PARKINSON DISEASE PREDICTION USING MACHINE LEARNING

Project Report

Submitted by

SANTHOSH.S

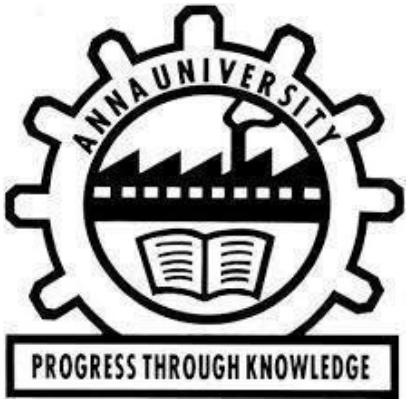
(Register No: 510619104059)

In partial fulfillment for the award of the degree of

BACHELOR IN ENGINEERING

in

COMPUTER SCIENCE



ANNA UNIVERSITY

CHENNAI – 600 025

SEPTEMBER, 2022

BONAFIDE CERTIFICATE

Certified that this project report titled "**PARKINSON DISEASE PREDICTION USING MACHINE LEARNING**" is the bonafide work of **SANTHOSH.S** (Reg. No.: **510619104059)** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

(Signature)	**(Signature)**
Dr.G.SUKUMAR, M.Tech., Ph.D.	**Dr.K.LOKWSHWARAN, M.Tech.,Ph.D.**
Internal Guide	Head of the Department
Department of CSE	Department of CSE
C.Abdul Hakeem College of Engg. & Tech.	C.Abdul Hakeem College of Engg.& Tech
Hakeem Nagar,	Hakeem Nagar,
Melvisharam – 632509	Melvisharam - 632509

Submitted to the project Viva-Voce Examination held on_____

Place: **Melvisharam**

Date:_____

INTERNAL EXAMINER EXTERNAL EXAMINER

C. ABDUL HAKEEM COLLEGE OF ENGG & TECH
(ISO Certified, NBA & NAAC Accredited Institution)
Hakeem Nagar, Melvisharam – 632509
Ranipet District
Phone : 04172 267387 Email : info.cahcet@gmail.com

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **SANTHOSH.S** bearing the Register Number **510619104059** has successfully finished the project work entitled "**PARKINSON DISEASE PREDICTION USING MACHINE LEARNING**" under our supervision. The project period is from 20/04/2022 to 24/08/2022.

Internal Guide HOD

Date :29.08.2022

Place : Melvisharam

ABSTRACT

Using pattern recognition techniques and multiple machine learning approaches, Parkinson's disease is classified and the risk is predicted to the extent using the given speech signal and speech data from the patients. The dataset is collected from the UCI repository.
This model is aimed to provide greater accuracy than other complex models. In this project, Light Gradient Boosting Model is used to classify Parkinson disease. The main objective of this project is to train, test the data and predict the data to find the similarities and differences among the data and also classify based on the LGBM model as it shows higher accuracy compared to the other models. And the other objective is to check which classification algorithm gives high accuracy rate and less error rate for the given data. The Pycaret package is being used for the training and classification purpose the csv data has to be uploaded to the system and the backend takes care of the prediction process and gets the csv files with the prediction results to download for the user which can be used for later analysis. The Pyplot library is used for the dynamic graphs that are displayed on the final frontend of the system which is created using the datagram of the final csv file with other pyplot parameters required for the plot generation.

ACKNOWLEDGEMENT

I first offer my thanks to the almighty who has given me the strength and good health during the course of this project.

I express my sincere gratitude to our honorable Chairman **Haji. Janab S. Ziauddin Ahmed, B. A.,** C. Abdul Hakeem College of Engineering and Technology.

I express my profound gratefulness to our Correspondent **Haji. Janab V. M. Abdul Latheef, B. E.,** and **Dr. R. DHANASEKARAN, Ph. D.,**
Principal, C. Abdul Hakeem College of Engineering and Technology.

I would like to express my deeply felt gratefulness to our beloved Head of the department and my guide **Dr. M. Sadique Basha, M.C.A., M.Tech. Ph.D.,** for his guidance and encouragement.

I also thank all my faculty members who were instrumental in the completion of this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

DNA	DeoxyriboNucleicAcid
PD	Parkinson's Disease
IT	Information
KNN	K-Nearest Neighbor
CSV	Comma Separated Value
NB	Naïve Bayes
HTML	Hyper Text Markup Language
LGBM	Light Gradient Boosting Model

CHAPTER 1

INTRODUCTION

1.1 Introduction

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

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

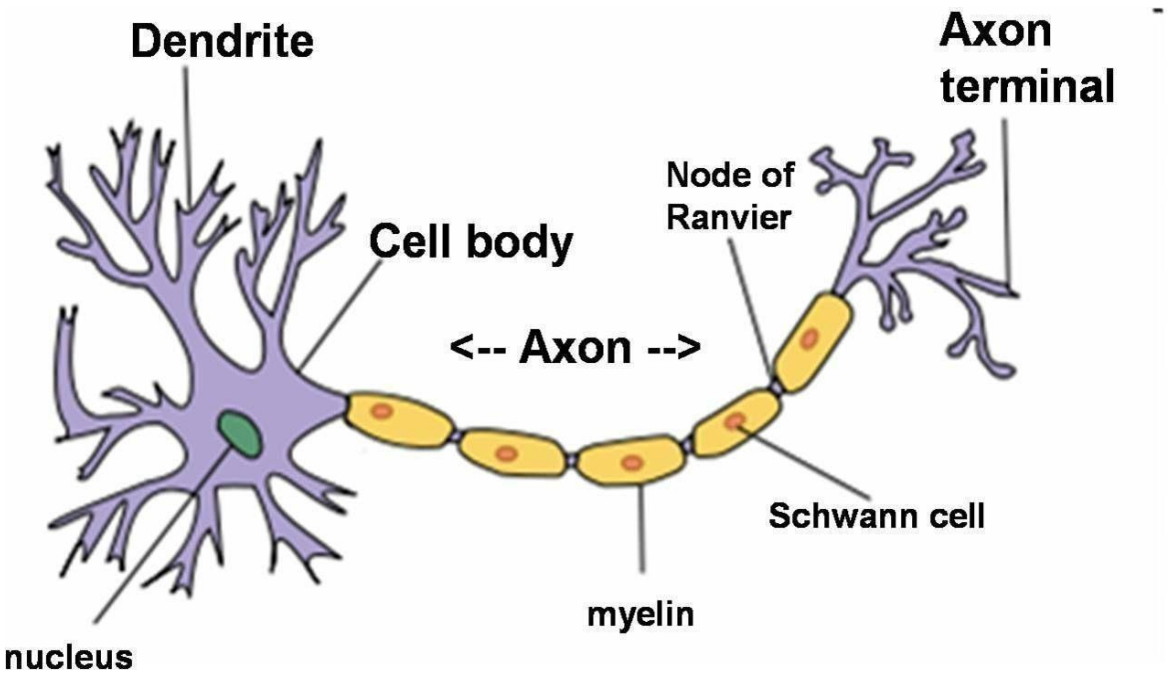


Fig. No - 1.1 Structure of Neuron

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

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

- Air Quality: Pollution results in respiratory diseases.
- Water quality: Biotic and Abiotic contaminants present in water lead to water pollution.
- Unhealthy lifestyle: It leads to obesity and a sedentary lifestyle.
- Psychological stress: It increases the level of stress hormone that depletes the functions of neurons.

- **Brain injuries or Biochemical Factors:** The brain is the control center of our complete body. Due to certain trauma, people have brain injuries which leads some biochemical enzymes to come into the picture which provides neurons stability and provides support to some chromosomes and genes in maintenance.

- **Aging Factor:** Aging is one of the reasons for the development of Parkinson's disease. According to the author in India, 11,747,102 people out of 1, 065, 070, 6072 are affected by Parkinson's disease.

- **Genetic factors:** Genetic factor is considered as the main molecular physiological cause which leads to neurodegenerative disorders. The size, depth, and effect of actions of different genes define the status or level of neurodegenerative disease which increases itself gradually over time. Mainly the genetic factors which lead to Neurodegenerative disorders are categorized into pharmacodynamics and pharmacokinetics.
- **Speech Articulation Factors:** Due to the condition associated with Parkinson's disease (rigidity and bradykinesia), some speech-language pathology such as voice, articulation and swallowing alterations are found. There are various ways in which Parkinson's disease (PD) might affect the individual.
- The voice get breathy and softer.
- Speech may be smeared.

- The person finds difficulty in finding the right words due to which speech becomes slower.

1.2 Parkinson's Disease Symptoms

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

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

1.3 Introduction to Machine Learning

Machine Learning may be a sub-area of AI, whereby the term refers to the power of IT systems to independently find solutions to problems by recognizing patterns in databases. In other words: Machine Learning enables IT systems to acknowledge patterns in the idea of existing algorithms and data sets and to develop adequate solution concepts. Therefore, in Machine Learning, artificial knowledge is generated on the idea of experience. In order to enable the software to independently generate solutions, the prior action of people is important. For example, the required algorithms and data must be fed into the systems in advance and the

respective analysis rules for the recognition of patterns in the data stock must be defined. Once these two steps have been completed, the system can perform the following tasks by Machine Learning:

- Finding, extracting and summarizing relevant data
- Making predictions based on the analysis data
- Calculating probabilities for specific results

Basically, algorithms play a crucial role in Machine Learning: On the one hand, they're liable for recognizing patterns and on the opposite hand, they will generate solutions.
Algorithms can be divided into different categories:

1.3.1 Supervised Learning

In the course of monitored learning, example models are defined beforehand. So as to ensure an adequate allocation of the knowledge to the respective model groups of the algorithms, these then need to be specified. In other words, the system learns on the idea of given input and output pairs. Within the course of monitored learning, a programmer, who acts as a sort of teacher, provides the acceptable values for specific input. The aim is to coach the system within the context of successive calculations with different inputs and outputs to determine connections.

Supervised learning is where you've got input variables (X) and an output variable (Y) and you employ an algorithm to find out the mapping function from the input to the output. Y = f(X) The goal is to approximate the mapping function so well that once you have a new input file (X) that you simply can predict the output variables (Y) for that data. It's called supervised learning because the method of an algorithm learning from the training dataset is often thought of as an educator supervising the training process. We all know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected. Learning stops when the algorithm achieves a suitable level of performance.

Techniques of Supervised Machine Learning algorithms include linear and logistic regression, multi-class classification, Decision Tree, and Support Vector Machine.

Supervised Learning problems are a kind of machine learning technique often further grouped into Regression and Classification problems. The difference between these two is that the dependent attribute is numerical for regression and categorical for classification:

- **Regression:**

  Linear regression could also be a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and thus the only output variable (y). More specifically, that y is usually calculated from a linear combination of the input variables (x).

  When there's one input variable (x), the tactic is mentioned as simple linear regression. When there are multiple input variables, literature from statistics often refers to the tactic as multiple linear regression.

- **Classification:**

  Classification could also be a process of categorizing a given set of data into classes, It is often performed on both structured or unstructured data. The tactic starts with predicting the category of given data points. The classes are often mentioned as target, label, or categories.

  In short, classification either predicts categorical class labels or classification data supports the training set and thus the values(class labels) in classifying attributes and uses it in classifying new data.

  There are a variety of classification models. Classification models include Logistic Regression, Decision Tree, Random Forest, Gradient Boosted Tree, One-vs.-One, and Naïve Bayes.

1.3.2 Unsupervised Learning

In unsupervised learning, AI learns without predefined target values anhd without rewards. It's mainly used for learning segmentation (clustering). The machine tries to structure and type the info entered consistent with certain characteristics. For instance, a machine could (very simply) learn that coins of various colors are often sorted consistent with the characteristic "color" so as to structure them. Unsupervised Machine Learning algorithms are used when the knowledge used to train is neither classified nor labeled. The system doesn't determine the right

output but it explores the data and should draw inferences from datasets to elucidate hidden structures from unlabeled data. Unsupervised Learning is that the training of Machines using information that's neither classified nor labeled and allowing the algorithm to act thereon information without guidance.

Unsupervised Learning is accessed into two categories of algorithms:

- Clustering: A clustering problem is where you would like to get the inherent grouping in the data such as grouping customers by purchasing behavior.
- Association: An Association rule learning problem is where you would wish to get rules that describe large portions of your data such as folks that buy X also tend to shop for Y.

1.3.3 Applications of Machine Learning:

Virtual Personal Assistants:

Siri, Alexa, Google Now are a number of the favored samples of virtual personal assistants. As the name suggests, they assist in finding information, when asked over voice. Machine learning is a crucial part of these personal assistants as they collect and refine the knowledge on the idea of your previous involvement with them.

Later, this set of knowledge is employed to render results that are tailored to your preferences.

Virtual Assistants are integrated to a spread of platforms. For example:

- Smart Speakers : Amazon Echo and Google Home
- Smartphone : Samsung Bixby on Samsung S8
- Mobile Apps : Google Allo

Videos Surveillance

- Imagine one person monitoring multiple video cameras! Certainly, a difficult job to try to do and boring also. This is why the thought of coaching computers to try to do this job is sensible.
- The video closed-circuit television nowadays is powered by AI that creates it possible to detect crimes before they happen. They track unusual behavior of individuals like

standing motionless for an extended time, stumbling, or napping on benches, etc. The system can thus give an awareness of human attendants, which may ultimately help to avoid mishaps. And when such activities are reported and counted to be true, they assist to enhance the surveillance services. This happens with machine learning doing its job at the backend.

Social Media Services

From personalizing your news feed to raised ads targeting, social media platforms are utilizing machine learning for his or her own and user benefits.

- People You May Know
- Face Recognition

Search Engine Result Refining

Google and other search engines use machine learning to enhance the search results for you. Every time you execute an inquiry, the algorithms at the backend keep a watch on how you answer the results. If you open the highest results and stay on the online page for long, the program assumes that the results it displayed were in accordance with the query. Similarly, if you reach the second or third page of the search results but don't open any of the results, the program estimates that the results served did not match the requirement. This way, the algorithms performing at the backend improve the search results.

1.4 Motivation of the Work

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

Parkinson's disease is increasing, it is very important to find the solution which can predict it in its early stages.

1.5 Problem Statement

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

1.6 Organization of Thesis

The chapters in this document is described as follows:

Chapter-1 is about the introduction of Parkinson's disease, the different type of symptoms on it and we have given clear insights about our project domain and related concepts.

Chapter-2 which gives an account of the review on literature survey where all different existing methods and models are examined.

Chapter-3 which deals with the problem statement and specifies a proposed system with a system architecture along with machine learning techniques used.

Chapter-4 specifies the experimental analysis of our system along with performance measures and comparisons between different models. It also specifies about implementation along with sample code.

Chapter-5 gives the conclusion to our work and future scope.

CHAPTER 2

LITERATURE SURVEY

2.1 Literature Survey

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

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

Arvind Kumar Tiwari Proposed the paper titled “Machine Learning-based Approaches for Prediction of Parkinson’s Disease” [3]. In this paper, minimum redundancy maximum relevance feature selection algorithms were used to select the most important feature among all the features to predict Parkinson diseases. Here, it was observed that the random forest with 20

number of features selected by minimum redundancy maximum relevance feature selection algorithms provide the overall accuracy 90.3%, precision 90.2%, 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, random subspace, support vector machine, multilayer perceptron, and decision tree based methods.

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

There are some limitations to this paper namely:

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

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

based DNN model for the prediction of Motor-UPDRS and Total-UPDRS in Parkinson's Disease progression. The DNN model was evaluated on a real-world PD dataset taken from UCI. Being a DNN model, the performance of the proposed model may improve with the addition of more data points in the datasets.

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

Siva Sankara Reddy Donthi Reddy and Udaya Kumar Ramanadham proposed the paper “Prediction of Parkinson’s Disease at Early Stage using Big Data Analytics” [21]. 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 using different
Big Data Analytical techniques like Hadoop, Hive, R Programming, MapReduce, PIG, Zookeeper, HBase, Cassandra, Mahout etc…

Daiga Heisters proposed the paper titled “Parkinson’s: symptoms, treatments and research” [9]. 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" [23].This paper deals with the application of seven classification algorithms on the acquired data set and then drawing out a comparison of the results to one another and also predicting the outcome whether the person is healthy or Parkinson disease effected from the given data. The results of the selected algorithms namely Naïve Bayes, Random Forest, Neural Networks, Decision Trees, AdaBoost, SVM, KNN, LGBM 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 the Naïve Bayes Algorithm which has the optimal accuracy of 70.82% and lastly Light Gradient Boosting Model has the optimal accuracy of 90% Finally, this algorithm can help in classifying whether a person get affected with Parkinson's disease or not.

M. Abdar and M. Zomorodi-Moghadam proposed a paper "Impact of Patients' Gender on Parkinson's disease using Classification Algorithms" [10]. The output variable is Sex and other factors are input, the author provides an approach for finding relationships between genders. The result obtained is that the SVM algorithm gives better performance than the Bayesian Network with 90.98% accuracy.

Dragana Miljkovic, et al, proposed a paper "Machine Learning and Data Mining Methods for Managing Parkinson's Disease" [7]. 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.

Sriram, T. V., et al. proposed a paper "Intelligent Parkinson Disease Prediction Using Machine Learning Algorithms" [22]. 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%.

Dr. R.GeethaRamani, G.Sivagami, and ShomonaGraciajacob proposed a paper "Feature Relevance Analysis and Classification of Parkinson's Disease TeleMonitoring data Through Data Mining" [6]. In this paper, the author used thirteen classification algorithms to diagnose the disease. The author used the Tele-monitoring dataset which contains 16 biomedical voice features for evaluating the system. The aim of this paper is to predict motor UPDRS and total UPDRS from the voice measures.

A. Ozcift, proposed a paper "SVM feature selection based rotation forest ensemble classifiers to improve computer-aided diagnosis of Parkinson disease" [1]. In this paper, the author summarizes the improvement of 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.

2.2 Existing System

In the existing systems, it tends that they generally used well known techniques Naïve Bayes, K-Nearest Neighbor and decision tree etc. for predicting and classifying the Parkinson disease and some of the algorithms and resulting accuracy has been low in many cases and also the error rate has been high for the most frequently used models and algorithms which makes the classification and prediction a complex one unstable. Also, some algorithms won't work properly on complex systems and datasets and results being error prone in several cases. Few

others also tried to use an ensemble method which outperformed other individual models including more complex ones like neural networks still lacks some accuracy. As this is a voice-based measurement dataset few frequent techniques failed to classify the disease and accuracy was much lower and error rate is also high in this case.

2.3 Proposed system

We are using the Light Gradient Boosting Machine technique for prediction and classifying the disease because its accuracy is high compared to Decision tree, Naïve Bayes, SVM algorithms. Also, the error rate is comparatively lesser than the Decision tree, Naïve Bayes, SVM and other frequently used algorithms. Also, when the data is complex LGBM gives a better result and accuracy.

LIGHT GRADIENT BOOSTING MACHINE ALGORITHM

Light GBM is a gradient boosting framework based on decision trees to increase the efficiency of the model and reduce memory usage. It uses two novel techniques: Gradient-based One Side Sampling and Exclusive Feature Bundling (EFB) which fulfills the limitations of histogram-based algorithm that is primarily used in all GBDT (Gradient Boosting Decision Tree) frameworks. The two techniques of GOSS and EFB described below form the characteristics of Light GBM Algorithm. They comprise together to make the model work efficiently and provide it a cutting edge over other GBDT frameworks. Highdimensional data are usually very sparse which provides us a possibility of designing a nearly lossless approach to reduce the number of features. Specifically, in a sparse feature space, many features are mutually exclusive, i.e., they never take nonzero values simultaneously. The exclusive features can be safely bundled into a single feature (called an Exclusive Feature Bundle). Hence, the speed for training framework is improved without hurting accuracy in

Light GBM

.

2.4 Advantages of proposed system

- Ability to learn and extract complex features
- Accuracy is good
- With its simplicity and fast processing time, the proposed algorithm gives better execution time.

CHAPTER 3

METHODOLOGY

3.1 Proposed System

3.1.1 System Architecture

Machine learning has given computer systems the ability to automatically learn without being explicitly programmed. In this, we have used the LGBM algorithm which produces better accuracy. The architecture diagram describes the high-level overview of major system components and important working relationships. It represents the flow of execution and it involves the following five major steps:

- The architecture diagram is defined with the flow of the process which is used to refine the raw data and used for predicting Parkinson's data.
- The next step is preprocessing the collected raw data into an understandable format.
- Then we have to train the data by splitting the dataset into train data and test data.
- Parkinson's data is evaluated with the application of a machine learning algorithm that is LGBM, and the classification accuracy of this model is found.
- After training the data with these algorithms we have to test on the same algorithms.
- Finally, the result of these three algorithms is compared on the basis of classification accuracy.

3.2 Modules Division

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

3.2.1 Speech Dataset

The main aim of this step is to spot and acquire all data-related problems. during this step, we'd like to spot the various data sources, as data are often collected from various

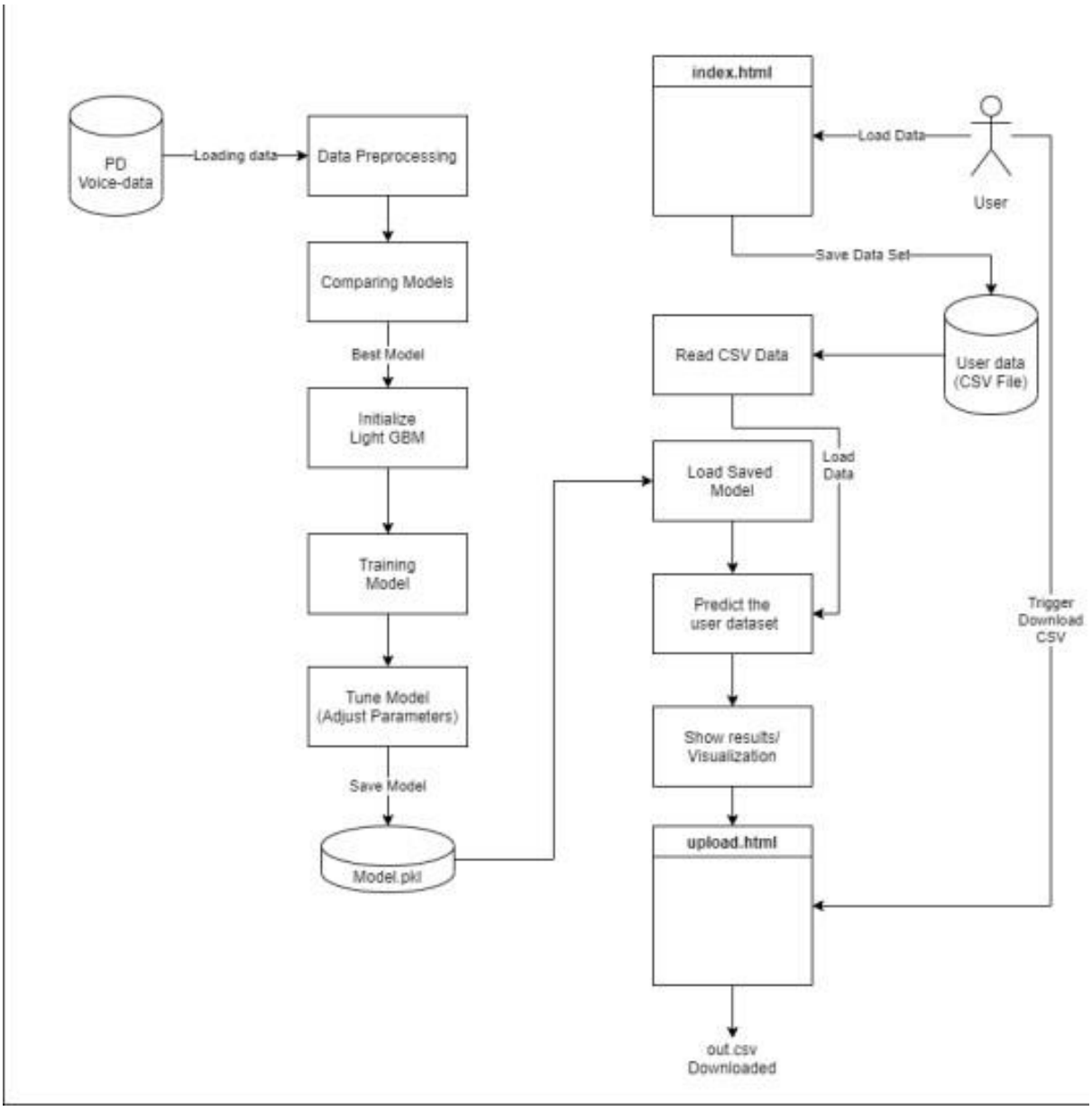
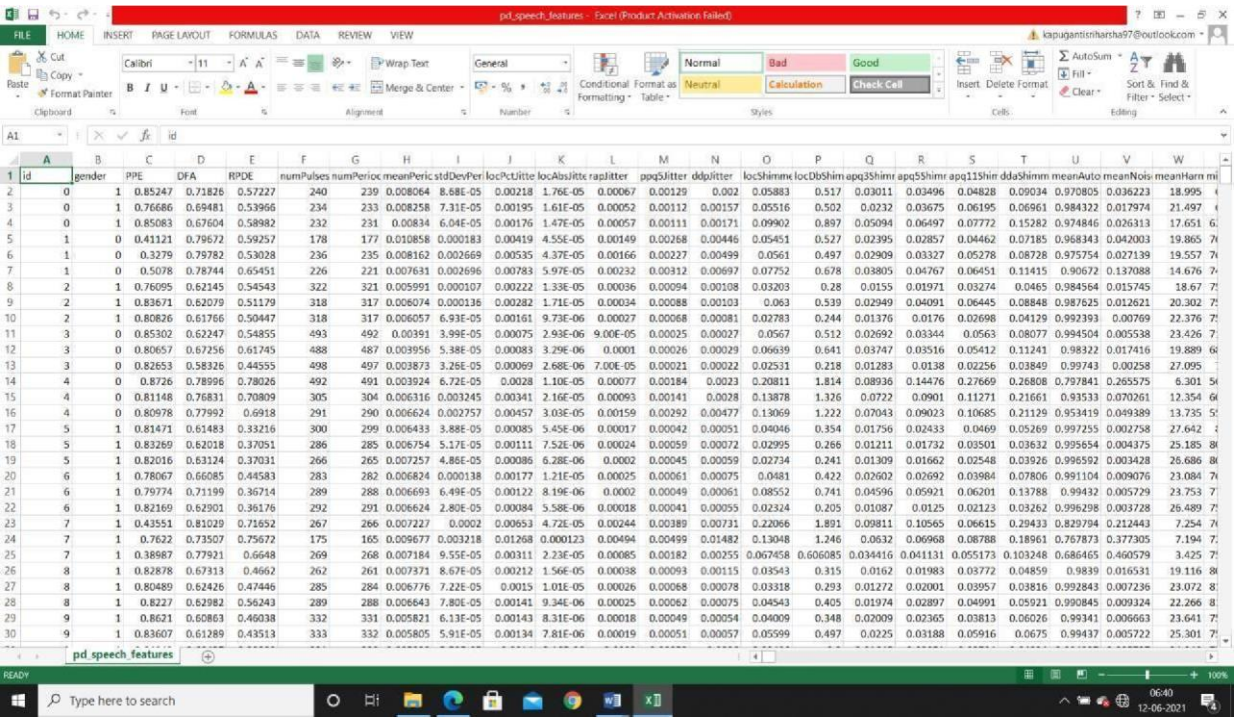


Fig. No - 3.1 – Architecture Diagram

The final system has been visually represented in this architecture diagram, the training data is from the Kaggle website and the user data is being processed and the final output files is out.csv with the predicted data and the score of accuracy for each record of the data.



id	gender	PPE	DFA	RPDE	numPulses	numPeriodPulses	meanPeriodPulses	stdDevPeriodPulses	locPctJitter	locAbsJitter	rapJitter	ppq5Jitter	ddpJitter	locShimmer	locDbShimmer	apq3Shimmer	apq5Shimmer	apq11Shimmer	ddaShimmer	meanAutoCorr	meanNoiseToHarm	meanHarmToNoise
0	1	0.85247	0.71826	0.57227	240	239	0.008064	8.68E-05	0.00218	1.76E-05	0.00067	0.00129	0.002	0.05883	0.517	0.03011	0.03496	0.04828	0.09034	0.970805	0.036223	18.995
0	1	0.76686	0.69481	0.53966	234	233	0.008258	7.31E-05	0.00195	1.61E-05	0.00052	0.00112	0.00157	0.05516	0.502	0.0232	0.03675	0.06195	0.06961	0.984322	0.017974	21.497
0	1	0.85083	0.67604	0.58982	232	231	0.00834	6.04E-05	0.00176	1.47E-05	0.00057	0.00111	0.00171	0.09902	0.897	0.05094	0.06497	0.07772	0.15282	0.974846	0.026313	17.651
1	0	0.41121	0.79672	0.59257	178	177	0.010858	0.000183	0.00419	4.55E-05	0.00149	0.00268	0.00446	0.05451	0.527	0.02395	0.02857	0.04462	0.07185	0.968343	0.042003	19.865
1	0	0.3279	0.79782	0.53028	236	235	0.008162	0.002669	0.00535	4.37E-05	0.00166	0.00227	0.00499	0.0561	0.497	0.02909	0.03327	0.05278	0.08728	0.975754	0.027139	19.557
1	0	0.5078	0.78744	0.65451	226	221	0.007631	0.002696	0.00783	5.97E-05	0.00232	0.00312	0.00697	0.07752	0.678	0.03805	0.04767	0.06451	0.11415	0.90672	0.137088	14.676
2	1	0.76095	0.62145	0.54543	322	321	0.005991	0.000107	0.00222	1.33E-05	0.00036	0.00094	0.00108	0.03203	0.28	0.0155	0.01971	0.03274	0.0465	0.984564	0.015745	18.67
2	1	0.83671	0.62079	0.51179	318	317	0.006074	0.000136	0.00282	1.71E-05	0.00034	0.00088	0.00103	0.063	0.539	0.02949	0.04091	0.06445	0.08848	0.987625	0.012621	20.302
2	1	0.80826	0.61766	0.50447	318	317	0.006057	6.93E-05	0.00161	9.73E-06	0.00027	0.00068	0.00081	0.02783	0.244	0.01376	0.0176	0.02698	0.04129	0.992393	0.00769	22.376
3	0	0.85302	0.62247	0.54855	493	492	0.00391	3.99E-05	0.00075	2.93E-06	9.00E-05	0.00025	0.00027	0.0567	0.512	0.02692	0.03344	0.0563	0.08077	0.994504	0.005538	23.426
3	0	0.80657	0.67256	0.61745	488	487	0.003956	5.38E-05	0.00083	3.29E-06	0.0001	0.00026	0.00029	0.06639	0.641	0.03747	0.03516	0.05412	0.11241	0.98322	0.017416	19.889
3	0	0.82653	0.58326	0.44555	498	497	0.003873	3.26E-05	0.00069	2.68E-06	7.00E-05	0.00021	0.00022	0.02531	0.218	0.01283	0.0138	0.02256	0.03849	0.99743	0.00258	27.095
4	0	0.8726	0.78996	0.78026	492	491	0.003924	6.72E-05	0.0028	1.10E-05	0.00077	0.00184	0.0023	0.20811	1.814	0.08936	0.14476	0.27669	0.26808	0.797841	0.265575	6.301
4	0	0.81148	0.76831	0.70809	305	304	0.006316	0.003245	0.00341	2.16E-05	0.00093	0.00141	0.0028	0.13878	1.326	0.0722	0.0901	0.11271	0.21661	0.93533	0.070261	12.354
4	0	0.80978	0.77992	0.6918	291	290	0.006624	0.002757	0.00457	3.03E-05	0.00159	0.00292	0.00477	0.13069	1.222	0.07043	0.09023	0.10685	0.21129	0.953419	0.049389	13.735
5	1	0.81471	0.61483	0.33216	300	299	0.006433	3.88E-05	0.00085	5.45E-06	0.00017	0.00042	0.00051	0.04046	0.354	0.01756	0.02433	0.0469	0.05269	0.997255	0.002758	27.642
5	1	0.83269	0.62018	0.37051	286	285	0.006754	5.17E-05	0.00111	7.52E-06	0.00024	0.00059	0.00072	0.02995	0.266	0.01211	0.01732	0.03501	0.03632	0.995654	0.004375	25.185
5	1	0.82016	0.63124	0.37031	266	265	0.007257	4.86E-05	0.00086	6.28E-06	0.0002	0.00045	0.00059	0.02734	0.241	0.01309	0.01662	0.02548	0.03926	0.996592	0.003428	26.686
6	1	0.78067	0.66085	0.44583	283	282	0.006824	0.000138	0.00177	1.21E-05	0.00025	0.00061	0.00075	0.0481	0.422	0.02602	0.02692	0.03984	0.07806	0.991104	0.009076	23.084
6	1	0.79774	0.71199	0.36714	289	288	0.006693	6.49E-05	0.00122	8.19E-06	0.0002	0.00049	0.00061	0.08552	0.741	0.04596	0.05921	0.06201	0.13788	0.99432	0.005729	23.753
6	1	0.82169	0.62901	0.36176	292	291	0.006624	2.80E-05	0.00084	5.58E-06	0.00018	0.00041	0.00055	0.02324	0.205	0.01087	0.0125	0.02123	0.03262	0.996298	0.003728	26.489
7	1	0.43551	0.81029	0.71652	267	266	0.007227	0.0002	0.00653	4.72E-05	0.00244	0.00389	0.00731	0.22066	1.891	0.09811	0.10565	0.06615	0.29433	0.829794	0.212443	7.254
7	1	0.7622	0.73507	0.75672	175	165	0.009677	0.003218	0.01268	0.000123	0.00494	0.00499	0.01482	0.13048	1.246	0.0632	0.06968	0.08788	0.18961	0.767873	0.377305	7.194
7	1	0.38987	0.77921	0.6648	269	268	0.007184	9.55E-05	0.00311	2.23E-05	0.00085	0.00182	0.00255	0.067458	0.606085	0.034416	0.041131	0.055173	0.103248	0.686465	0.460579	3.425
8	1	0.82878	0.67313	0.4662	262	261	0.007371	8.67E-05	0.00212	1.56E-05	0.00038	0.00093	0.00115	0.03543	0.315	0.0162	0.01983	0.03772	0.04859	0.9839	0.016531	19.116
8	1	0.80489	0.62426	0.47446	285	284	0.006776	7.22E-05	0.0015	1.01E-05	0.00026	0.00068	0.00078	0.03318	0.293	0.01272	0.02001	0.03957	0.03816	0.992843	0.007236	23.072
8	1	0.8227	0.62982	0.56243	289	288	0.006643	7.80E-05	0.00141	9.34E-06	0.00025	0.00062	0.00075	0.04543	0.405	0.01974	0.02897	0.04991	0.05921	0.990845	0.009324	22.266
9	1	0.8621	0.60863	0.46038	332	331	0.005821	6.13E-05	0.00143	8.31E-06	0.00018	0.00049	0.00054	0.04009	0.348	0.02009	0.02365	0.03813	0.06026	0.99341	0.006663	23.641
9	1	0.83607	0.61289	0.43513	333	332	0.005805	5.91E-05	0.00134	7.81E-06	0.00019	0.00051	0.00057	0.05599	0.497	0.0225	0.03188	0.05916	0.0675	0.99437	0.005722	25.301

Fig. No - 3.2 Sample of acquired speech dataset from kaggle

In the above Fig-3.2, we can see the speech dataset that has been collected from kaggle website. This acquired dataset has around 756 patient's data and each row has 755 different voice features. But in this paper, we chose 10 main features that required us to find the prediction.

The features are listed below:

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

- rapJitter
- locShimmer, etc.

Fig. No - 3.3 Reading the dataset from the CSV file into notebook

The dataset we chose is in the form of CSV (Comma Separated Value) file. After acquiring the data our next step is to read the data from the CSV file into the Google colab also called a Python notebook. Python notebook is used in our project for data preprocessing, features selection, and for model comparison. In the fig-3.3, we have shown how to read data

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

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

```
Mounted at /content/drive
```

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

	id	gender	PPE	DFA	RPDE	numPulses	numPeriodsPulses	meanPeriodPulses	stdDevPeriodPulses	locPctJitter	locAbsJitter	rapJitter	ppq5Jitter	ddpJitter	locShimmer	loc
0	0	1	0.85247	0.71826	0.57227	240	239	0.008064	0.000087	0.00218	0.000018	0.00067	0.00129	0.00200	0.05883	
1	0	1	0.76686	0.69481	0.53966	234	233	0.008258	0.000073	0.00195	0.000016	0.00052	0.00112	0.00157	0.05516	
2	0	1	0.85083	0.67604	0.58982	232	231	0.008340	0.000060	0.00176	0.000015	0.00057	0.00111	0.00171	0.09902	
3	1	0	0.41121	0.79672	0.59257	178	177	0.010858	0.000183	0.00419	0.000046	0.00149	0.00268	0.00446	0.05451	
4	1	0	0.32790	0.79782	0.53028	236	235	0.008162	0.002669	0.00535	0.000044	0.00166	0.00227	0.00499	0.05610	
5	1	0	0.50780	0.78744	0.65451	226	221	0.007631	0.002696	0.00783	0.000060	0.00232	0.00312	0.00697	0.07752	
6	2	1	0.76095	0.62145	0.54543	322	321	0.005991	0.000107	0.00222	0.000013	0.00036	0.00094	0.00108	0.03203	
7	2	1	0.83671	0.62079	0.51179	318	317	0.006074	0.000136	0.00282	0.000017	0.00034	0.00088	0.00103	0.06300	
8	2	1	0.80826	0.61766	0.50447	318	317	0.006057	0.000069	0.00161	0.000010	0.00027	0.00068	0.00081	0.02783	
9	3	0	0.85302	0.62247	0.54855	493	492	0.003910	0.000040	0.00075	0.000003	0.00009	0.00025	0.00027	0.06670	

from CSV files using the inbuilt python functions that are part of the pandas library.

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

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

```
man: 390 woman: 366
```

Fig. No - 3.4 Gender count

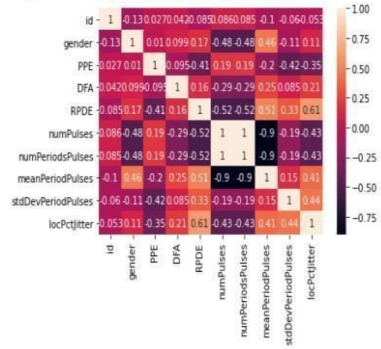
3.2.2 Data Pre-Processing

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

Actually, in this dataset, we have 755 features out of which some may not be useful in building our model. So, we have to leave out all those unnecessary features which are not responsible for producing the output. If we take more features in this model the accuracy we got is less. When we check the correlation of the features, some of them are the same. In Fig 3.5, a screenshot of our notebook is shown the correlation of the columns where two of the columns have similar values. So, one of them is removed.

```
sns.heatmap(df[df.columns[0:10]].corr(),annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0d998f1650>
```




```
df[["numPulses","numPeriodsPulses"]].corr()
```

	numPulses	numPeriodsPulses
numPulses	1.000000	0.999917
numPeriodsPulses	0.999917	1.000000

Fig. No - 3.5 Correlation matrix

As the correlation values of the two attributes are similar and one of them can be removed. This kind of feature must be dropped. As our data is now stored as a data frame in a python notebook, we can easily drop those unnecessary features using the inbuilt functions. In Fig 3.6, a screenshot of our notebook is shown where we have dropped some features.

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

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

```
df.info()

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

Fig. No - 3.6 Dropping unnecessary features from data frame

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

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

- stdDevPeriodPulses
- locPctJitter

After pre-processing the acquired data, the next step is to identify the best features. The identified best features should be able to give high efficiency. In Fig 3.7, a screenshot of our notebook is shown how to select k best features using scikit learn. The classes within the sklearn.feature_selection module are often used for feature selection/dimensionality reduction on sample sets, either to enhance estimators' accuracy scores or to spice up their performance on very high-dimensional datasets.

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

```
x.head()
```

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

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

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

Fig. No - 3.7 Process of Feature selection and sample data

3.2.2.1 Training Data

In machine learning data preprocessing, we have to break our dataset into both a training set and test set. This is often one among the crucial steps of knowledge preprocessing as by doing this, we will enhance the performance of our machine learning model. Suppose, if we've given training to our machine learning model by a dataset and that we test it by a totally different dataset. Then, it'll create difficulties for our model to know the correlations between the models. If we train our model alright and its training accuracy is additionally very high, but we offer a replacement dataset there too, then it'll decrease the performance. So we always attempt to make a machine learning model which performs well with the training set and also with the test dataset.

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

Fig. No - 3.8 Splitting dataset into training data and test data

Usually, we split the dataset into train and test in the ratio of 7:3 i.e., 70 percent of data is used for training and 30 percent of data is used for testing the model. We have done it in the same way and it has been shown in the above Fig 3.8.

3.2.2.2 Compared with Other Machine Learning Algorithms

Now, we've both the train and test data. The subsequent step is to spot the possible training methods and train our models. As this is often a classification problem, we've used three different classification methods KNN, Naïve Bayes, and Logistic Regression. Each algorithm has been run over the Training dataset and their performance in terms of accuracy is evaluated alongside the prediction wiped out the testing data set.

3.2.4.1 K-Nearest Neighbor

The k-nearest neighbors (KNN) algorithm may be a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand. It belongs to the supervised learning domain.

Let m be the amount of training data samples. Let p be an unknown point.

- Store the training samples in an array of data points arr[]. This means each element of this array represents a tuple (x, y).
- for i=0 to m:
- Calculate Euclidean distance d(arr[i],p)
- Make the set S of K smallest distances obtained. Each of those distances corresponds to an already classified datum.
- Return the majority label among S.

Let's see this algorithm can be seen with the help of a simple example. Suppose the dataset have two variables, which are plotted and shown in fig 3.9.

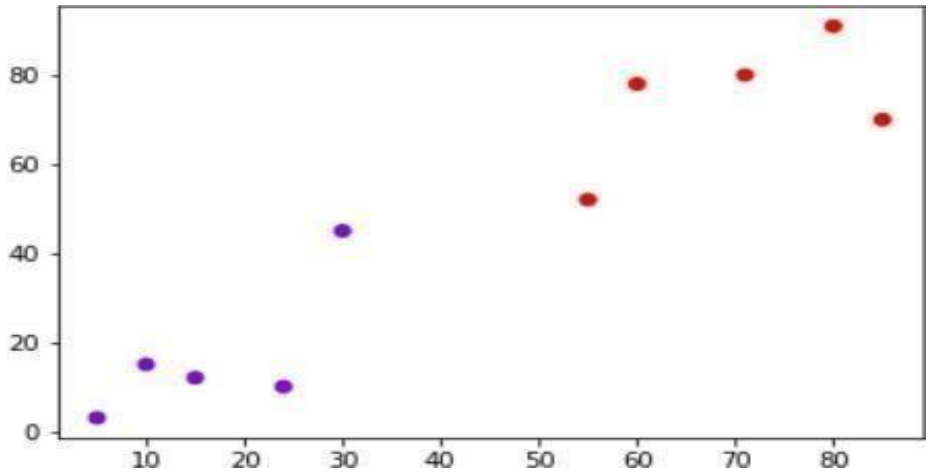


Fig. No - 3.9 KNN Graph-1

Your task is to classify a replacement datum with 'X' into the "Blue" class or "Red" class. The coordinate values of the info point are x=45 and y=50. If the K value is of 3 then the KNN algorithm starts by calculating the space of point X from all the points. Then it finds the nearest three points with the least distance to point X. This process can be shown in the fig 3.10. The three nearest points in the results have been encircled.

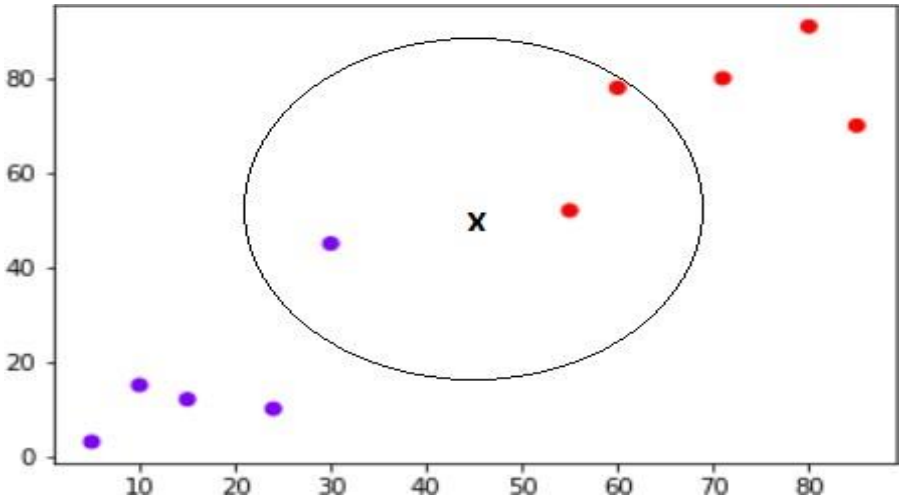


Fig. No - 3.10 KNN Graph-2

The final step of the KNN algorithm is to assign a replacement point to the category to which the bulk of the three nearest points belong. From the figure above we can see that two of the three nearest points belong to the class "Red" while one belongs to the class "Blue". Therefore the new datum is going to be classified as "Red".

In KNN, finding the value of K is not so easy. So we used an optimal way to identify the k value through error rate. We will find the error value at each k value and from that we identify the value which gives minimal error. It is shown in Fig 3.11.

```
error_rate = []
for i in range(1,100):
 knn = KNeighborsClassifier(n_neighbors=i)
 knn.fit(x_train,y_train)
 pred_i = knn.predict(x_test)
 error_rate.append(np.mean(pred_i != y_test))

plt.figure(figsize=(10,6))
plt.plot(range(1,100),error_rate,color='blue', linestyle='dashed',
         marker='o',markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
print("Minimum error:-",min(error_rate),"at K =",error_rate.index(min(error_rate)))
```

```
Minimum error:- 0.18061674008810572 at K = 10
```

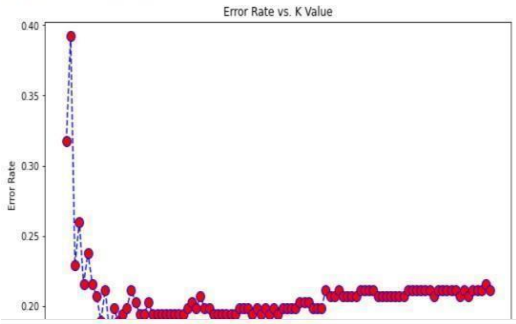


Fig. No - 3.11 Finding K value using error rate

In scikit-learn python library, from sklearn.neighbors import KNeighborsClassifier Module is used for carrying out the K Nearest Neighbor. We have to specify the value of K from the above and assign an object to the classifier. We will use our training dataset to fit the model.

Fig 3.12 shows the sample code for the training model using K-Nearest Neighbor.

```
k=10
knn = KNeighborsClassifier(n_neighbors = k)
knn.fit(x_train,y_train)
y_head=knn.predict(x_test)
print("KNN Algorithm test accuracy",knn.score(x_test,y_test))
```

```
KNN Algorithm test accuracy 0.801762114537445
```

```
classid,tn,fp,fn,tp=perf_measure(y_test,y_head)
auc_scor.append(roc_auc_score(y_test,y_head))
score_list.append(accuracy(classid,tn,fp,fn,tp))
precision_scor.append(precision(classid,tn,fp,fn,tp))
recall_scor.append(recall(classid,tn,fp,fn,tp))
f1_scor.append(f1_score(y_test,y_head,average='macro'))
NPV_scor.append(NPV(classid,tn,fp,fn,tp))
specificity_scor.append(specificity(classid,tn,fp,fn,tp))

LR_plus.append((recall(classid,tn,fp,fn,tp)/(1-specificity(classid,tn,fp,fn,tp))))
LR_minus.append(((1-recall(classid,tn,fp,fn,tp))/specificity(classid,tn,fp,fn,tp)))
odd_scor.append(((recall(classid,tn,fp,fn,tp)/(1-specificity(classid,tn,fp,fn,tp))))/(((1-recall(classid,tn,fp,fn,tp))/specificity(classid,tn,fp,fn,tp))))
youden_scor.append((recall(classid,tn,fp,fn,tp)+specificity(classid,tn,fp,fn,tp)-1))
print("KNN algorithm report: \n",classification_report(y_test,y_head))
```

```
KNN algorithm report:
              precision    recall  f1-score   support

           0       0.54      0.44      0.48        48
           1       0.86      0.90      0.88       179

    accuracy                           0.80       227
   macro avg       0.70      0.67      0.68       227
weighted avg       0.79      0.80      0.79       227
```

Fig. No - 3.12 KNN classifier model Naïve Bayes

Naive Bayes may be a statistical classification technique supported by Bayes Theorem. It is one of the simplest supervised learning algorithms. Naive Bayes classifier may be a fast, accurate, and reliable algorithm. Naive Bayes classifiers have high accuracy and speed on large datasets. Naive Bayes classifier assumes that the effect of a specific feature during a class is independent of other features. For example, a loan applicant is desirable or not counting on his/her income, previous loan and transaction history, age, and site. Even if these features are interdependent, these features are still considered independently. This assumption simplifies computation, and that's why it is considered naive. This assumption is called class conditional independence.

Naive Bayes classifier makes an assumption that every particular feature in the dataset is independent of all other features. For example, whether a patient is having Parkinson's or not depends on the speech features of the patient.

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)} \rightarrow \quad . \qquad (1)$$

P (h): the probability of hypothesis h being true (regardless of the data). This is known as the prior probability of h.

P (D): the probability of the data (regardless of the hypothesis). This is known as the prior probability.

P (h|D): the probability of hypothesis h given the data D. This is known as posterior probability.

P (D|h): the probability of data d given that the hypothesis h was true. This is known as posterior probability.

We can frame classification as a conditional classification problem with Bayes Theorem as follows:

P(yi | x1, x2, …, xn) = P(x1, x2, …, xn | yi) * P(yi) / P(x1, x2, …, xn)

The prior P(yi) is easy to estimate from a dataset, but the conditional probability of the observation based on the class P(x1, x2, , xn | yi) is not feasible unless the number of examples is extraordinarily large,large enough to effectively estimatethe probability distribution for all different possible combinations of values. As such, the direct application of Bayes Theorem also becomes intractable, especially as the number of variables or features (*n*) increases. Naive Bayes classifier calculates the probability of an event in the following steps:

Step 1: Calculate the prior probability for given class labels

Step 2: Find Likelihood probability with each attribute for each class

Step 3: Put these values in Bayes Formula and calculate posterior probability.

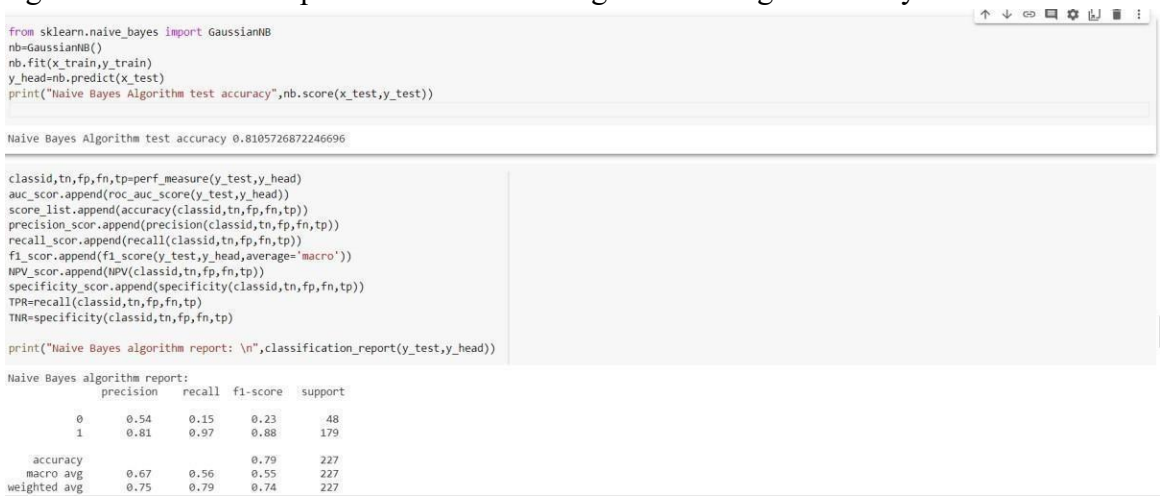
Step 4: See which class has a higher probability, given the input belongs to the higher probability class.

Types of Naive Bayes Algorithms:

- **Gaussian Naïve Bayes:** When the feature values are continuous in nature then there is an assumption to be made that the values linked with each category are dispersed according to Gaussian Normal Distribution.
- **Multinomial Naïve Bayes:** Multinomial Naive Bayes is mostly favored to be used on the data that is multinomial distributed. It is widely utilized in text classification in NLP. Each event in text classification constitutes the presence of a word in a document.
- **Bernoulli Naïve Bayes:** When data is deleted according to the multivariate Bernoulli distributions then comes the Bernoulli Naive Bayes. That means there can exist a

different number of features but each one is assumed to contain a binary value. So, it requires features to be binary-valued.

In scikit-learn python library, from sklearn.naive_bayes import GaussianNB Module is used for carrying out the Naïve Bayes classifier. We will use our training dataset to fit the model.
Fig 3.13 shows the sample code for the training model using Naïve Bayes.



```
from sklearn.naive_bayes import GaussianNB
nb=GaussianNB()
nb.fit(x_train,y_train)
y_head=nb.predict(x_test)
print("Naive Bayes Algorithm test accuracy",nb.score(x_test,y_test))
```

```
Naive Bayes Algorithm test accuracy 0.8105726872246696
```

```
classid,tn,fp,fn,tp=perf_measure(y_test,y_head)
auc_scor.append(roc_auc_score(y_test,y_head))
score_list.append(accuracy(classid,tn,fp,fn,tp))
precision_scor.append(precision(classid,tn,fp,fn,tp))
recall_scor.append(recall(classid,tn,fp,fn,tp))
f1_scor.append(f1_score(y_test,y_head,average='macro'))
NPV_scor.append(NPV(classid,tn,fp,fn,tp))
specificity_scor.append(specificity(classid,tn,fp,fn,tp))
TPR=recall(classid,tn,fp,fn,tp)
TNR=specificity(classid,tn,fp,fn,tp)

print("Naive Bayes algorithm report: \n",classification_report(y_test,y_head))
```

```
Naive Bayes algorithm report:
              precision    recall  f1-score   support

           0       0.54      0.15      0.23        48
           1       0.81      0.97      0.88       179

    accuracy                           0.79       227
   macro avg       0.67      0.56      0.55       227
weighted avg       0.75      0.79      0.74       227
```

Fig. No - 3.13 Naïve Bayes Classifier Model

3.2.4.3 Logistic Regression

Logistic regression is additionally one among the foremost popular Machine Learning algorithms, which comes under the Supervised Learning technique. it's used for predicting the specific variable employing a given set of independent variables. It becomes a classification technique only a choice threshold is brought into the image. The setting of the edge value may be a vital aspect of Logistic regression and depends on the classification problem itself.

The decision for the worth of the edge value is majorly suffering from the values of precision and recall. Ideally, we would like both precision and recall to be 1, but this seldom is that the case. within the case of Precision-Recall tradeoffs we use the subsequent arguments to make a decision upon the threshold:

- **Low Precision/High Recall:** In applications where we would like to scale back the amount of false negatives without necessarily reducing the amount of false positives, we elect a choice value that features a low value of Precision or a high value of Recall.
- **High Precision/Low Recall:** In applications where we would like to scale back the amount of false positives without necessarily reducing the amount of false Negatives, we elect a choice value that features a high value of Precision or a low value of Recall.

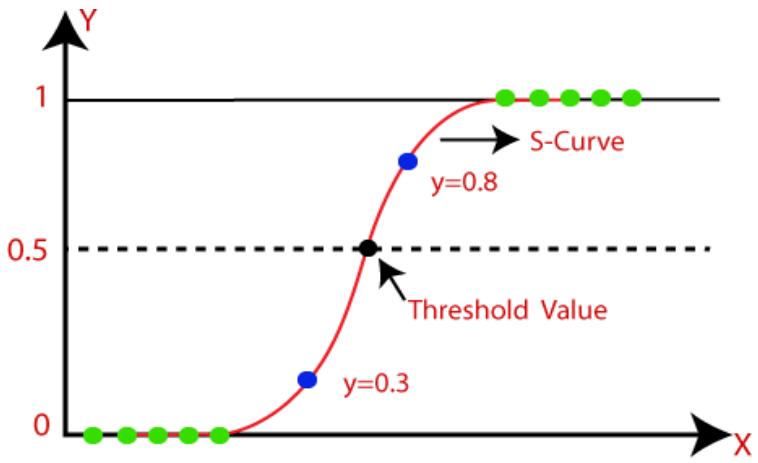


Fig. No - 3.14 S-shaped curve

The sigmoid function is a mathematical function used to map the predicted values to probabilities. It maps any real value into another value within a variety of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot transcend this limit, so it forms a curve just like the "S" form. The S-form curve is called the sigmoid function or the logistic function which is shown above in Fig3.14.

x = input to the function e = base of natural logarithm.

On the idea of the categories, Logistic Regression are often classified into three types:

- **Binomial:** The target variable can have only 2 possibilities either “0” or “1” which may represent “win” or “loss”, “pass” or “fail”, “dead” or “alive”, etc.
- **Multinomial:** In multinomial Logistic regression, the target variable can have 3 or more possibilities which are not ordered that means it has no measure in quantity like “disease A” or “disease B” or “disease C”.
- **Ordinal:** In ordinal Logistic regression, the target variables deals with ordered categories. For example, a test score can be categorized as: “very poor”, “poor”, “good”, and “very good”. Here, each category can be given a score like 0, 1, 2, and 3.

In scikit-learn python library, from sklearn.linear_model importLogisticRegression Module is used for carrying out the Logistic Regression. We have to specify the iterations to the function parameter and assign an object to the classifier. We will use our training dataset to fit the model. Fig 3.15 shows the sample code for the training model using Logistic Regression.

```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression(random_state=0,max_iter=300)
lr.fit(x_train,y_train)
y_head=lr.predict(x_test)
print("Logistic Regression testaccuracy ",lr.score(x_test,y_test))
```

```
Logistic Regression testaccuracy  0.7929515418502202
```

```
classid,tn,fp,fn,tp=perf_measure(y_test,y_head)
auc_scor.append(roc_auc_score(y_test,y_head))
score_list.append(accuracy(classid,tn,fp,fn,tp))
precision_scor.append(precision(classid,tn,fp,fn,tp))
recall_scor.append(recall(classid,tn,fp,fn,tp))
f1_scor.append(f1_score(y_test,y_head,average='macro'))
NPV_scor.append(NPV(classid,tn,fp,fn,tp))
specificity_scor.append(specificity(classid,tn,fp,fn,tp))
```

```
Logistic Regression report:
              precision    recall  f1-score   support

           0       0.54      0.15      0.23        48
           1       0.81      0.97      0.88       179

    accuracy                           0.79       227
   macro avg       0.67      0.56      0.55       227
weighted avg       0.75      0.79      0.74       227
```

Fig. No - 3.15 Logistic Regression model

3.2.5 Testing Data

Once Parkinson’s disease Prediction model has been trained on the preprocessed dataset, then the model is tested using different data points. In this testing step, the model is checked for correctness and accuracy by providing a test dataset to it. All the training methods need to be verified for finding out the best model to be used. In figures 3.12, 3.13, 3.15, after fitting our

model with training data, we used this model to predict values for the test dataset. These predicted values on testing data are used for model comparison and accurate calculation.

3.3 Light GBM

Our Front-End implementation is completed using HTML, CSS, JavaScript and Flask Framework in the Scientific Python Development Environment (Spyder). The user interface is extremely essential for any project because everyone who tries to utilize the system for a purpose will attempt to access it using an interface. Indeed, our system also features a user interface built to facilitate users to utilize the services we provide. Where we have used HTML terminology, utilized for creating web sites. One of the useful aspects of HTML is, it can embed programs written during a scripting language like JavaScript, which is liable for affecting the behavior and content of web pages. CSS inclusion would affect the layout and appearance of the content.

Flask gives all sorts of choices when developing web applications. It provides you with tools, libraries, and mechanics that allow you to build, create various applications but it will not enforce any dependencies or tell you the way how the project should look like.

We are using the Light Gradient Boosting Machine technique for prediction and classifying the disease because its accuracy is high compared to Decision tree, Naïve Bayes, SVM algorithms. Also, the error rate is comparatively lesser than the Decision tree, Naïve Bayes, SVM and other frequently used algorithms. Also, when the data is complex LGBM gives a better result and accuracy.

LIGHT GRADIENT BOOSTING MACHINE ALGORITHM

Light GBM is a gradient boosting framework based on decision trees to increases the efficiency of the model and reduces memory usage. It uses two novel techniques: Gradient-based One Side Sampling and Exclusive Feature Bundling (EFB) which fulfills the limitations of histogram-based algorithm that is primarily used in all GBDT (Gradient Boosting Decision Tree) frameworks. The two techniques of GOSS and EFB described below form the characteristics of Light GBM Algorithm. They comprise together to make the model work efficiently and provide it a cutting edge over other GBDT frameworks.

High-dimensional data are usually very sparse which provides us a possibility of designing a nearly lossless approach to reduce the number of features. Specifically, in a sparse feature space, many features are mutually exclusive, i.e., they never take nonzero values simultaneously. The exclusive features can be safely bundled into a single feature (called an Exclusive Feature Bundle). Hence, the speed for training framework is improved without hurting accuracy in Light GBM.

FEATURES

Our project has following features to be shared of

- Uploading the own dataset as CSV from the user-end.
- Predicts the presence of Parkinson's disease with Light GBM withthe accuracy of each and every record are to be saved.
- Download the predicted data as CSV with Label and Accuracy score .

CHAPTER-4

EXPERIMENTAL ANALYSIS AND RESULTS

4.1.1 System Requirements

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

Requirements can be divided into two major categories:

- Functional Requirements.
- Non-Functional Requirements.

4.1.2 Functional Requirements

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

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

4.1.3 Non-Functional Requirements

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

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

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

- Accuracy
- Reliability
- Flexibility

4.2 System Configuration 4.2.1

Software Requirements ●

Software:

- Spyder
- Google Colab
- Operating System: Windows 10

- Tools: Web Browser
- Python Libraries: numpy, pandas, matplotlib, seaborn, sklearn, pickle.

4.2.1.1 Introduction to Python

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

Python does the following:

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

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

4.2.1.2 Introduction to Flask Framework

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

Flask constructor takes the name of the current module (name) as argument.The route () function of the Flask class may be a decorator, which tells the appliance which URL should call the associated function.

The rule parameter represents URL binding with the function.
The options is an inventory of parameters to be forwarded to the underlying Rule object.
Finally the run() method of Flask class runs the application on the local development server.
Open the URL (localhost:5000) in the browser. The message will be displayed on it.

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

4.2.1.3 Python Libraries

NumPy

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

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

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

Pandas

Pandas is an open-source library that's built on top of NumPy library. It is a Python package that gives various data structures and operations for manipulating numerical data and time series. It is fast and it has high-performance & productivity for users. It provides high-performance and easy-to-use data structures and data analysis tools for the Python language.

Pandas is employed during a wide range of fields including academic and commercial domains including economics, Statistics, analytics, etc.

Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It is an open-source Python library that implements a variety of machine learning, pre-processing, cross-validation and visualization algorithms employing a unified interface. Sklearn provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. This library, which is essentially written in Python, is made upon NumPy, SciPy and Matplotlib.

Pickle

Python pickle module is employed for serializing and de-serializing a Python object structure. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the thing in another python script. Pickling is beneficial for applications where you would like a point of persistence in your data. Your program's state data are often saved to disk, so you'll continue working on it later on.

Matplotlib

It is a very powerful plotting library useful for those working with Python and NumPy. And for creating statistical inference, it becomes very necessary to visualize our data and Matplotlib is the tool which will be very helpful for this purpose. It provides a MATLAB like interface. The only difference is that it uses Python and is open source.

Seaborn

Seaborn may be a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

It offers the following functionalities:

- Dataset oriented API to determine the relationship between variables.
- Automatic estimation and plotting of linear regression plots.
- It supports high-level abstractions for multi-plot grids.

- Visualizing univariate and bivariate distribution.

4.2.2 Hardware Requirements

- RAM: 4 GB or above
- Storage: 30 to 50 GB
- Processor: Any Processor above 500MHz

4.3 Feasibility Study

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

- Economical Feasibility
- Technical Feasibility
- Operational Feasibility

4.3.1 Economic Feasibility

As systems are often developed technically which are going to be used if installed must still be an honest investment for the organization. In the economic feasibility, the event cost in creating the system is evaluated against the last word benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It doesn't require any additional hardware or software. Since the interface for this system is developed using the existing resources and technologies java1.6 open source, there is nominal expenditure and economic feasibility for certain.

4.3.2 Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the

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

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

4.3.3 Operational Feasibility

Proposed projects are beneficial only if they can be turned into an information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as a crucial part of the project implementation.

Some of the important issues raised are to check the operational feasibility of a project includes the following:

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?

4.4 Sample Code

4.4.1. Data Pre-Processing

```
[1]  ▷ ▸≡ M↓
     import pandas as pd
     data = pd.read_csv('./Data/pd_speech_features.csv')
```

Loading the training dataset

```
[3]
data.isnull().values.any() # No null/missing data found

False
```

Checking for null values in the dataset

```
[4]
#import cla.ssification module
from pycaret.classification import *

#intialize the setup/preprocessing
dataSetup = setup(data, target = 'class', train_size = 0.7)
```

Preprocessing the initial data

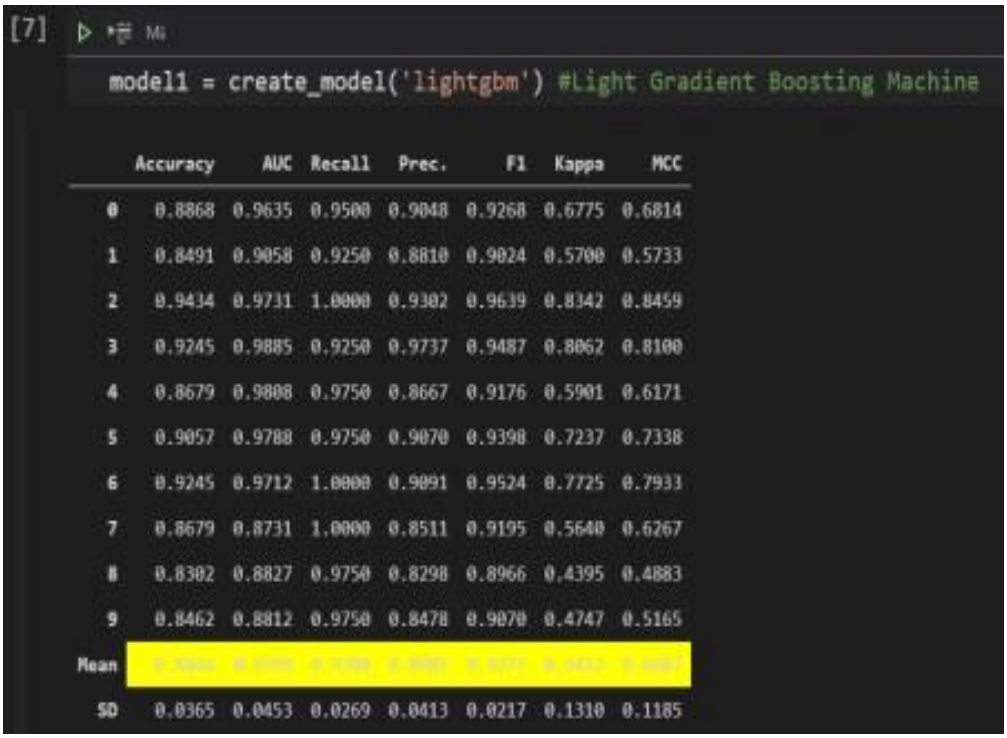
```
[5]
# return best model
best = compare_models()
```

Comparing all ML models

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
lightgbm	Light Gradient Boosting Machine	[illegible]	0.9399	0.9700	[illegible]	[illegible]	[illegible]	[illegible]
et	Extra Trees Classifier	0.8808	[illegible]	[illegible]	0.8771	0.9262	0.6196	0.6564
gbc	Gradient Boosting Classifier	0.8790	0.9283	0.9775	0.8781	0.9248	0.6184	0.6433
rf	Random Forest Classifier	0.8619	0.9091	0.9775	0.8608	0.9150	0.5530	0.5891
ada	Ada Boost Classifier	0.8582	0.8884	0.9350	0.8843	0.9085	0.5928	0.6018
lr	Logistic Regression	0.8109	0.7841	0.9375	0.8337	0.8820	0.4065	0.4318
ridge	Ridge Classifier	0.7940	0.0000	0.9325	0.8206	0.8722	0.3432	0.3764
dt	Decision Tree Classifier	0.7900	0.7269	0.8500	0.8704	0.8590	0.4449	0.4492
nb	Naive Bayes	0.7466	0.7232	0.8775	0.8062	0.8372	0.2458	0.2701
knn	K Neighbors Classifier	0.7203	0.6279	0.8850	0.7762	0.8262	0.1103	0.1280
svm	SVM - Linear Kernel	0.6918	0.0000	0.8200	0.8024	0.7826	0.0854	0.1020
lda	Linear Discriminant Analysis	0.6709	0.6611	0.6925	0.8467	0.7593	0.2501	0.2660
qda	Quadratic Discriminant Analysis	0.3457	0.5026	0.1975	0.3510	0.1790	0.0028	0.0162

Comparison of Models

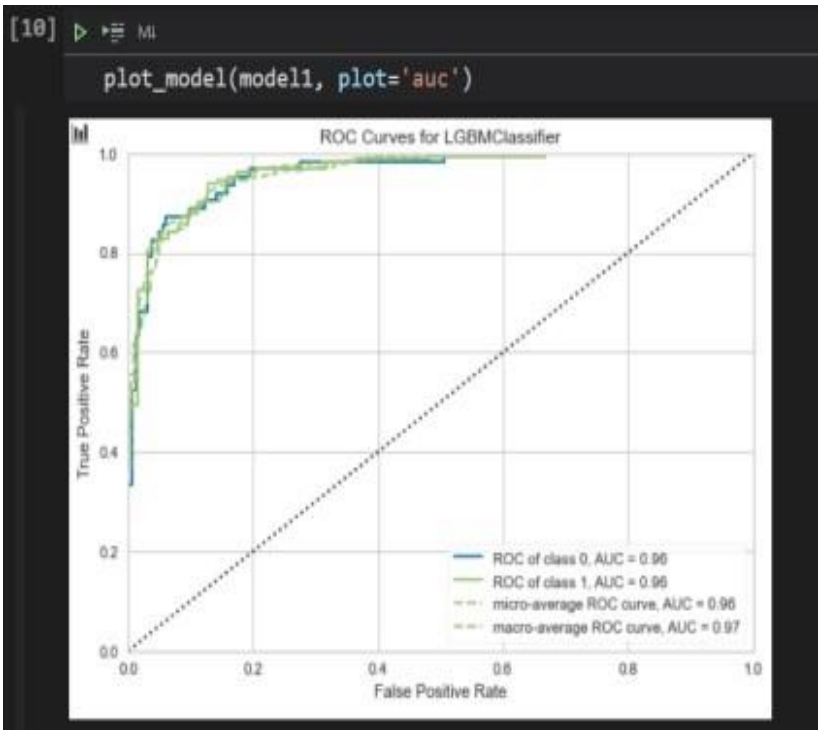
4.4.2 Light Gradient Boosting Machine:



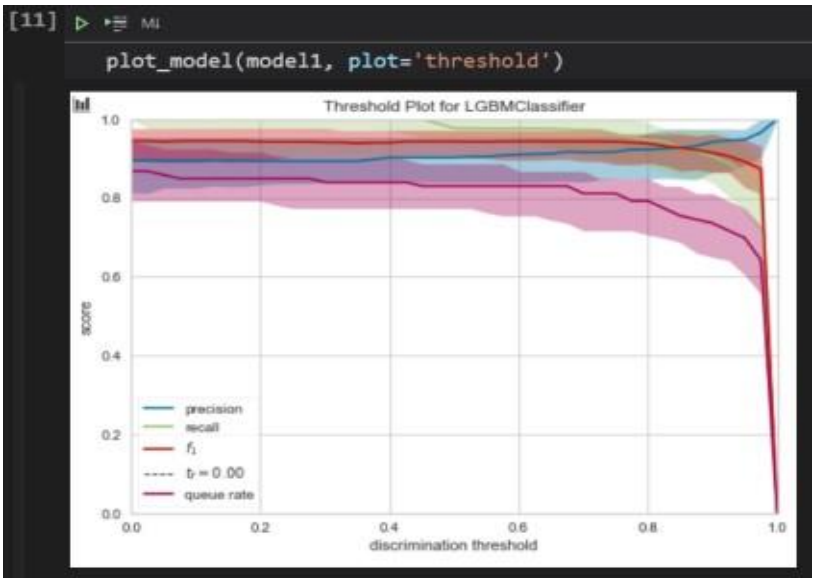
```
[7]
model1 = create_model('lightgbm') #Light Gradient Boosting Machine
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.8868	0.9635	0.9500	0.9048	0.9268	0.6775	0.6814
1	0.8491	0.9058	0.9250	0.8810	0.9024	0.5700	0.5733
2	0.9434	0.9731	1.0000	0.9302	0.9639	0.8342	0.8459
3	0.9245	0.9885	0.9250	0.9737	0.9487	0.8062	0.8100
4	0.8679	0.9808	0.9750	0.8667	0.9176	0.5901	0.6171
5	0.9057	0.9788	0.9750	0.9070	0.9398	0.7237	0.7338
6	0.9245	0.9712	1.0000	0.9091	0.9524	0.7725	0.7933
7	0.8679	0.8731	1.0000	0.8511	0.9195	0.5640	0.6267
8	0.8302	0.8827	0.9750	0.8298	0.8966	0.4395	0.4883
9	0.8462	0.8812	0.9750	0.8478	0.9070	0.4747	0.5165
Mean	[illegible]						
SD	0.0365	0.0453	0.0269	0.0413	0.0217	0.1310	0.1185

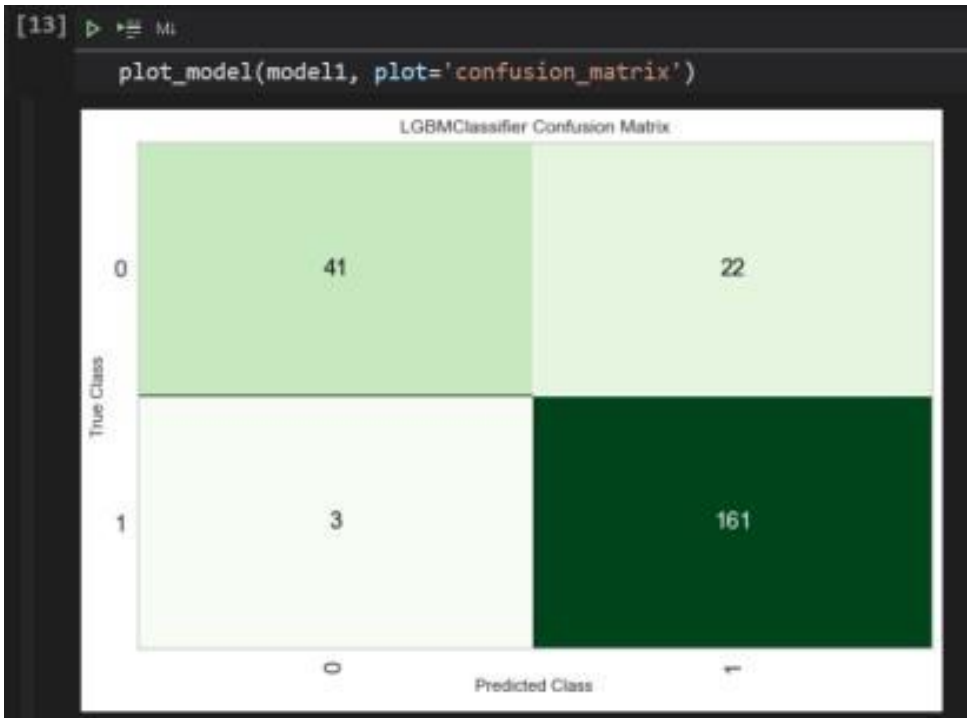
Creating the model for LGBM



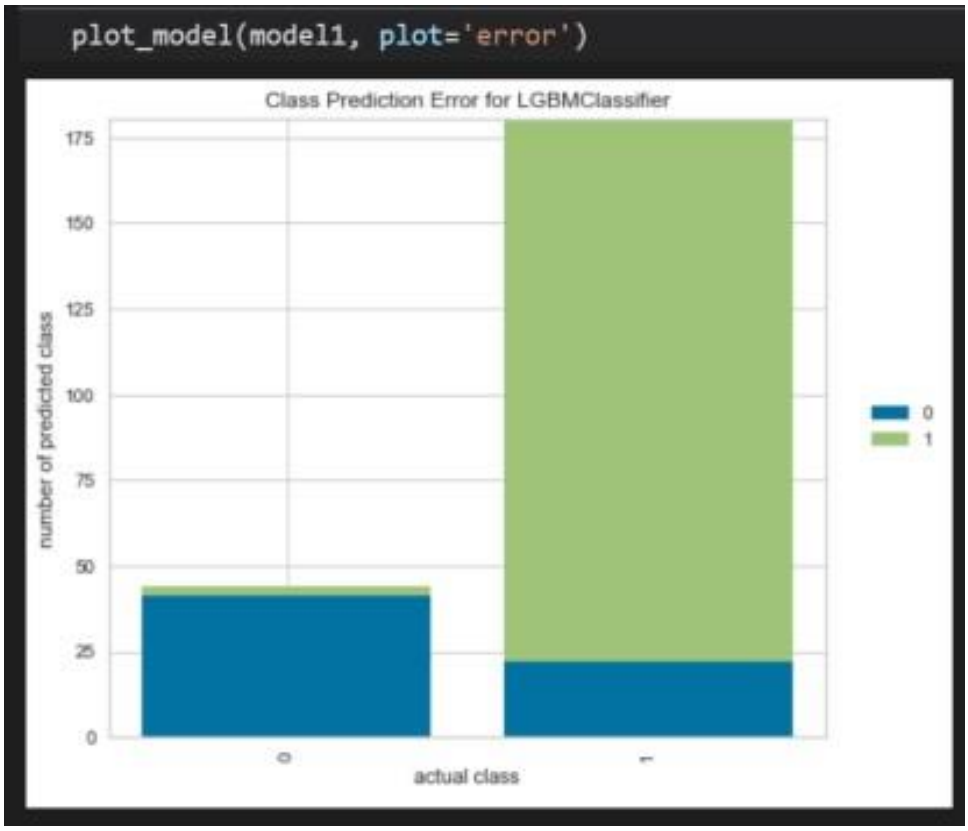
Area Under Cure for LGBM



Threshold on LGBM

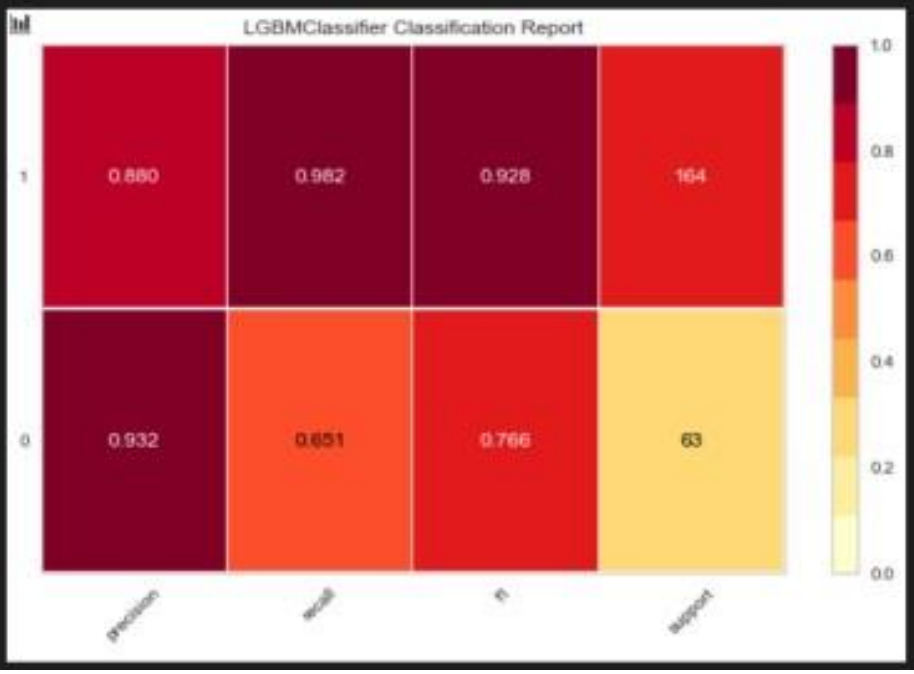


Confusion Matrix before finalization



Error rate on classification

```
[15]
plot_model(model1, plot='class_report')
```



Class Report on precision, recall, accuracy and support with classification

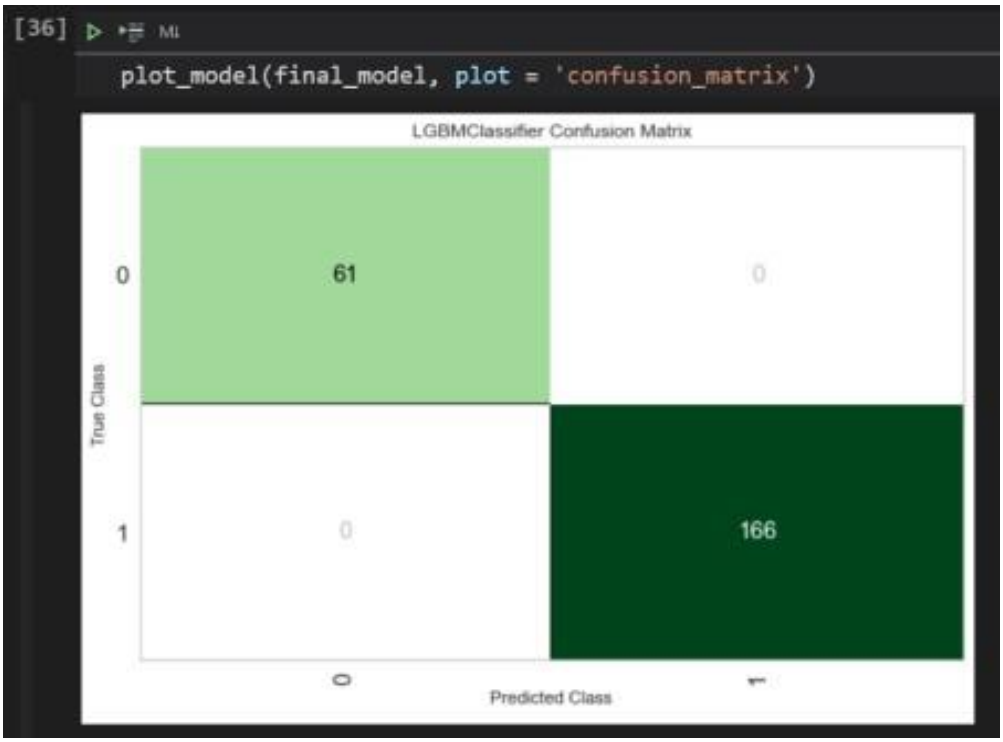
```
[33]
final_model = finalize_model(model1)

[34]
final_model

LGBMClassifier(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0,
               importance_type='split', learning_rate=0.1, max_depth=-1,
               min_child_samples=20, min_child_weight=0.001, min_split_gain=0.0,
               n_estimators=100, n_jobs=-1, num_leaves=31, objective=None,
               random_state=4602, reg_alpha=0.0, reg_lambda=0.0, silent=True,
               subsample=1.0, subsample_for_bin=200000, subsample_freq=0)

[35]
save_model(final_model, 'model_data_mining')
Transformation Pipeline and Model Succesfully Saved
```

Finalization of LGBM with its parameters and saving the final model



Confusion Matrix after finalization

The model have been saved as **model_data_mining.pkl** with all the parameters that have been optimized for the

4.3. Code

App.py (Flask Server)

```
import os
from flask import * import
pandas as pd import
chart_studio import
chart_studio.tools as tls import
chart_studio.plotly as py import
plotly.express as px
from pycaret.classification import * from
werkzeug.utils import secure_filename from
dotenv import load_dotenv load_dotenv()
```

```
tls.set_credentials_file(os.getenv('USERID'),os.getenv('APIKEY'))


#Changing the upload directory
UPLOAD_FOLDER = 'D:\\Study\\Winter 2020 -
2021\\Data
Mining\\Project \\Review
3\\uploads'
app = Flask(__name__,static_folder= './frontend',
template_folder= '
./frontend/views')
app.config[ 'UPLOAD_FOLDER'] =
UPLOAD_FOLDER
def
computePrediction():
 global lengthD,acc1,acc2,acc3,plot1,plot2,plot3
DM_loaded = load_model('model_data_mining') data
= pd.read_csv('./uploads/' + fileName) predictions =
predict_model(DM_loaded, data=data)
 lengthD = len(predictions)
 acc1 = round((predictions['Score'].mean())*100,2) acc2 =
round(predictions[predictions['Label']==0]['Score'].mean()*100,2)
 acc3 = round(predictions[predictions['Label']==1]['Score'].mean()*100,2)
                                                    # Graph
 label_info = pd.DataFrame(predictions['Label'].value_counts().re 1
set_index().values, columns=["Label", "No. of Data"]) p1 =
px.pie(label_info,names=label_info['Label'],values=label_in
'No. of
Data'],title="Label of Parkinson Disease") plot1 = fo[
 _open
tls.get_embed(py.plot(p1,filename='Parkinson Disease',au
 # Graph 2
                                                    =False))


 male_count = predictions[predictions['gender']==1] male_info =
pd.DataFrame(male_count['Label'].value_counts().rese t_index().values,
columns=["Label", "No. of Data"])
 p2=px.pie(male_info,names=male_info['Label'],values=male_info['N o. of
Data'],title="Male info of Parkinson Disease") plot2 =
tls.get_embed(py.plot(p2,filename='Male info of Parkinso n
Disease',auto_open=False))
```

```
# Graph 3  female_count = predictions[predictions['gender']==0]
female_info = pd.DataFrame(female_count['Label'].value_counts().
reset_index().values, columns=["Label", "No. of Data"])


p3=px.pie(female_info,names=female_info['Label'],values=female_i
nfo['No. of Data'],title="Female info of Parkinson Disease") plot3 =
tls.get_embed(py.plot(p3,filename='Female info of Parkin son
Disease',auto_open=False))
predictions.to_csv('out.csv', index=False)
var = '<iframe id="igraph" scrolling="no" style="border:none;" seam less="seamless"
src="https://plotly.com/~santhosh_d/1.embed" height= "525"
width="100%"></iframe>'
@app.route('/', methods = ['GET', 'POST']) def
home():
 global fileName  if
request.method == 'POST':
 file = request.files['csv']  fileName =
secure_filename(file.filename)
 file.save(os.path.join(app.config['UPLOAD_FOLDER'], fileName  ))
 computePrediction()

return    render_template('upload.html',lengthD    =    lengthD,    acc1
=acc1,acc2=acc2,acc3=acc3,plot1=plot1,plot2=plot2,plot3=plot3)  else:
return render_template('index.html')

# Upload page
@app.route('/upload', methods = ['GET', 'POST'])
def upload():
 if request.method == 'POST':
 return render_template('upload.html',var=var)                    else:
 return render_template('upload.html')
# Download Action
@app.route("/download") def

return send_from_directory(directory=r'D:\Study\Winter 2020 - 20
Mining\Project\Review 3', filename='out.csv',as_attachment=T rue)
download():
```

```
21\Data


if __name__ == '__main__':
app.run()
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```
  <meta
   name="description"
   content="Parkinson Disease Prediction and Classification."
  />
  <title>Parkinson Disease | Data Mining</title>
  <script
   src="https://kit.fontawesome.com/f3a06c157c.js"
   crossorigin="anonymous"
  ></script>
  <link rel="stylesheet" href="../css/style.css" />
  <link rel= "stylesheet" type= "text/css" href= "{{ url_for('static',filename='css/style.css') }}"
>
 </head>
 <body>
  <header>
   <nav id="navbar">
    <div class="container">
     <h1 class="logo"><a href="index.html">Parkinson Disease</a></h1>
     <ul>
      <li><a class="current" href="/">Home</a></li>
      <li><a href="upload">Upload</a></li>
     </ul>
    </div>
   </nav>
   <div id="showcase">
```

```
      <div class="container">
        <div class="showcase-content">
          <h1>
            <span class="text-highlight">Parkinson</span> Disease
            Classification and Prediction
          </h1>
          <p class="text-big">
            Parkinson's disease is a progressive nervous system disorder that
            affects movement. Symptoms start gradually, sometimes starting
            with a barely noticeable tremor in just one hand. Tremors are
            common, but the disorder also commonly causes stiffness or slowing
            of movement.
          </p>
          <section id="contact-form" class="py-3">
            <form action="/" method="POST" enctype=multipart/form-data >
              <div class="form-group">
                <input type="file" id="csv" name="csv" accept=".csv" />
              </div>
              <button type="submit" class="btn">Submit</button>
            </form>
          </section>
        </div>
      </div>
    </div>
  </header>
  <div class="clr"></div>
  <footer id="main-footer">
    <p>
      Parkinson Disease Prediction & Classification &copy 2021, All Rights
      Reserved
    </p>
  </footer>
 </body>
</html>
```

Upload.html

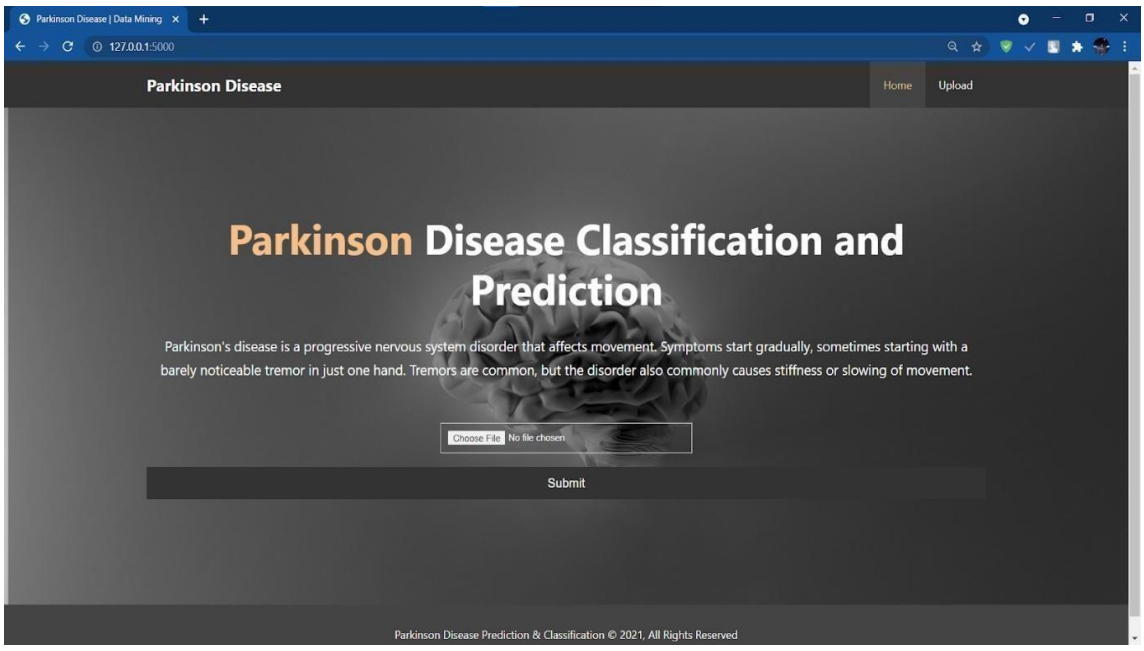
```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta
    name="description"
    content="Parkinson Disease Prediction and Classification."
  />
  <title>Parkinson Disease | Data Mining</title>
  <script
    src="https://kit.fontawesome.com/f3a06c157c.js"
    crossorigin="anonymous"
  ></script>
  <link rel="stylesheet" href="../css/style.css" />
  <link
    rel="stylesheet"
    type="text/css"
    href="{{ url_for('static',filename='css/style.css') }}"
  />
</head>
<body>
  <header>
    <nav id="navbar">
      <div class="container">
        <h1 class="logo"><a href="/">Parkinson Disease</a></h1>
        <ul>
          <li><a href="index.html">Home</a></li>
          <li><a class="current" href="upload">Upload</a></li>
        </ul>
```

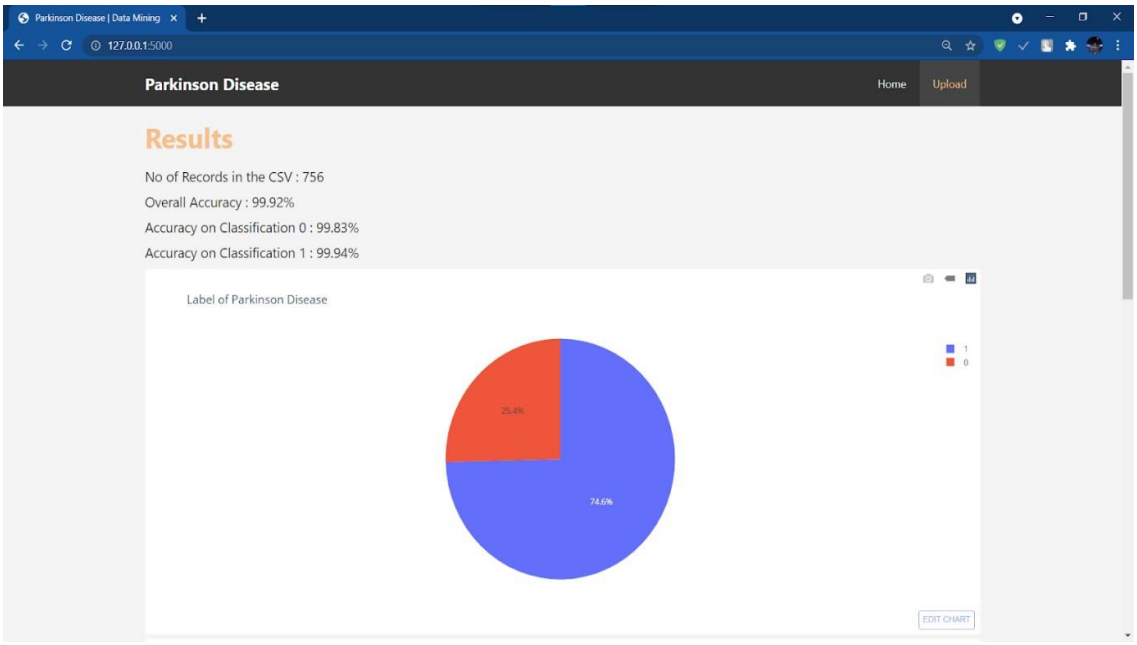
```
      </div>
    </nav>
  </header>
  <section id="about-info" class="bg-light py-3">
    <div class="container">
      <h1 class="l-heading">
        <span class="text-highlight">Results</span>
      </h1>
      <div class="showcase-content">
```

```
      <p class="text-big">No of Records in the CSV : {{lengthD}}</p>
      <p class="text-big">Overall Accuracy : {{acc1}}%</p>
      <p class="text-big">Accuracy on Classification 0 : {{acc2}}%</p>
      <p class="text-big">Accuracy on Classification 1 : {{acc3}}%</p>
    </div>
    <div>{{plot1|safe}}</div>
    <div>{{plot2|safe}}</div>
    <div>{{plot3|safe}}</div>
    <form action="/download">
      <button type="submit" class="btn">Download Predicted File</button>
    </form>
  </div>
 </section>
 <div class="clr"></div>
 <footer id="main-footer">
  <p>
   Parkinson Disease Prediction & Classification &copy 2021, All Rights
   Reserved
  </p>
 </footer>
</body>
</html>
```

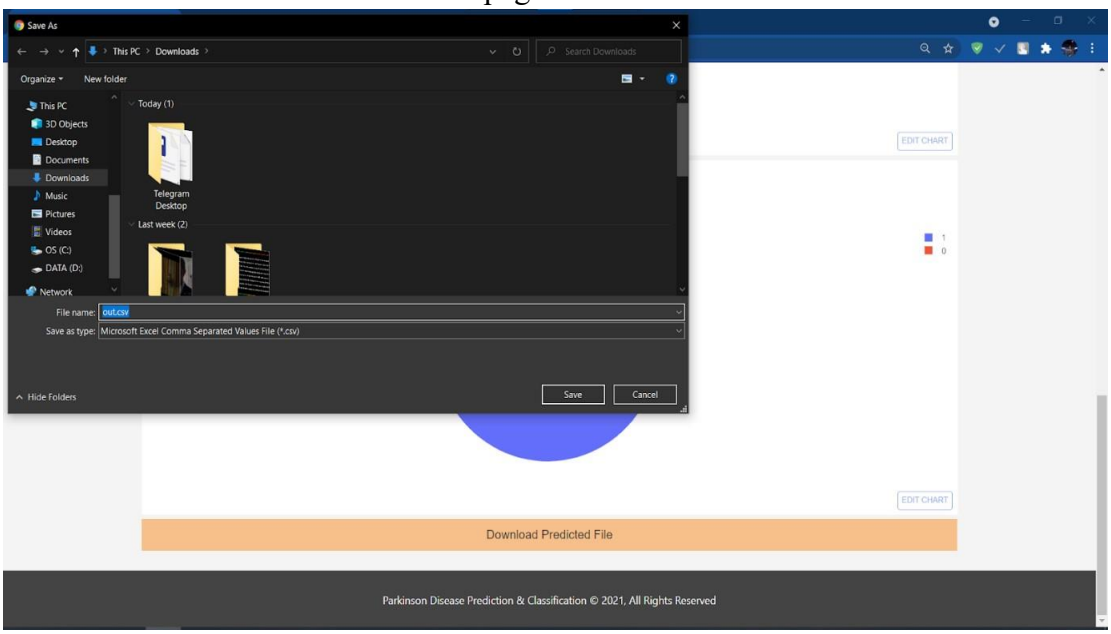
4.4. RESULTS



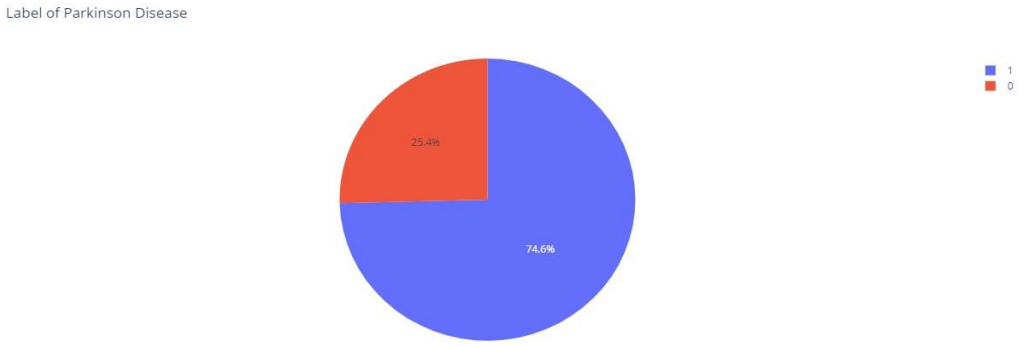
Parkinson's Disease Home Page



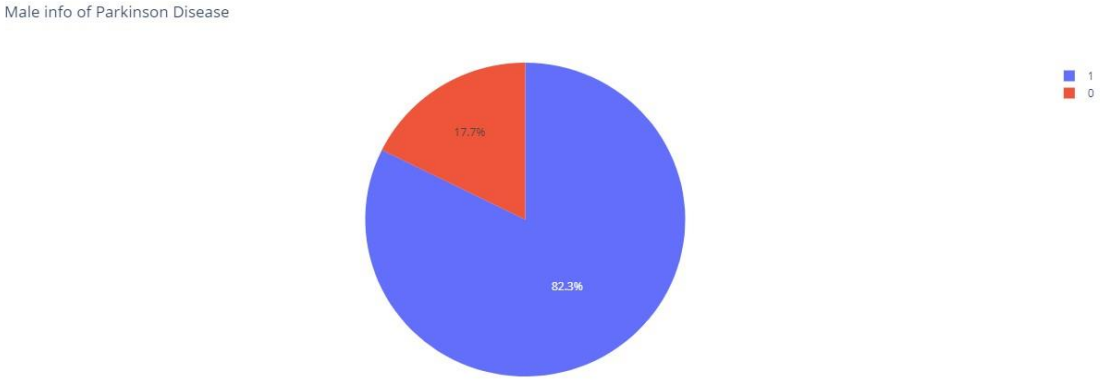
Results page of user dataset



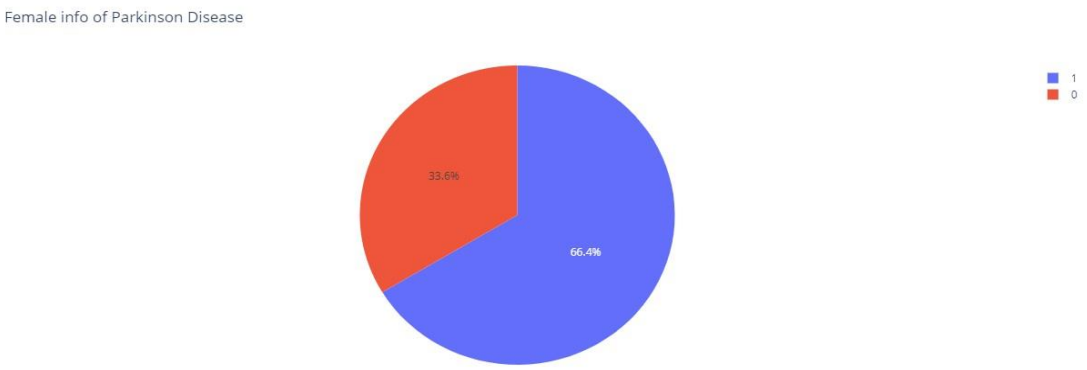
Output CSV file of the User data with predicted label and accuracy scores



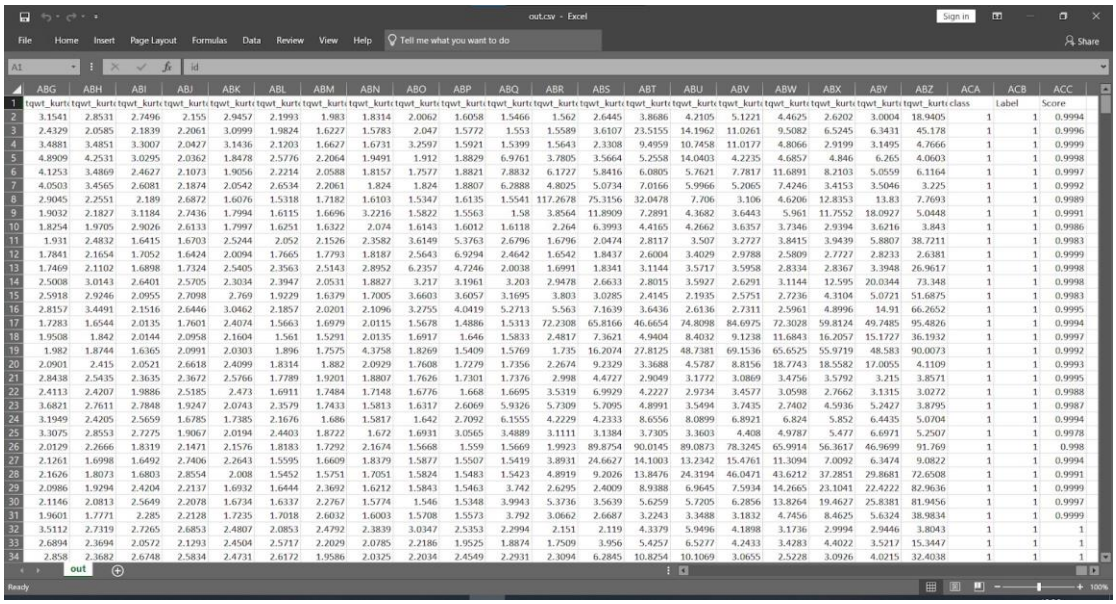
No. of positive and negative Parkinson disease patient in the given dataset



No. of Male positive and negative Parkinson's patient in the Dataset



No. of Female positive and negative Parkinson's patient in the Dataset



ABG	ABH	ABI	ABJ	ABK	ABL	ABM	ABN	ABO	ABP	ABQ	ABR	ABS	ABT	ABU	ABV	ABW	ABX	ABY	ABZ	ACA	ACB	ACC
tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	tqwt_kurto	class	Label	Score
3.1541	2.8531	2.7496	2.155	2.9457	2.1993	1.983	1.8314	2.0062	1.6058	1.5466	1.562	2.6445	3.8686	4.2105	5.1221	4.4625	2.6202	3.0004	18.9405	1	1	0.9994
2.4329	2.0585	2.1839	2.2061	3.0999	1.9824	1.6227	1.5783	2.047	1.5772	1.553	1.5589	3.6107	23.5155	14.1962	11.0261	9.5082	6.5245	6.3431	45.178	1	1	0.9996
3.4881	3.4851	3.3007	2.0427	3.1436	2.1203	1.6627	1.6731	3.2597	1.5921	1.5399	1.5643	2.3308	9.4959	10.7458	11.0177	4.8066	2.9199	3.1495	4.7666	1	1	0.9999
4.8909	4.2531	3.0295	2.0362	1.8478	2.5776	2.2064	1.9491	1.912	1.8829	6.9761	3.7805	3.5664	5.2558	14.0403	4.2235	4.6857	4.846	6.265	4.0603	1	1	0.9998
4.1253	3.4869	2.4627	2.1073	1.9056	2.2214	2.0588	1.8157	1.7577	1.8821	7.8832	6.1727	5.8416	6.0805	5.7621	7.7817	11.6891	8.2103	5.0559	6.1164	1	1	0.9997
4.0503	3.4565	2.6081	2.1874	2.0542	2.6534	2.2061	1.824	1.824	1.8807	6.2888	4.8025	5.0734	7.0166	5.9966	5.2065	7.4246	3.4153	3.5046	3.225	1	1	0.9992
2.9045	2.2551	2.189	2.6872	1.6076	1.5318	1.7182	1.6103	1.5347	1.6135	1.5541	117.2678	75.3156	32.0478	7.706	3.106	4.6206	12.8353	13.83	7.7693	1	1	0.9989
1.9032	2.1827	3.1184	2.7436	1.7994	1.6115	1.6696	3.2216	1.5822	1.5563	1.58	3.8564	11.8909	7.2891	4.3682	3.6443	5.961	11.7552	18.0927	5.0448	1	1	0.9991
1.8254	1.9705	2.9026	2.6133	1.7997	1.6251	1.6322	2.074	1.6143	1.6012	1.6118	2.264	6.3993	4.4165	4.2662	3.6357	3.7346	2.9394	3.6216	3.843	1	1	0.9986
1.931	2.4832	1.6415	1.6703	2.5244	2.052	2.1526	2.3582	3.6149	5.3763	2.6796	1.6796	2.0474	2.8117	3.507	3.2727	3.8415	3.9439	5.8807	38.7211	1	1	0.9983
1.7841	2.1654	1.7052	1.6424	2.0094	1.7665	1.7793	1.8187	2.5643	6.9294	2.4642	1.6542	1.8437	2.6004	3.4029	2.9788	2.5809	2.7727	2.8233	2.6381	1	1	0.9999
1.7469	2.1102	1.6898	1.7324	2.5405	2.3563	2.5143	2.8952	6.2357	4.7246	2.0038	1.6991	1.8341	3.1144	3.5717	3.5958	2.8334	2.8367	3.3948	26.9617	1	1	0.9998
2.5008	3.0143	2.6401	2.5705	2.3034	2.3947	2.0531	1.8827	3.217	3.1961	3.203	2.9478	2.6633	2.8015	3.5927	2.6291	3.1144	12.595	20.0344	73.348	1	1	0.9998
2.5918	2.9246	2.0955	2.7098	2.769	1.9229	1.6379	1.7005	3.6603	3.6057	3.1695	3.803	3.0285	2.4145	2.1935	2.5751	2.7236	4.3104	5.0721	51.6875	1	1	0.9983
2.8157	3.4491	2.1516	2.6446	3.0462	2.1857	2.0201	2.1096	3.2755	4.0419	5.2713	5.563	7.1639	3.6436	2.6136	2.7311	2.5961	4.8996	14.91	66.2652	1	1	0.9995
1.7283	1.6544	2.0135	1.7601	2.4074	1.5663	1.6979	2.0115	1.5678	1.4886	1.5313	72.2308	65.8166	46.6654	74.8098	84.6975	72.3028	59.8124	49.7485	95.4826	1	1	0.9994
1.9508	1.842	2.0144	2.0958	2.1604	1.561	1.5291	2.0135	1.6917	1.646	1.5833	2.4817	7.3621	4.9404	8.4032	9.1238	11.6843	16.2057	15.1727	36.1932	1	1	0.9997
1.982	1.8744	1.6365	2.0991	2.0303	1.896	1.7575	4.3758	1.8269	1.5409	1.5769	1.735	16.2074	27.8125	48.7381	69.1536	65.6525	55.9719	48.583	90.0073	1	1	0.9992
2.0901	2.415	2.0521	2.6618	2.4099	1.8314	1.882	2.0929	1.7608	1.7279	1.7356	2.2674	9.2329	3.3688	4.5787	8.8156	18.7743	18.5582	17.0055	4.1109	1	1	0.9993
2.8438	2.5435	2.3635	2.3672	2.5766	1.7789	1.9201	1.8807	1.7626	1.7301	1.7376	2.998	4.4727	2.9049	3.1772	3.0869	3.4756	3.5792	3.215	3.8571	1	1	0.9995
2.4113	2.4207	1.9886	2.5185	2.473	1.6911	1.7484	1.7148	1.6776	1.668	1.6695	3.5319	6.9929	4.2227	2.9734	3.4577	3.0598	2.7662	3.1315	3.0272	1	1	0.9988
3.6821	2.7611	2.7848	1.9247	2.0743	2.3579	1.7433	1.5813	1.6317	2.6069	5.9326	5.7309	5.7095	4.8991	3.5494	3.7435	2.7402	4.5936	5.2427	3.8795	1	1	0.9987
3.1949	2.4205	2.5659	1.6785	1.7385	2.1676	1.686	1.5817	1.642	2.7092	6.1555	4.2229	4.2333	8.6556	8.0899	6.8921	6.824	5.852	6.4435	5.0704	1	1	0.9994
3.3075	2.8553	2.7275	1.9067	2.0194	2.4403	1.8722	1.672	1.6931	3.0565	3.4889	3.1111	3.1384	3.7305	3.3603	4.408	4.9787	5.477	6.6971	5.2507	1	1	0.9978
2.0129	2.2666	1.8319	2.1471	2.1576	1.8183	1.7292	2.1674	1.5668	1.559	1.5669	1.9923	89.8754	90.0145	89.0873	78.3245	65.9914	56.3617	46.9699	91.769	1	1	0.998
2.1261	1.6998	1.6492	2.7406	2.2643	1.5595	1.6609	1.8379	1.5877	1.5507	1.5419	3.8931	24.6627	14.1003	13.2342	15.4761	11.3094	7.0092	6.3474	9.0822	1	1	0.9994
2.1626	1.8073	1.6803	2.8554	2.008	1.5452	1.5751	1.7051	1.5824	1.5483	1.5423	4.8919	9.2026	13.8476	24.3194	46.0471	43.6212	37.2851	29.8681	72.6508	1	1	0.9991
2.0986	1.9294	2.4204	2.2137	1.6932	1.6444	2.3692	1.6212	1.5843	1.5463	3.742	2.6295	2.4009	8.9388	6.9645	7.5934	14.2665	23.1041	22.4222	82.9636	1	1	0.9999
2.1146	2.0813	2.5649	2.2078	1.6734	1.6337	2.2767	1.5774	1.546	1.5348	3.9943	5.3736	3.5639	5.6259	5.7205	6.2856	13.8264	19.4627	25.8381	81.9456	1	1	0.9997
1.9601	1.7771	2.285	2.2128	1.7235	1.7018	2.6032	1.6003	1.5708	1.5573	3.792	3.0662	2.6687	3.2243	3.3488	3.1832	4.7456	8.4625	5.6324	38.9834	1	1	0.9999
3.5112	2.7319	2.7265	2.6853	2.4807	2.0853	2.4792	2.3839	3.0347	2.5353	2.2994	2.151	2.119	4.3379	5.9496	4.1898	3.1736	2.9994	2.9446	3.8043	1	1	1
2.6894	2.3694	2.0572	2.1293	2.4504	2.5717	2.2029	2.0785	2.2186	1.9525	1.8874	1.7509	3.956	5.4257	6.5277	4.2433	3.4283	4.4022	3.5217	15.3447	1	1	1
2.858	2.3682	2.6748	2.5834	2.4731	2.6172	1.9586	2.0325	2.2034	2.4549	2.2931	2.3094	6.2845	10.8254	10.1069	3.0655	2.5228	3.0926	4.0215	32.4038	1	1	1

Final CSV file with the Predicted Label and Accuracy Score as out.csv file that the user will finally download for later use.

CHAPTER-5

CONCLUSION AND FUTURE WORK

5.1.1 Conclusion

Parkinson’s disease is the second most dangerous neurodegenerative disease which has no cure till now and to make it reduce prediction is important. In this project, we have used three various prediction models to predict Parkinson's disease which are Machine Learning Techniques i.e.LGBM. The dataset is trained using these models and we also compared these different models built using different methods and identified the best model that fits.

The aim is to use various evaluation metrics such as Accuracy, Precision, Recall, Specificity, F1-score, LR+, LR- and Youden score that produce the predicted disease efficiently. We have used the Speech dataset that contains voice features of the patients which is available in the Kaggle website. The dataset consists of more than 700 features and 750 patient details. The models are built using the five best features which were identified by feature selection. From these results, Naïve Bayes stands out from the other two machine learning algorithms with an accuracy of 81%. This system we designed can make the predictions of Parkinson's disease.

5.1.2 Future Work

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

References

[1] A. Ozcivit, “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.

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

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

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

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

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

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

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

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

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

[11] M. A. E. Van Stiphout, J. Marinus, J. J. Van Hilten, F. Lobbezoo, and C. De Baat, "Oral health of Parkinson's disease patients: a case-control study" Parkinson's disease, vol-2018, Article ID 9315285, 8 pages, 2018.

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

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

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

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

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

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

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

[19] Shahid, A.H., Singh, M.P. A deep learning approach for prediction of Parkinson's disease progression, https://doi.org/10.1007/s13534-020-00156-7, Biomed. Eng. Lett. 10, 227–239, 2020.

[20] Shubham Bind, et al, "A survey of machine learning based approaches for Parkinson disease prediction" International Journal of Computer Science and Information Technologies vol-6, Issue 2, pp. 1648- 1655, 2015.

[21] Siva Sankara Reddy Donthi Reddy and Udaya Kumar Ramanadham "Prediction of Parkinson's Disease at Early Stage using Big Data Analytics"ISSN: 2249 – 8958, Volume- 9

Issue-4, April 2020

[22] Sriram, T. V., et al. “Intelligent Parkinson Disease Prediction Using Machine Learning Algorithms” International Journal of Engineering and InnovativeTechnology, vol-3, Issue 3, September 2013.

[23] 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.

[24] 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.