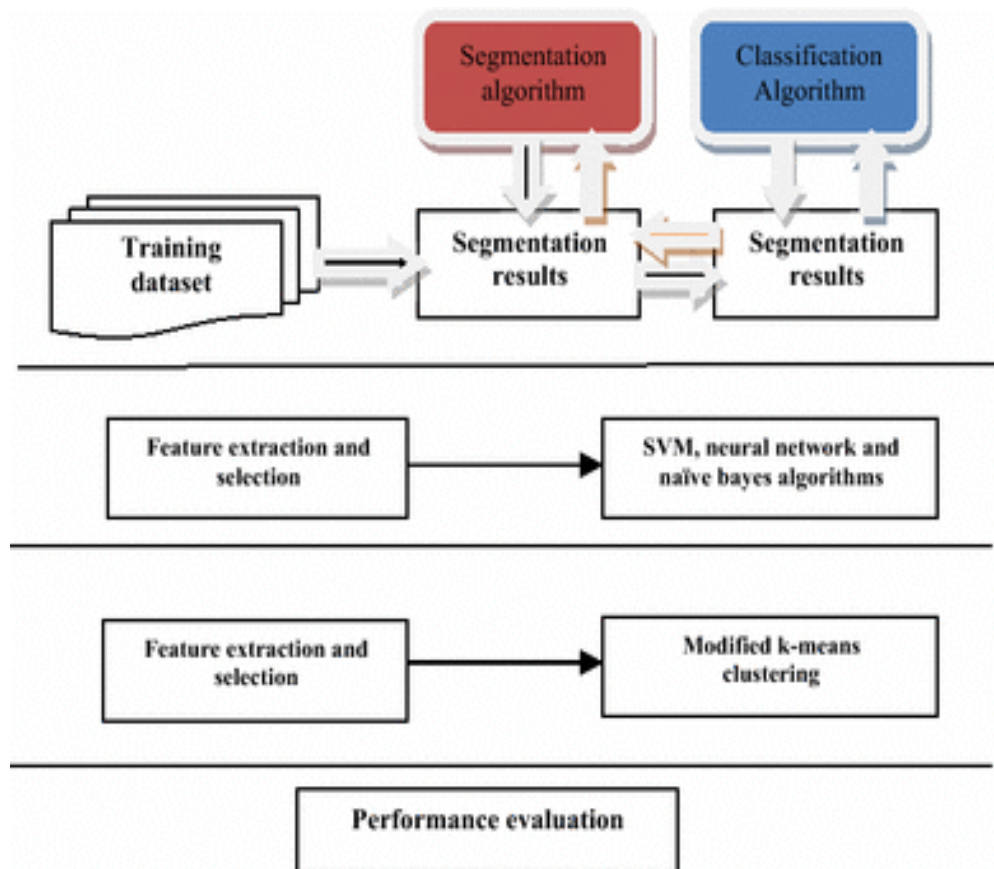
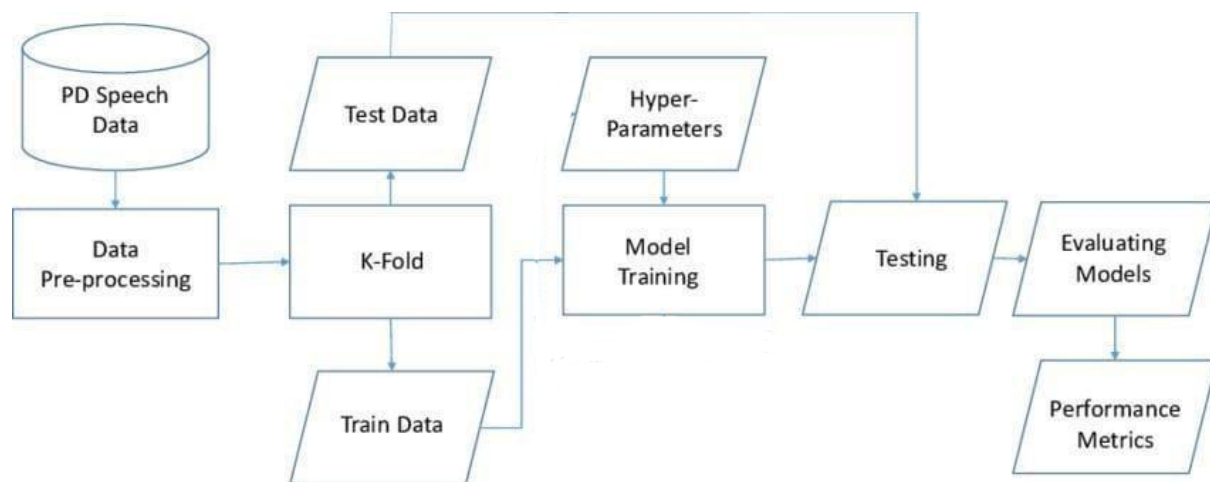


Figure-1: Architecture Diagram

### 3. DESIGN

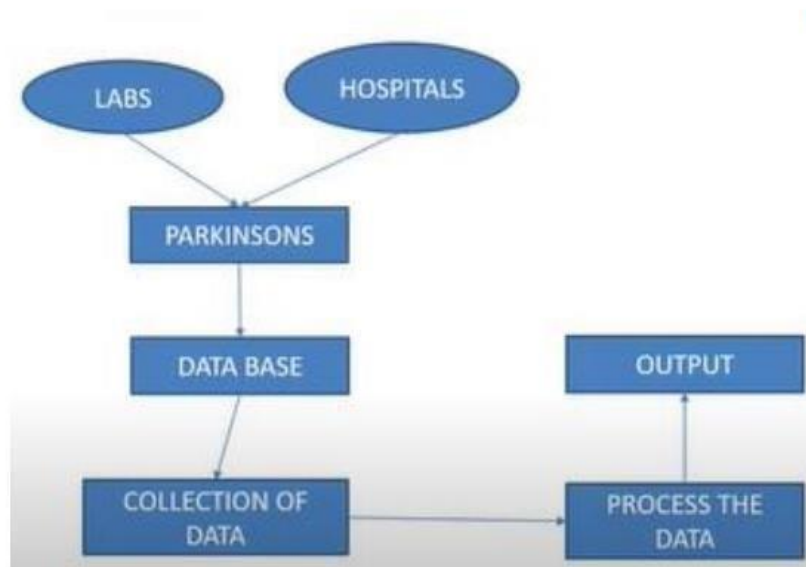
A Data Flow Diagram (DFD) is a visual representation that illustrates the flow of data within a system. It shows how data moves through various processes and interactions between different components. Below is a simple example of a Data Flow Diagram.



**Figure-2: Flow of processes**

## CLASS DIAGRAM

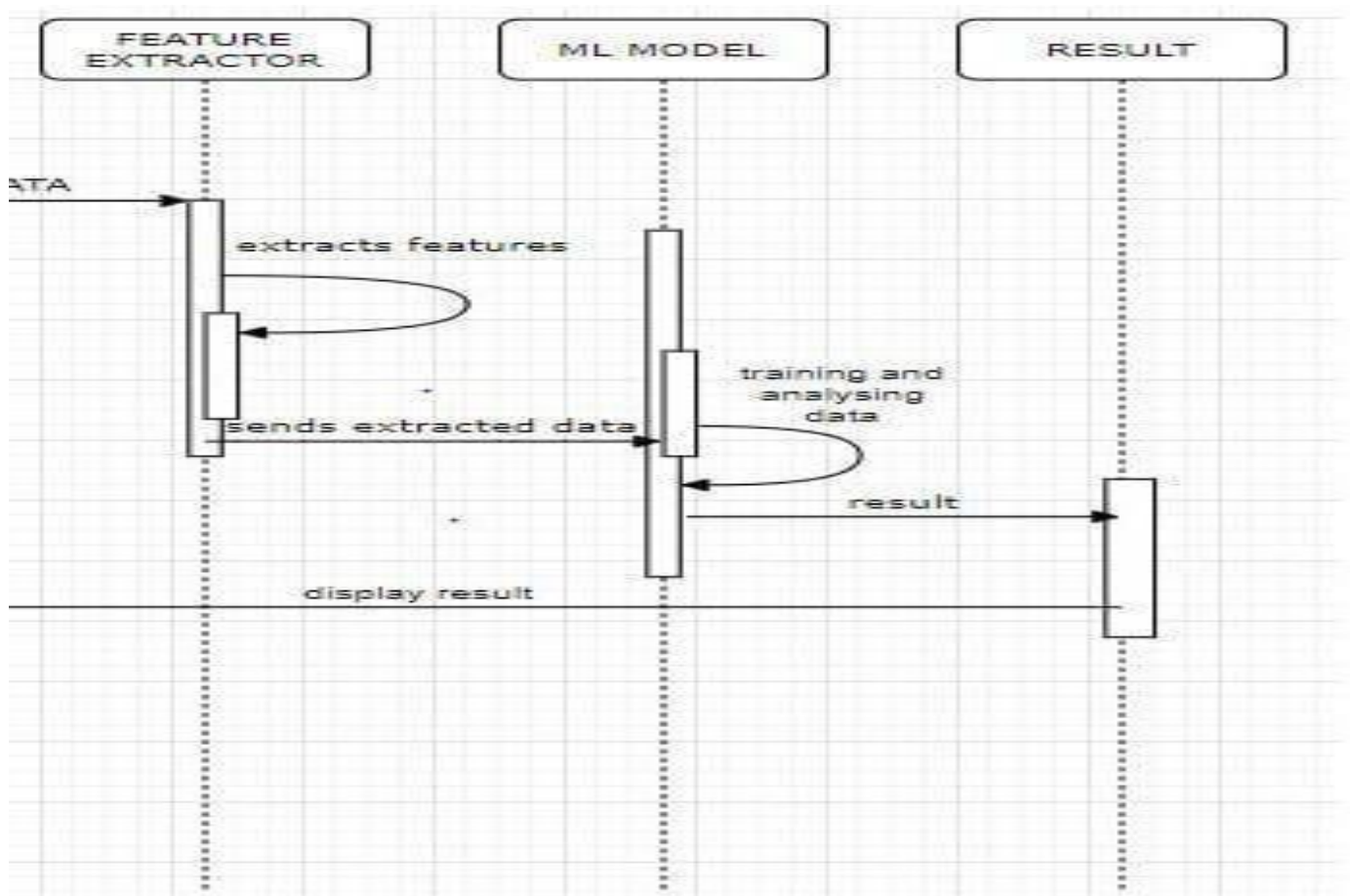
A class diagram in Unified Modeling Language (UML) serves as a visual blueprint for system structure. In the context of a user system and dataset, it comprises classes like "User" and "Dataset," each with attributes (e.g., "username" and "name") defining their properties. Relationships are illustrated through lines connecting classes, such as the ownership relationship between "User" and "Dataset." Multiplicity indicates the number of objects involved in a relationship, specifying, for instance, that a "User" can own multiple "Datasets" (one-to-many). Associations show how classes are related, highlighting that a "User" is associated with a "Dataset" through ownership. Class diagrams are a critical tool for system design and modeling, offering a clear and structured representation of a system's components and their relationships.



**Figure-3: Class diagram**

## SEQUENCE DIAGRAM

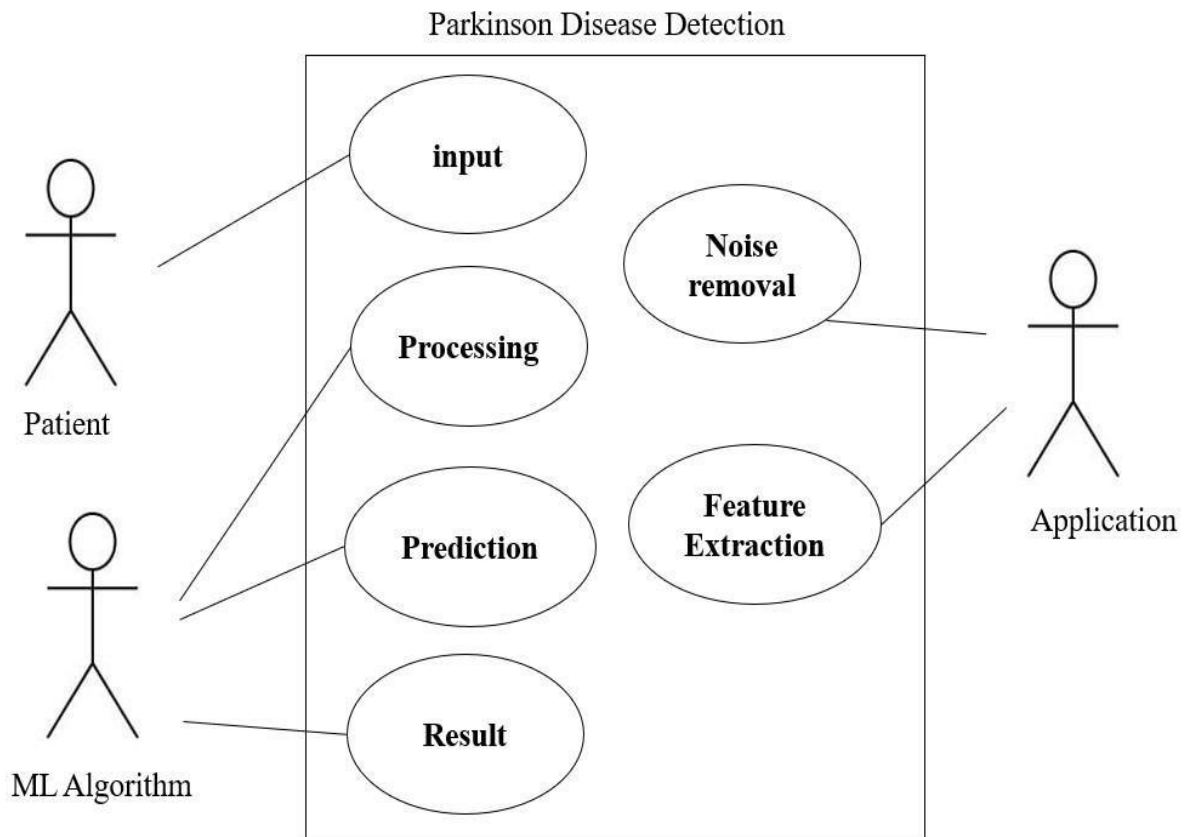
The process begins with the initiation of the evaluation, with the goal of assessing the model's predictive performance for liver disease. Liver disease-related data is loaded and meticulously preprocessed. This includes tasks like data cleaning, normalization, and feature selection to ensure the dataset is ready for model training. The dataset is logically divided into training and testing subsets, essential for model development and performance assessment. A specialized model tailored for liver disease prediction is trained using the prepared training data. The trained model is rigorously tested on the separate testing dataset to gauge its predictive capabilities and generalization. Key evaluation metrics, including accuracy, precision, recall, and F1 score, are meticulously computed based on the model's predictions and actual outcomes.



**Figure-4: Sequence Diagram**

## USE CASE DIAGRAM

Actors represent entities interacting with the system. They can include healthcare professionals, patients, and the prediction system itself. Use cases depict specific functions or actions the system can perform, such as "Inputting Patient Data," "Predicting Liver Disease," and "Displaying Results." Arrows connecting actors and use cases illustrate how actors initiate or participate in specific actions, showcasing the flow of interactions. The diagram delineates the system's boundaries, clarifying what is within and outside the scope of the liver disease prediction system. The use case diagram offers a concise and understandable overview of the system's functionality, aiding in communication among stakeholders and ensuring that all relevant interactions and features are considered during system development.



**Figure-5: Use case Diagram**



## ACTIVITY DIAGRAM

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. We gather the relevant medical data, including patient demographics, blood test results, and medical history. Clean the data by handling missing values, outliers, and noise. Normalize or scale features if necessary. Identify the most important features that contribute to liver disease prediction. This step helps in reducing dimensionality. Divide the dataset into training and testing sets. The training set is used to train the machine learning model, while the testing set is used to evaluate its performance. Choose an appropriate machine learning algorithm (e.g., logistic regression, random forest, or support vector machine) and train the model on the training data. Evaluate the trained model on the testing dataset using metrics such as accuracy, precision, recall, F1 score.

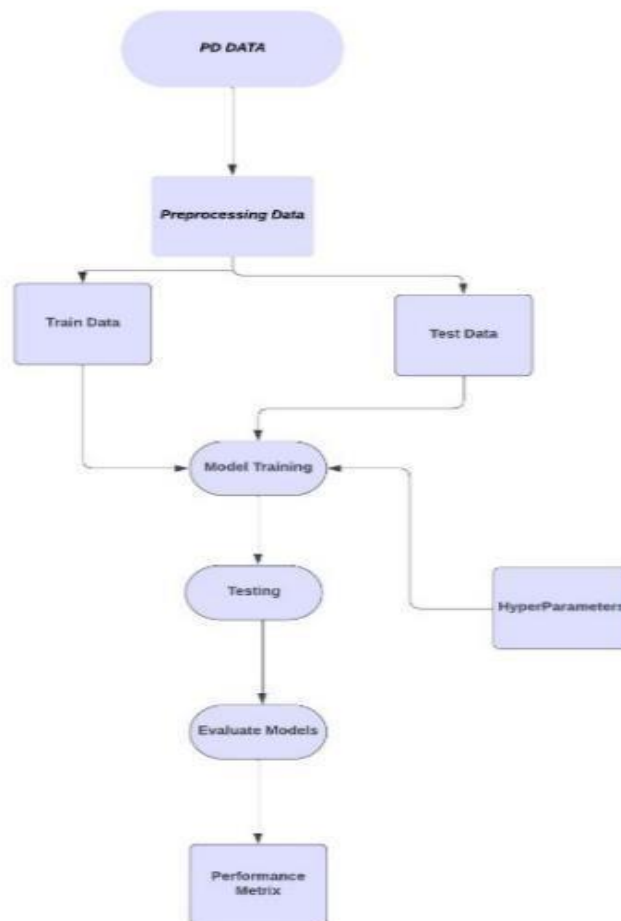


Figure-6: Activity Diagram

## 4. IMPLEMENTATION

Parkinson's disease prediction using machine learning algorithms involves several steps to build accurate and robust models for early detection. First, the dataset is loaded and pre-processed by handling missing data and splitting it into training and testing sets. Feature scaling is applied to ensure fair comparisons between algorithms, especially for distance based models like KNN . Subsequently, a variety of machine learning models, such as KNN, Naive Bayes, Decision Trees are trained on the scaled training data.

Each model's performance is evaluated using key metrics like accuracy, precision, recall, F1score. on the separate testing set. This evaluation helps to identify the model that best suits the specific requirements of the Parkinson's disease prediction task. Careful consideration is given to choose a model with high accuracy and good balance between precision and recall, as both false positives and false negatives can have significant consequences in medical diagnosis.

Once the best model is selected, it can be deployed in real-world scenarios to predict whether an individual is at risk of Parkinson's disease or not based on their input features. Early detection of Parkinson's disease can lead to timely medical interventions, improving patient outcomes and reducing the burden on healthcare systems. Regular updates and re-evaluation of the model's performance are crucial to maintain its effectiveness as new data becomes available and the healthcare landscape evolves. Overall, employing machine learning algorithms for Parkinson's disease prediction holds great promise in advancing healthcare and promoting better health.

In this project different algorithms were used are-

- KNN Algorithm
- Naïve Bayes Algorithm
- Decision Tree Algorithm

## 4.1 K-NEAREST NEIGHBOR ALGORITHM

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique. In KNN, all the available data is retained during the training phase, forming the basis for classification during the prediction phase. When a new data point is encountered, KNN looks at its k-nearest neighbor's (i.e., the k data points in the training set that are closest to the new data point) based on a chosen distance metric. These neighbors' cast votes for the category they belong to, and the new data point is assigned to the category with the highest number of votes. KNN can be applied to both regression and classification problems, but it is more commonly used for classification tasks. In regression, the output is predicted as the average of the K-nearest neighbors target values. One of the notable characteristics of KNN is that it is a **non-parametric** algorithm, meaning it makes no assumptions about the underlying data distribution. Instead, it relies solely on the data.

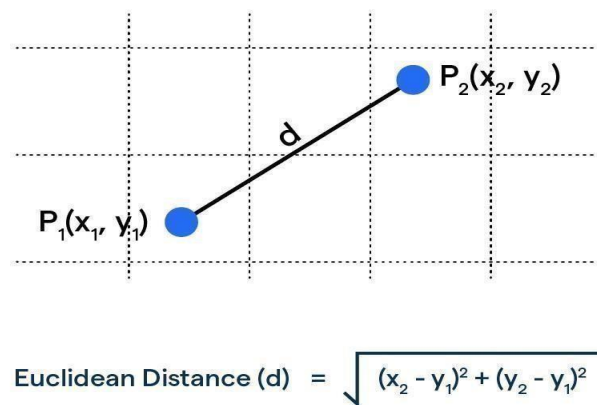


Figure-7: Formula to be used in knn algorithm

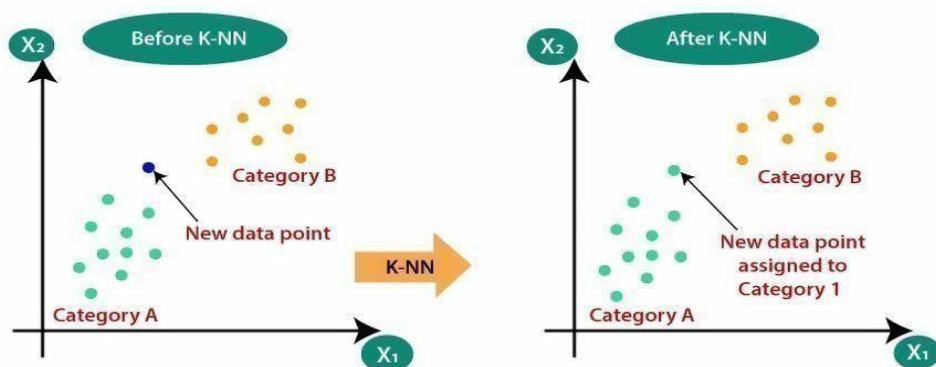
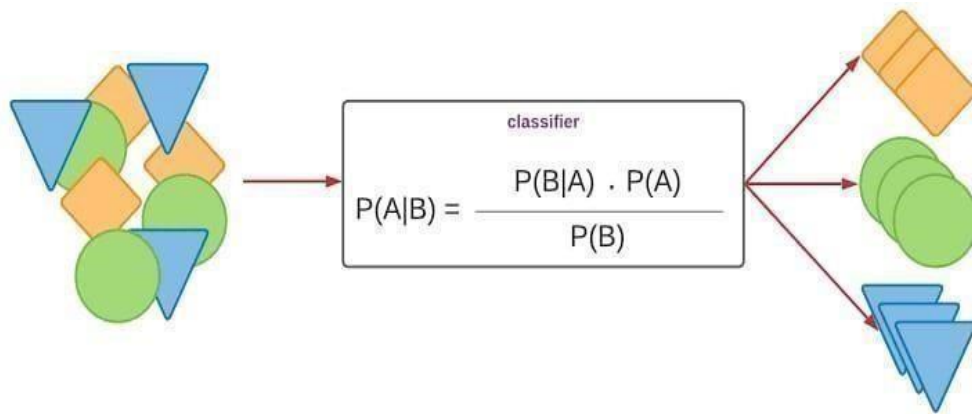


Figure-8: Knn algorithm example

## 4.2 NAIVE BAYES ALGORITHM

The Naive Bayes algorithm is a supervised learning technique rooted in Bayes' theorem, primarily utilized for classification tasks. Its application extends to various domains, with notable use cases in text classification involving high-dimensional training datasets. This algorithm, characterized by its simplicity and effectiveness, excels at swiftly constructing machine learning models capable of making quick predictions. It operates as a probabilistic classifier, basing its predictions on the probability of an object belonging to a particular class. Some prominent applications of the Naive Bayes Algorithm include spam filtration, sentiment analysis, and article classification. This versatility and efficiency make Naive Bayes a valuable tool in the field of machine learning and data analysis.



**Figure-9: Formula to be used in Navie bayes algorithm**

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

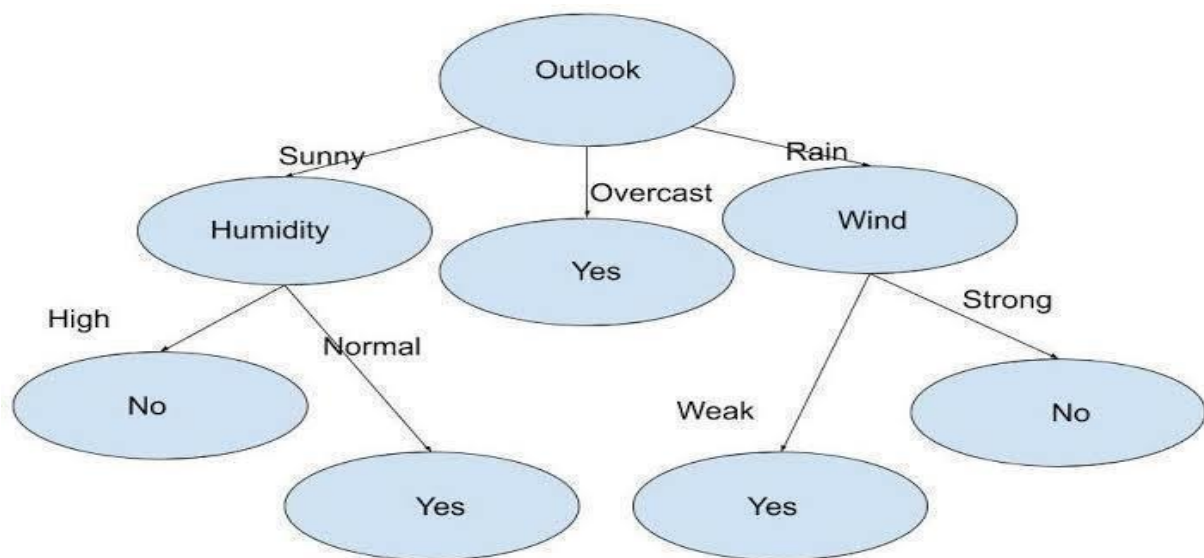
### 4.3 DECISION TREE ALGORITHM

Decision tree induction is the learning of decision trees from class-labelled training tuples. A decision tree is a flowchart-like tree structure. Decision tree induction is a non-parametric approach for building classification models. Finding an optimal decision tree is an NP complete problem. Techniques developed for constructing decision trees are computationally inexpensive, making it possible to construct models even when the training set size is very large. Decision trees, especially smaller-sized trees, are relatively easy to interpret. Decision trees provide an expressive representation for learning discrete valued functions.

The presence of redundant attributes does not adversely affect the accuracy of the decision tree. The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery.

Decision trees can handle high dimensional data. Their representation of acquired knowledge in tree form is intuitive and generally easy to assimilate by humans. The learning and classification steps of decision tree induction are simple and fast.

In general, decision tree classifiers have good accuracy. Decision tree induction algorithms have been used for classification in many application areas, such as medicine, manufacturing and production, financial analysis, astronomy, and molecular biology.



**Figure-10: Decision Tree Example**

## 4.4 ABOUT THE DATASET

In the context of your Parkinson's Disease prediction project, the dataset serves as a foundational component, providing the necessary input for training and evaluating machine learning models. It comprises a collection of structured data points representing various attributes and symptoms associated with Parkinson's Disease patients. The dataset is instrumental in enabling the development of accurate predictive models for early diagnosis and intervention.

### **Key Aspects of the Dataset:**

**Data Sources:** The dataset is sourced from reputable medical institutions, research databases, or clinical studies that have collected relevant patient information over a period of time.

**Attributes:** Each data point in the dataset corresponds to a patient and includes a set of attributes that characterize their symptoms, medical history, and demographic information. These attributes may encompass gait measurements, tremor patterns, age, gender, and other relevant factors.

**Data Variability:** The dataset captures the diverse manifestations of Parkinson's Disease, including variations in symptom severity, disease progression, and patient profiles. This variability is essential for training robust and generalizable machine learning models.

**Data Preprocessing:** Prior to model development, the dataset undergoes preprocessing steps to clean, normalize, and transform the raw data. This ensures that the data is consistent, relevant, and suitable for analysis.

**Feature Extraction:** From the raw data, meaningful features are extracted that encapsulate essential information for disease prediction. These features contribute to the model's ability to recognize patterns and make accurate predictions.

**Data Splitting:** The dataset is typically divided into training, validation, and testing subsets. The training set is used to train the machine learning models, while the validation and testing sets are used to fine-tune and assess the models' performance, respectively.

Number of Instances: 197

Number of Attributes: 23

Missing Values? N/A

**Source:**

The dataset was created by Max Little of the University of Oxford, in collaboration with the National Centre for Voice and Speech, Denver Colorado, who recorded the speech signals. The original study published the feature extraction methods for general voice disorders.

**Data Set Information:**

This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). Each column in the table is a particular voice measure, and each row corresponds one of 195 voice recording from these individuals ("name" column). The main aim of the data is to discriminate healthy people from those with PD, according to "status" column which is set to 0 for healthy and 1 for PD.

The data is in ASCII CSV format. The rows of the CSV file contain an instance corresponding to one voice recording. There are around six recordings per patient, the name of the patient is identified in the first column.

**Attribute Information:**

Matrix column entries (attributes):

name - ASCII subject name and recording number MDVP:Fo(Hz)

- Average vocal fundamental frequency

MDVP:Fhi(Hz) - Maximum vocal fundamental frequency

MDVP:Flo(Hz) - Minimum vocal fundamental frequency

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

MDVP:Shimmer,MDVP:Shimmer(dB),Shimmer:APQ3,Shimmer:APQ5,MDVP:

APQ,Shimmer: DDA - Several measures of variation in amplitude

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

- 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

## 4.5 TRAINING AND TESTING CODE SNIPPETS

The function in the code snippet below will do exactly the step defined above.

```
from sklearn.preprocessing import
StandardScaler import pandas as pd import
numpy as np import seaborn as sns import
matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler from
sklearn.model_selection import train_test_split from
sklearn.metrics import confusion_matrix, classification_report

from sklearn.neighbors import
KNeighborsClassifier from sklearn.tree import
DecisionTreeClassifier from sklearn.naive_bayes
import GaussianNB df =
pd.read_csv("parkinsons.csv")

df.head()

df_train = df.copy().drop(columns=["name"])
col_names = df_train.columns.tolist() target_col
= ["status"] col_names.remove(target_col[0])
df_train
= df_train[col_names + target_col] std =
StandardScaler()

scaled = std.fit_transform(df_train[col_names]) # Standardize the columns to get them
on the same scale scaled = pd.DataFrame(scaled, columns=col_name
```



### KNN CODE SNIPPET:

```
conf_matx = {}

knn = KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='manhattan',
metric_params=None, n_neighbors = 2, weights='distance')

knn.fit(train_X,train_y)                # Fit the training data set to the algorithm passed.

predictions = knn.predict(test_X)        # Get all predictions

print("Classification report:")           # Print the classification report
print(classification_report(test_y, predictions))

conf_matrix = confusion_matrix(test_y, predictions) # Get confusion matrix using the
predictions tn, fp, fn, tp = conf_matrix.ravel() conf_matx['K-Nearest
Neighbours'] = conf_matrix
```

### NAVE BAYES CODE SNIPPET:

```
gnb = GaussianNB(priors=None, var_smoothing=1e-09)

gnb.fit(train_X,train_y)                # Fit the training data set to the algorithm passed.

predictions = gnb.predict(test_X)        # Get all predictions

print("Classification report:")           # Print the classification report
print(classification_report(test_y, predictions))

conf_matrix = confusion_matrix(test_y, predictions) # Get confusion matrix using the
predictions tn, fp, fn, tp = conf_matrix.ravel()
```

```
conf_matx['Naive Bayes'] = conf_matrix
```

#### DECISION TREE CODE SNIPPET :

```
dt = DecisionTreeClassifier(random_state=0)

dt.fit(train_X,train_y)          # Fit the training data set to the algorithm passed.

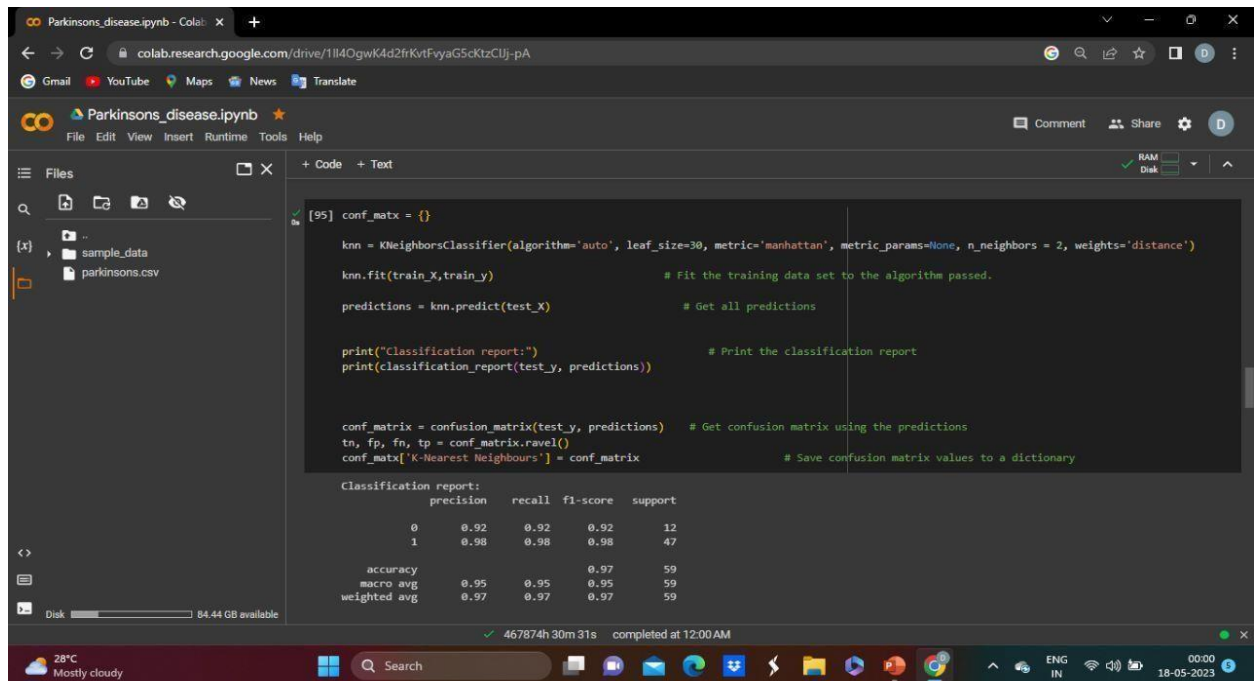
predictions = dt.predict(test_X)    # Get all predictions

print("Classification report:")      # Print the classification report
print(classification_report(test_y, predictions))

conf_matrix = confusion_matrix(test_y, predictions) # Get confusion matrix using the
predictions tn, fp, fn, tp = conf_matrix.ravel() conf_matx['Decision Trees'] = conf_matrix
```

## 5. EVALUATING MODEL ON TEST DATA

### KNN MODEL :



```
[95] conf_matx = {}

knn = KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='manhattan', metric_params=None, n_neighbors = 2, weights='distance')
knn.fit(train_X, train_y) # Fit the training data set to the algorithm passed.
predictions = knn.predict(test_X) # Get all predictions

print("Classification report:") # Print the classification report
print(classification_report(test_y, predictions))

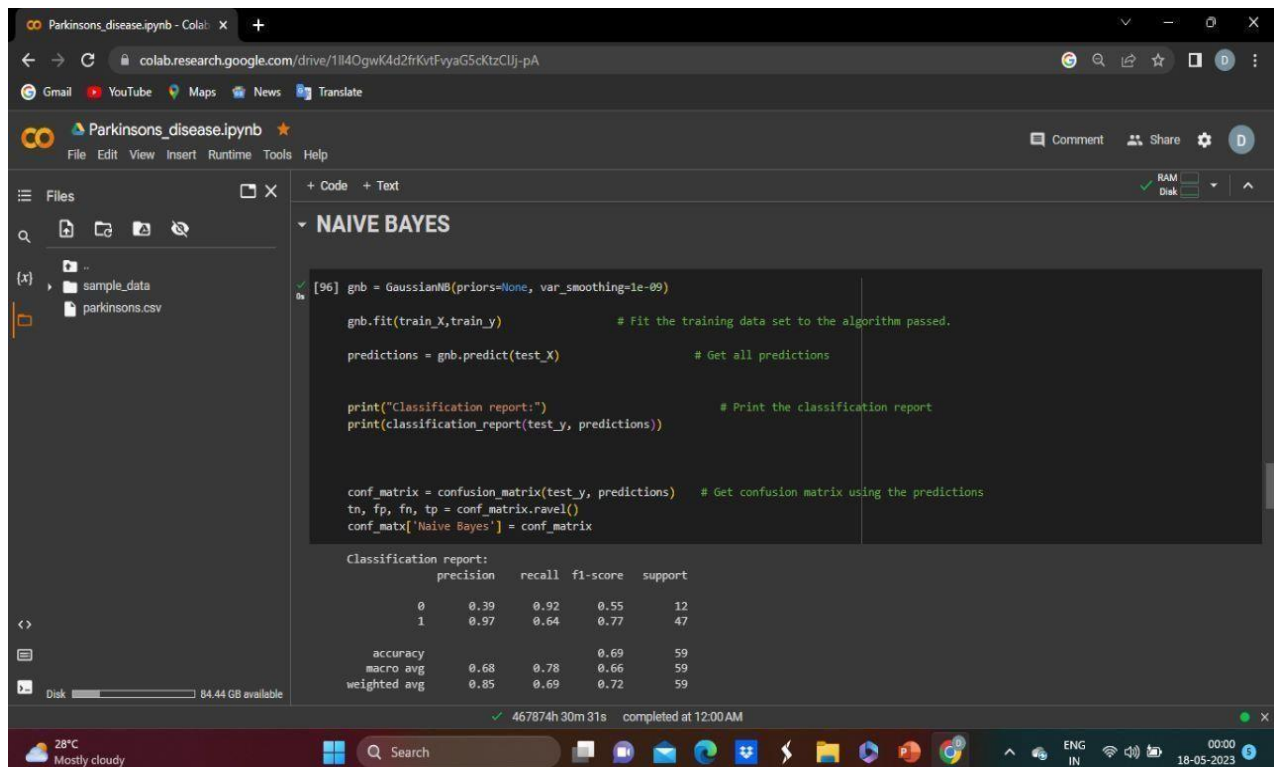
conf_matrix = confusion_matrix(test_y, predictions) # Get confusion matrix using the predictions
tn, fp, fn, tp = conf_matrix.ravel()
conf_matx['K-Nearest Neighbours'] = conf_matrix

Classification report:
      precision    recall  f1-score   support

     0       0.92      0.92      0.92        12
     1       0.98      0.98      0.98        47

   accuracy      0.95      0.95      0.97        59
  macro avg      0.95      0.95      0.95        59
 weighted avg      0.97      0.97      0.97        59
```

### NAIVE BAYES MODEL :



```
[96] gnb = GaussianNB(priors=None, var_smoothing=1e-09)

gnb.fit(train_X, train_y) # Fit the training data set to the algorithm passed.
predictions = gnb.predict(test_X) # Get all predictions

print("Classification report:") # Print the classification report
print(classification_report(test_y, predictions))

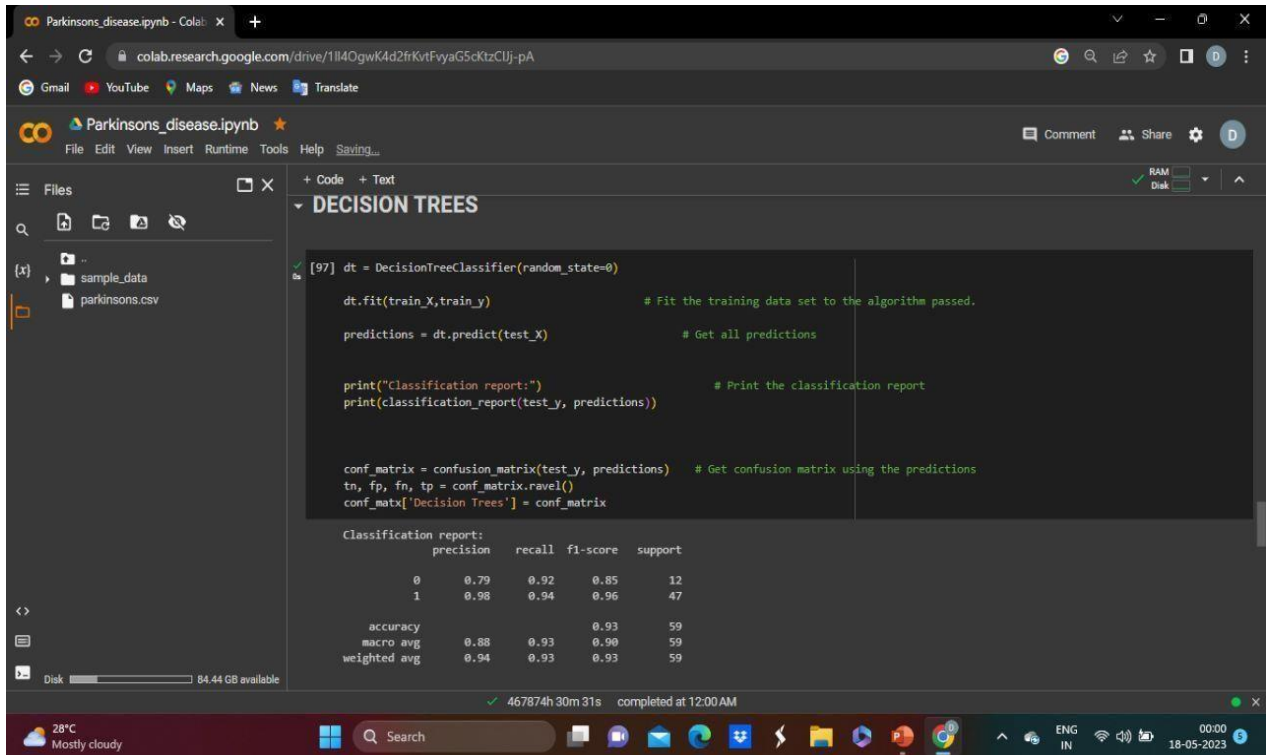
conf_matrix = confusion_matrix(test_y, predictions) # Get confusion matrix using the predictions
tn, fp, fn, tp = conf_matrix.ravel()
conf_matx['Naive Bayes'] = conf_matrix

Classification report:
      precision    recall  f1-score   support

     0       0.39      0.92      0.55        12
     1       0.97      0.64      0.77        47

   accuracy      0.68      0.78      0.69        59
  macro avg      0.68      0.78      0.66        59
 weighted avg      0.85      0.69      0.72        59
```

## DECISION TREE MODEL :



```
[97] dt = DecisionTreeClassifier(random_state=0)

dt.fit(train_X, train_y)          # Fit the training data set to the algorithm passed.

predictions = dt.predict(test_X)  # Get all predictions

print("Classification report:")    # Print the classification report
print(classification_report(test_y, predictions))

conf_matrix = confusion_matrix(test_y, predictions) # Get confusion matrix using the predictions
tn, fp, fn, tp = conf_matrix.ravel()
conf_mat['Decision Trees'] = conf_matrix
```

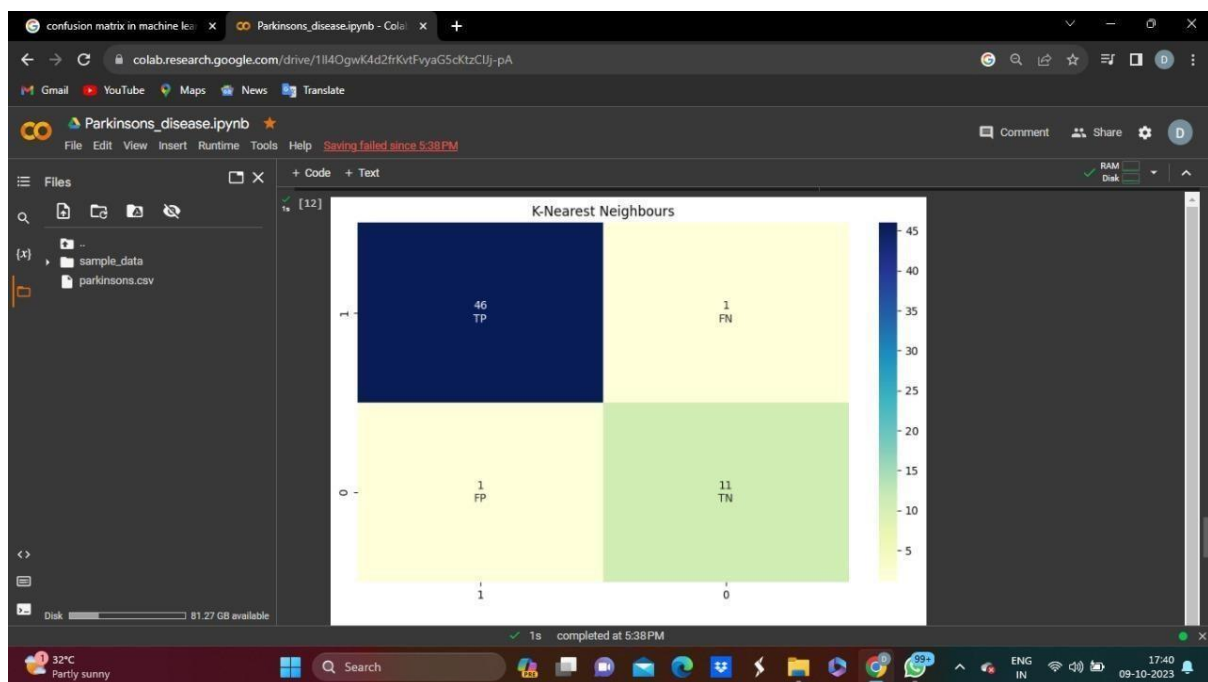
Classification report:				
	precision	recall	f1-score	support
0	0.79	0.92	0.85	12
1	0.98	0.94	0.96	47
accuracy			0.93	59
macro avg	0.88	0.93	0.90	59
weighted avg	0.94	0.93	0.93	59

### 5.1 CONFUSION MATRIX

In machine learning, confusion matrix is a table that is used to evaluate the performance of a classification model. It shows the number of correct and incorrect predictions made by the model compared to the actual outcomes. Confusion matrix provides a more detailed analysis of the performance of a classification model compared to other evaluation metrics such as accuracy, precision, recall, and F1 score. While accuracy only measures the overall performance of the model, confusion matrix breaks down the performance by showing the number of true positives, true negatives, false positives, and false negatives. It is a table that summarizes the performance of a model by comparing its predicted output with the actual output.

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) <b>Type II Error</b>	<b>Sensitivity</b> $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) <b>Type I Error</b>	True Negative (TN)	<b>Specificity</b> $\frac{TN}{(TN + FP)}$
		<b>Precision</b> $\frac{TP}{(TP + FP)}$	<b>Negative Predictive Value</b> $\frac{TN}{(TN + FN)}$	<b>Accuracy</b> $\frac{TP + TN}{(TP + TN + FP + FN)}$

**Figure-11: Confusion Matrix Example**



**Figure-12: KNN Confusion Matrix**

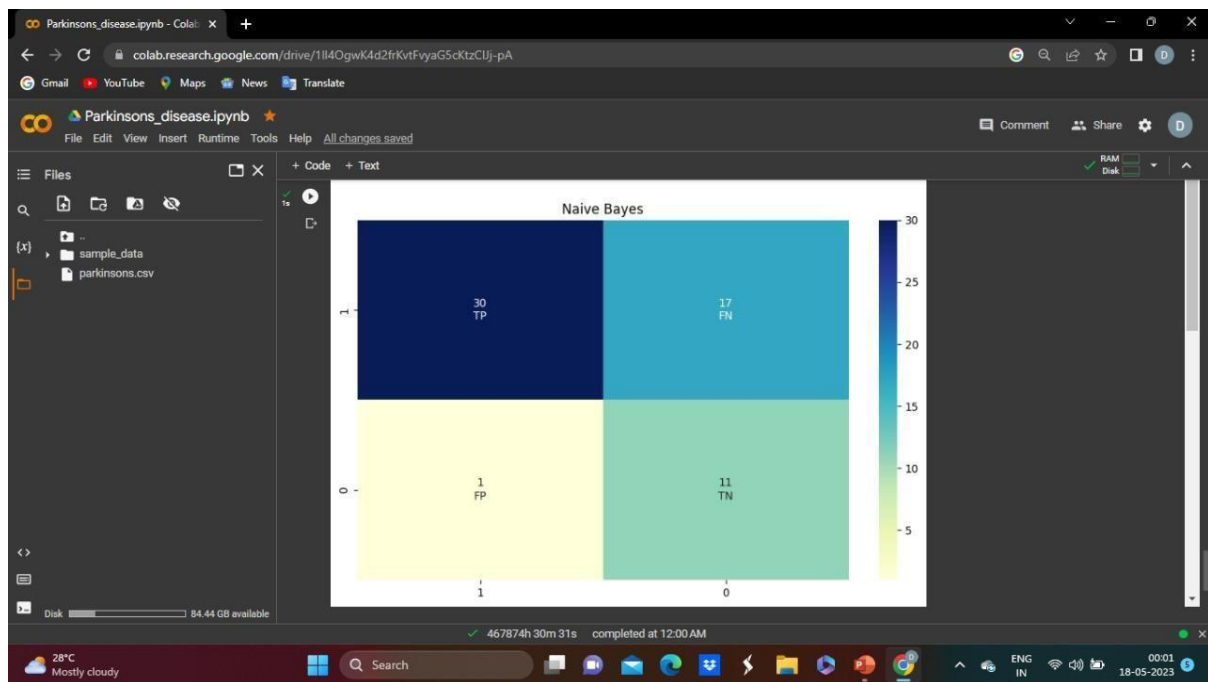


Figure-13: Naive Bayes Confusion Matrix

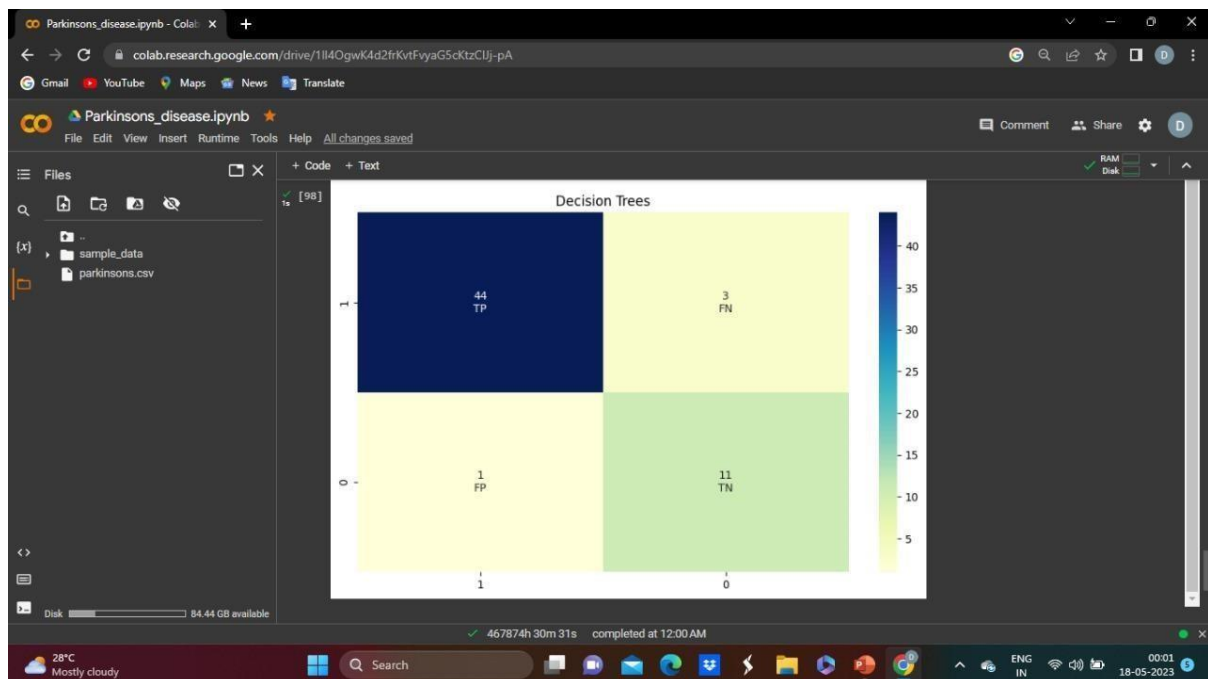


Figure-14: Decision Tree Confusion Matrix

## ACCURACY

Accuracy is one metric for evaluating classification models. Accuracy is the fraction of predictions for our model.  $\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$ . Accuracy can be easily described using the Confusion matrix terms such as True Positive, True Negative, False Positive, and False Negative. Accuracy is the fraction of predictions for our model.

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

## PRECISION

Precision in machine learning is a measure of how accurately a model identifies positive instances among the ones it predicts as positive. It calculates the ratio of true positive predictions to all positive predictions, helping evaluate the model's ability to avoid false positives. High precision means the model has fewer false positives, making it suitable for tasks where such errors are costly or undesirable like medical diagnoses

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

## RECALL

The percentage of data samples that a machine learning model correctly identifies as belonging to a class of interest—the “positive class”—out of the total samples for that class. recall measures the completeness of positive predictions.

$$\text{Recall} = \frac{TP}{TP + FN}$$

## F1-SCORE

It is a machine learning evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model. The accuracy metric computes how many times a model made a correct prediction across the entire dataset.

$$\text{F1 Score} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

## COMPARISON OF ALGORITHMS

Algorithm	Accuracy	Precision	Recall	Support
KNN	0.93	0.88	0.93	0.90
Naive Bayes	0.69	0.68	0.78	0.66
Decision Tree	0.97	0.95	0.98	0.95

Table-1: Performance Comparison

## 5.2 ACCURACIES

```
[ ] print("Accuracy Scores")
    print("KNN= ",knn_accuracy)
    print("Naive bayes= ",nb_accuracy)
    print("Decision tree = ",dt_accuracy)
```

```
Accuracy Scores
KNN= 0.9661016949152542
Naive bayes= 0.6949152542372882
Decision tree = 0.9322033898305084
```

Figure-15: Accuracies

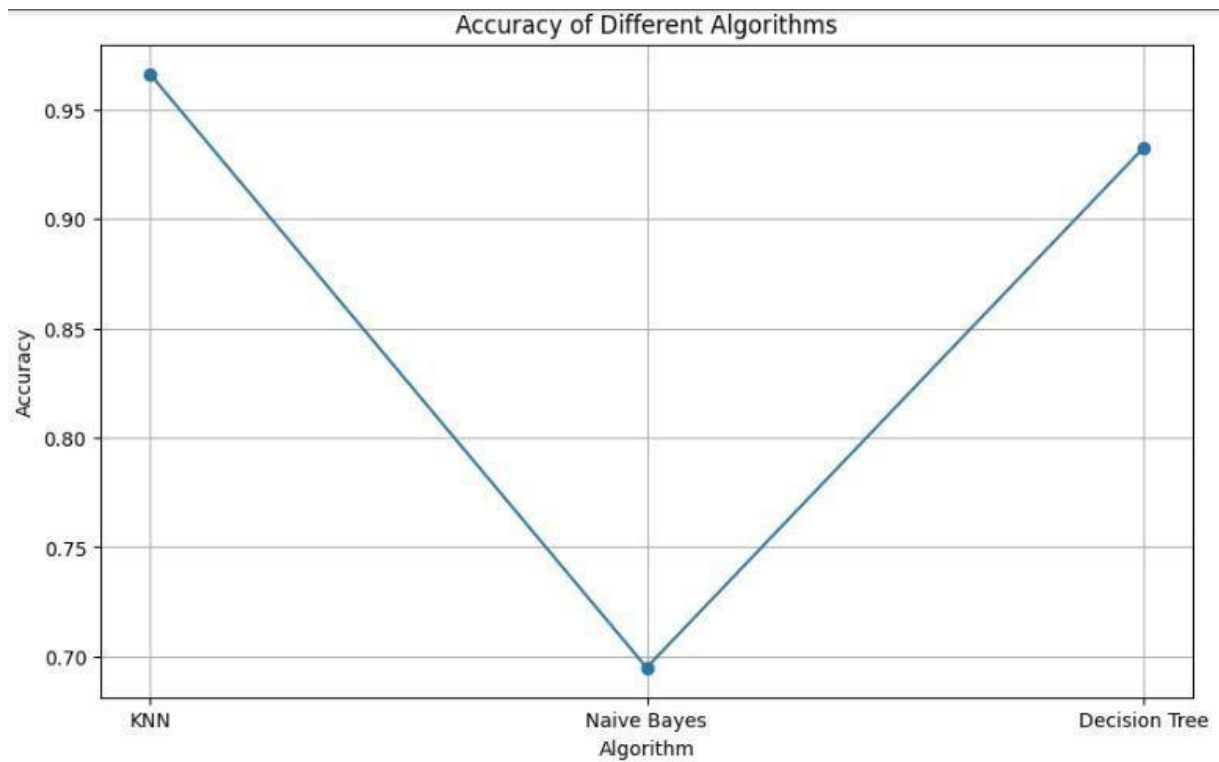


## 5.3 LINE GRAPH

```
[ ] #line graphs

accuracy_scores = {
    'KNN': knn_accuracy,
    'Naive Bayes': nb_accuracy,
    'Decision Tree': dt_accuracy
}

plt.figure(figsize=(10, 6))
plt.plot(list(accuracy_scores.keys()), list(accuracy_scores.values()), marker='o')
plt.xlabel('Algorithm')
plt.ylabel('Accuracy')
plt.title('Accuracy of Different Algorithms')
plt.grid(True)
plt.show()
```



**Figure-16: Line Graph**

## 5.4 BAR GRAPH

```
[ ] #bar_graphs

plt.figure(figsize= (10, 6))
plt.bar(list(accuracy_scores.keys()), list(accuracy_scores.values()))
plt.xlabel('Algorithm')
plt.ylabel('Accuracy')
plt.title('Comparison of Accuracy Across Algorithms')
plt.grid(True)
```

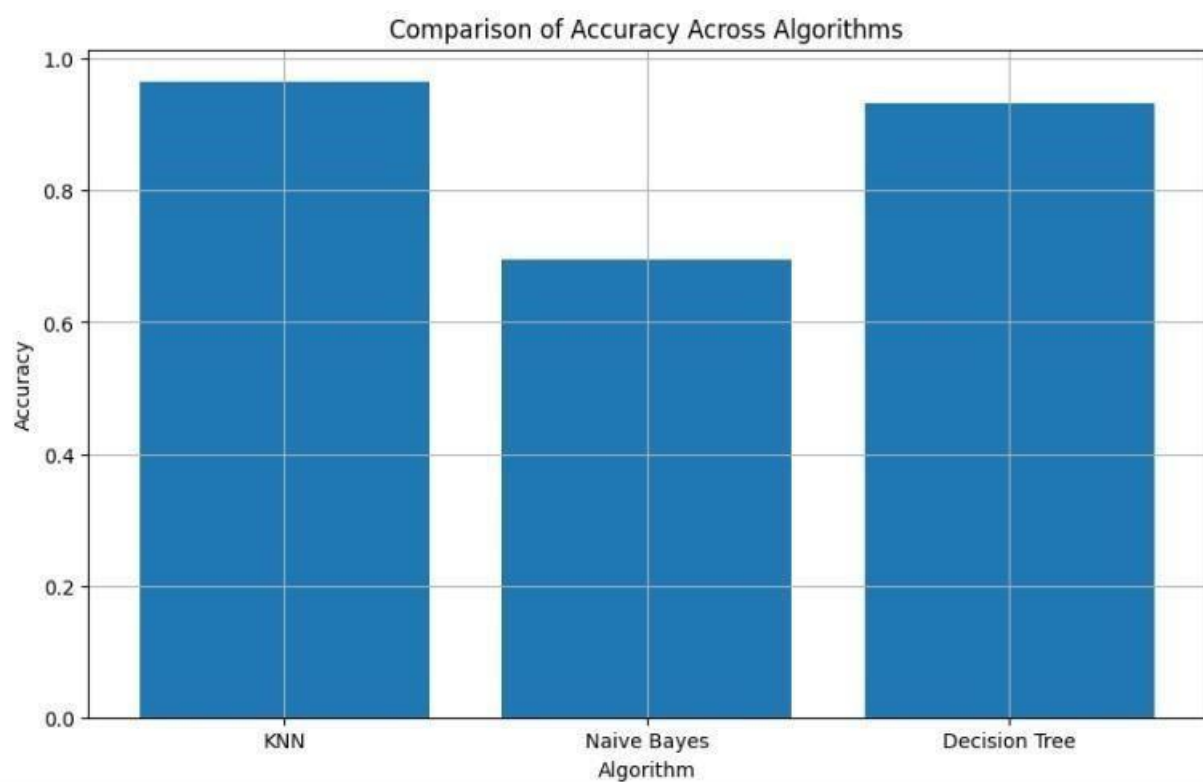


Figure-17: Bar Graph

## 5.5 TOOLS

The various tools used for the project are listed below along with their descriptions.



**Figure-18: Logos of various tools used**

### **EXCEL(CSV)**

CSV (Comma Separated Values) files can be opened in Excel by heading to the "File" tab, selecting "Open," and choosing the CSV file. Excel automatically recognizes the comma as the delimiter and imports the data into separate columns. If the CSV contains line breaks within a cell, Excel allows you to handle them by using the "Text to Columns" feature, ensuring proper data organization and display.

### **GOOGLE COLAB**

Google Colab, short for Google Colaboratory, is a free cloud-based platform that provides access to Jupyter notebooks for coding, data analysis, and machine learning. It offers GPU and TPU support, making it ideal for resource-intensive tasks. Users can collaborate in real-time and store their projects on Google Drive, simplifying data science and research workflows.

## 5.6 TECHNOLOGY USED



**Figure-19: Technology used**

**Python:** Python is a high-level, readable, versatile, and open-source programming language with a rich standard library and a strong community.

(Various libraries used are NumPy, Pandas, Sklearn, Matplotlib, Seaborn)

**Pandas:** Pandas is a python library used for data analysis. It is used to create a data frame that is a 2D data structure similar to excel sheet.

**NumPy:** NumPy is a python library used to handle multidimensional arrays.

**Matplotlib:** Matplotlib is a 2D plotting library for creating plots and graphs.

**Seaborn:** Seaborn is a Python data visualization library based on matplotlib. It provides a highlevel interface for drawing attractive and informative statistical graphics.

**Sklearn:** Scikit-learn is the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. We have used Random Forest Classifier, Decision Tree Classifier and K Neighbor Classifier.

## CONCLUSION

In conclusion, the development of a Parkinson's Disease prediction system using machine learning techniques presents a significant advancement in the field of medical diagnostics. Through the integration of data analysis, feature extraction, and the application of various classification algorithms, this project has successfully addressed the challenge of early and accurate Parkinson's Disease detection.

By leveraging a statistical approach that analyzes the correlation between common symptoms such as gait and tremors, we have achieved a notable milestone in enhancing diagnostic accuracy. The results and outcomes obtained from our predictive models underscore the potential of machine learning to assist healthcare professionals in identifying Parkinson's Disease at an early stage, thus enabling timely intervention and treatment.

The project's success in showcasing the early detection of Parkinson's Disease highlights its potential to significantly impact patient outcomes and quality of life. The ability to provide clinicians with a reliable tool for diagnosis, coupled with the prospect of extending patient lifespans through timely medical intervention, underscores the profound contribution of this project to the healthcare landscape.

As we move forward, the implications of this project extend beyond the realm of Parkinson's Disease diagnosis. The methodology, techniques, and insights gained can serve as a foundation for further research and development in the broader context of neurodegenerative diseases and medical applications of machine learning. The strides made in this project underscore the transformative potential of technology in healthcare, where innovative approaches hold the key to revolutionizing patient care and well-being.

In essence, this project not only addresses a critical medical challenge but also exemplifies the collaborative synergy between medical expertise and cutting-edge technology. The successful amalgamation of data analysis, machine learning, and medical knowledge paves the way for a brighter future in the early diagnosis and management of Parkinson's Disease, fostering optimism for further advancements and breakthroughs in the realm of medical research and innovation.

## FUTURE ENHANCEMENTS

Looking ahead, several avenues for future enhancements and expansion of the Parkinson's Disease prediction project present themselves, promising to further elevate its impact and capabilities:

**Multi-Symptom Analysis:** Extend the system to incorporate a wider array of symptoms and data sources beyond gait and tremors. This holistic approach can provide a more comprehensive view of Parkinson's Disease progression and enhance diagnostic accuracy.

**Longitudinal Tracking:** Implement the ability to track patient data over time to monitor disease progression and treatment effectiveness. Longitudinal analysis can offer insights into the evolution of symptoms and aid in personalized treatment planning.

**Deep Learning Architectures:** Explore the integration of deep learning architectures, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), to capture intricate patterns within patient data and potentially improve prediction accuracy.

**Real-Time Monitoring:** Develop a wearable device or smartphone application that continuously collects data, enabling real-time monitoring of symptoms. This could facilitate early intervention and enable patients to actively manage their condition.

**Integration of Biomarkers:** Incorporate biological markers, such as genetic information or biochemical markers, to enhance prediction models. This could provide insights into disease susceptibility and aid in early diagnosis.

**Clinician Decision Support:** Create a user-friendly interface that assists healthcare professionals in interpreting prediction results. This could include visualizations, trend analysis, and personalized recommendations for patient care.

**Collaborative Research:** Collaborate with medical institutions and researchers to validate the prediction models on larger and more diverse patient populations. Robust validation can strengthen the reliability and applicability of the system.

**Ethical Considerations:** Address ethical and privacy concerns by implementing advanced data anonymization techniques and ensuring compliance with evolving data protection regulations.

**Public Health Initiatives:** Partner with public health organizations to deploy the system on a larger scale, enabling early detection on a community level and contributing to public health initiatives.

**Cross-Disease Applications:** Explore the adaptability of the developed framework to other neurodegenerative diseases or medical conditions with similar symptomatology, broadening the potential impact.

## REFERENCES

- [1] 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.
- [2] Anila M Department of CS1, Dr G Pradeepini Department of CSE, “Diagnosis of Parkinson’s Disease using Artificial Neural Network”, JCR , 7(19): 7260-7269, 2020.
- [3] T. J. Wroge, Y. Özkanca, C. Demiroglu, D. Si, D. C. Atkins and R. H. Ghomi, "Parkinson’s Disease Diagnosis Using Machine Learning and Voice," IEEE Signal Processing in Medicine and Biology Symposium (SPMB), pp.1-7, doi: 10.1109/SPMB.2018.8615607, 2018.
- [4] Siva Sankara Reddy Donthi Reddy and Udaya Kumar Ramanadham “Prediction Parkinson’s Disease at Early Stage using Big Data Analytics”ISSN: 2249 – 8958, Volume9 Issue-4, April 2020.
- [5] Heisters. D, “Parkinson’s: symptoms, treatments and research”. British Journal Nursing, 20(9), 548–554. doi:10.12968/bjon.2011.20.9.548, 2011.
- [6] T. Swapna, Y. Sravani Devi, “Performance Analysis of Classification algorithms on Parkinson’s Dataset with Voice Attributes”. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 2 pp. 452-458, 2019.
- [7] Sriram, T. V., et al. “Intelligent Parkinson Disease Prediction Using Machine Learning Algorithms” International Journal of Engineering and Innovative Technology, vol-3, Issue 3, September 2013.
- [8] Dragana Miljkovic et al, “Machine Learning and Data Mining Methods for Managing Parkinson’s Disease” LNAI 9605, pp 209- 220, 2016.
- [9] Dr. R.Geetha Ramani, G.Sivagami, Shomon Gracia jacob “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.
- [10] A. Ozcift, “SVM feature selection based rotation forest ensemble classifiers to improve computer-aided diagnosis of Parkinson disease” Journal of medical systems, vol-36, no. 4, pp. 2141-2147, 2012.

## **APPENDIX A**

- PD: Parkinson's Disease
- ML: Machine Learning
- KNN: k-Nearest Neighbors
- Naive Bayes: Named after Thomas Bayes, a probabilistic classification technique
- Decision Trees: Tree-based model for classification and regression tasks
- Symptoms: Indications or manifestations of a medical condition
- Gait: Manner of walking or movement pattern
- Tremors: Involuntary shaking or oscillatory movement
- Accuracy: Measure of correctness in predictions
- Data Preprocessing: Data cleaning, transformation, and normalization

## **APPENDIX B – Software Installation Procedure**

- Platform- Google colab
- Python Libraries
- Windows 10 or current version of windows
- mGoogle colab platform

## **APPENDIX C– Software Usage Process**

- Run the Server in colab
- Execute line by line each cell