

Parkinson analysis using “MACHINE LEARNING

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

By

K Bhanu Siva Kumar37110352

Abhishikth Konchada37110355

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119, TAMILNADU

MARCH 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)



Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in

DEPARTEMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of K Bhanu Siva Kumar(37110352) who carried out the project entitled “**Parkinson analysis using “MACHINE LEARNING**”, “under my supervision from November 2020 to Mrch 2021.

Internal Guide

Dr.G.Meeragandhi, M.E., Ph.D.

HEAD OF DEPARTEMENT

Dr. S. VIGNESHWARI, M.E., Ph.D.

Dr. L. Lakshmanan, M.E., Ph.D.

Submitted for Viva Voice Examination held on _____

Internal Examiner

External Examiner

DECLARATION

K Bhanu Siva Kumar(37110352) who carried out the project entitled “Parkinson analysis using MACHINE LEARNING, under the guidance of **Dr.G.Meeragandhi,M.E,Ph.D.** at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

NAME OF THE CANDIDATE

ACKNOWLEDGEMENT

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

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

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

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

Paper acceptance – IEEE conf..

TABLE OF CONTENTS:

Chapter No.	Title.	Page No.
	Abstract	
	LIST OF FIGURES	
1.	INTRODUCTION	1
1.1	Introduction	1-2
1.2	Existing system	3
1.3	Proposed system	3-4
1.4	System requirements	4
2.	LITERATURE SURVEY	
2.1	Machine learning	5
2.2	Some machine learning methods	5-6
2.3	Applications of machine learning	6
2.4	Prevalence of parkinsons disease	6-7
2.5	Importance of machine learning to health care	7
2.6	Implementation of machine learning using python	8-10
3	SYSTEM ANALYSIS	
3.1	Scope of the project	11
3.2	Analysis	11-12
3.3	Data processing	13
3.4	Removing of outliers	14-16
3.5	Feature selection	15-17
3.6	classification	16-17
3.7	Confusion matrix	20-22
4.	APPENDIX	
	A. Coding	23-25
	B. Result analysis	26-30
	C. Screenshots	31-33
5.	Conclusions, future work	34

LIST OF FIGURES

FIGURE NO.	FIGURE NAME.	PAGE NO.
1.1	Proposed system	4
2.1	Prevalence of parkinson's data	7
3.1	Dataset	12
3.2	Dataset and description	13
3.3	Data pre-processing	14
3.5	Dataset	15
3.6	Representation of outliers	15
3.7	Before removing of outliers	16
3.8	Correlation for parkinson's disease dataset	17
3.9	XG boost	18
3.10	Extra tree classifier	19
3.11	Gradient boosting	19
3.12	Random forest	20
3.13	Confusion matrix	20
4.1	Before applying correlation	26
4.2	After applying correlation	26
4.3	Comparison of algorithms after removal of outliers	27
4.4	Comparison of algorithms after pca	27
4.5	Comparison of algorithms with correlation	28
4.6	Comparison of algorithms after correlation and pca	28
4.7	Comparison of algorithms after applying k-fold cross validation	29
4.8	Comparison of algorithms after applying information gain	29
4.9	Comparison of algorithms after applying smote	30
5.1	Home page	31
5.2	Input 1	31
5.3	Has disease	32
5.4	Input 2	32
5.5	Has No Disease	33

ABSTRACT

Parkinson's disease is a progressive disease of the nervous system marked by tremor, muscular rigidity, and slow, imprecise movement, chiefly affecting middle-aged and elderly people. It is associated with degeneration of the basal ganglia of the brain and a deficiency of the neurotransmitter dopamine. It slowly reduces the nerve cells. It can come on so slowly that you don't even notice it at first. But over time, what starts as a little shakiness in your hand can have an impact on how you walk, talk, sleep, and think. To overcome the issue, our project aims to detect the Parkinson's disease using machine learning techniques with Parkinson's dataset. The main objective of our project is to use classification algorithms to identify the patients who are suffering from Parkinson's disease.

INTRODUCTION

Parkinson's disease is a brain disorder that gets worse over time. It causes nerve cells in a part of the brain called the substantia nigra to die. This part of the brain is important for controlling movement. That's why people with Parkinson's often shake or show other abnormal movements.

Treatments can help with symptoms, but there is no way to slow or reverse the condition.

No one knows exactly why a person gets Parkinson's. It's probably due to a mix of things, including genes and exposure to certain toxins. There's usually no way to predict who will to run in families. Most of the time, it seems to happen randomly.

Both men and women get Parkinson's disease. It's 1.5 times more common in men. It's also more common in older people. Only about 4 out of every 100 cases happen in people under age 50. Each year, about 60,000 people in the U.S. find out they have Parkinson's. About 1 million people in the U.S. and 10 million around the world have this condition.

Parkinson's signs and symptoms may include:

- Tremors or shaking of hands, arms, legs, jaw, or head
- Stiffness of arms, legs, and trunk
- Slowed movement
- Trouble with balance and coordination
- Depression or other emotional changes
- Trouble chewing, swallowing, or speaking
- Trouble sleeping
- Constipation

Parkinson's is a progressive disease. That means that your symptoms usually get worse over time. The symptoms of Parkinson's also vary a lot from one person to the next. How quickly it worsens and how severe it gets can vary a lot, too. Early symptoms may be easy to ignore or dismiss. They might start on one side of your body, showing up on the other side only later.

At first, it can be hard for doctors to tell if you have Parkinson's. That's in part because symptoms vary so much. Other disorders also can look similar. There's no single test for it. Your doctor might order imaging tests to rule out other conditions.

Parkinson's affects nerve cells in your brain that make a chemical called dopamine. As a result, levels of the chemical fall. Doctors usually start treatment with levodopa (L-dopa). Your brain turns it into dopamine. But it can make you sick to your stomach, so you'll probably take it with another medicine called carbidopa to control these side effects.

The doctor may also give you something to treat the tremors that come with Parkinson's. These medications are called anticholinergics, and they put the brakes on a chemical in your brain that controls movement. Your doctor might give you one of these, alone or with another drug:

- **Dopamine agonists:** They act like dopamine but don't raise levels of it in your brain. You can take them with any drug that has levodopa. You might try pramipexole (Mirapex) or ropinirole (Requip).
- **COMT Inhibitors:** They help levodopa last longer. You might get entacapone (Comtan) or tolcapone (Tasmar).
- **MAO-B inhibitors:** These stop your brain from breaking down levodopa. You could get selegiline (Eldepryl, Zelapar) or rasagiline (Azilect).

If medicine doesn't work well enough, your doctor may suggest deep brain stimulation (DBS). In DBS, your doctor implants electrodes deep in the brain. A device connected to them delivers electrical pulses. Those pulses can help control the tremors caused by Parkinson's. In the past, doctors sometimes used other operations to damage the brain in ways to help with movement symptoms. But they rarely use those surgeries now.

Physical therapy can teach you exercises to improve strength and balance and help you stay independent. Occupational therapy shows you new ways to manage daily tasks. Speech therapy can help with slurred or unclear speech.

Along with your regular medical care, these approaches may help with mood, energy, and how you feel day to day:

- Massage
- Yoga

Eating healthy foods can help you feel better. It may also help with certain Parkinson's symptoms, such as constipation. Getting regular exercise will also boost strength, flexibility, and balance. Ask your doctor to recommend a physical therapist or exercise program.

1.2 Existing System

Nowadays, decisions are made by doctors based on their experience and knowledge. This practice may lead to errors, consume a lot of time and excessive medical costs which affects the quality of service provided to patients.

Disadvantages:

- a. Doesn't generate accurate and efficient results
- b. Computation time is very high
- c. Difficulty in maintenance of patient records
- d. Lacking of accuracy may result in lack of efficient further treatment

1.3 Proposed System

Parkinson's disease is a progressive disease of the nervous system marked by tremor, muscular rigidity, and slow, imprecise movement, chiefly affecting middle-aged and elderly people. It is associated with degeneration of the basal ganglia of the brain and a deficiency of the neurotransmitter dopamine. It slowly reduces the nerve cells.

It can come on so slowly

that you don't even notice it at first. But over time, what starts as a little shakiness in your hand can have an impact on how you walk, talk, sleep, and think. To overcome the issue, our project aims to detect the Parkinson's disease using machine learning techniques with Parkinson's dataset. The main objective of our project is to use classification algorithms to detect the patients who are suffering from Parkinson's disease.

Advantages:

- a. Generates accurate and efficient results
- b. Computation time is greatly reduced
- c. Reduces manual work
- d. Efficient further treatment
- e. Automated prediction

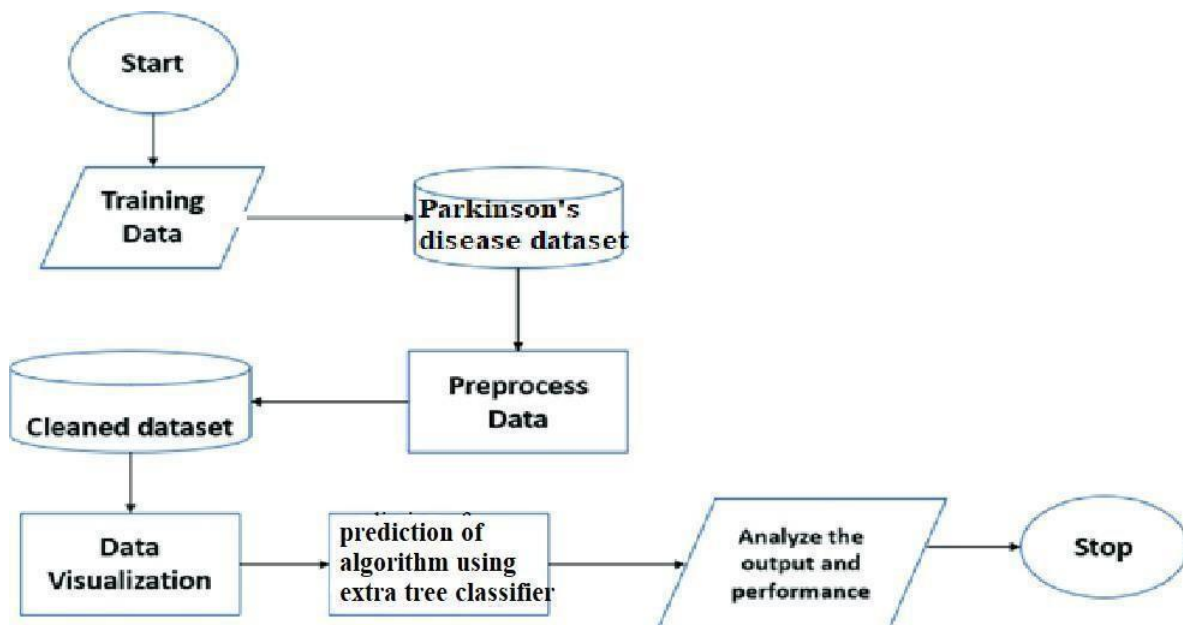


Fig:1.1 Proposed System

1.4. System Requirements

1.4.1 Hardware Requirements:

- System Type : intel®core™i7-7500UCPU@2.70gh
- Cache memory : 4MB(Megabyte)
- RAM : 4 gigabyte (GB)

1.4.2 Software Requirements:

- Operating System : Windows 10 Home, 64bit Operating System
- Coding Language : Python
- Python distribution : Anaconda, JupyterLab, spyder

2. LITERATURE SURVEY

2.1 Machine Learning

Machine learning is one of the applications of artificial intelligence (AI) that provides computers, the ability to learn automatically and improve from experience instead of explicitly programmed. It focuses on developing computer programs that can access data and use it to learn from themselves. The main aim is to allow computers to learn automatically without human intervention and also adjust actions accordingly.

2.2 Some machine learning methods

Machine learning algorithms are often categorized as supervised and unsupervised.

. ***Supervised machine learning algorithms*** can apply what has been learned in the past to new data using labelled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

┌ In contrast, ***unsupervised machine learning algorithms*** are used when the information used to train is neither classified nor labelled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabelled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabelled data.

. ***Semi-supervised machine learning algorithms*** fall somewhere in between supervised and unsupervised learning, since they use both labelled and unlabelled data for training – typically a small amount of labelled data and a large amount of

unlabelled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labelled data requires skilled and relevant resources

- in order to train it / learn from it. Otherwise, acquiring unlabelled data generally doesn't require additional resources.
- **Reinforcement machine learning algorithms** is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best. This is known as the reinforcement signal.

2.3 Applications of machine learning

- Virtual Personal Assistants □
- Predictions while Commuting □
- Videos Surveillance □
- Social Media Services □
- Email Spam and Malware Filtering □
- Online Customer Support □
- Search Engine Result Refining □
- Product Recommendations
- Online Fraud Detection □

2.4 Prevalence of Parkinson's disease

In a review by de Lau and Breteler in 2006, it was reported that an estimated 10 million people in the world (i.e., approximately 0.3% of the world population) and 1% of those above 60 years are found to be affected with PD. There are very few population-based studies determining the exact incidence and prevalence of PD in India. In a door-to-door survey done in Bangalore district in South Karnataka in India in 2004, the prevalence rate of Parkinsonism was found to be 33 per 100,000 (crude prevalence) and 76 per 100,000 (age adjusted). Rural population had a higher prevalence compared to the urban population (41 vs 14). But it was less compared to

other highly prevalent neurological disorders such as headache, epilepsy, stroke, and mental retardation. From a survey in Kolkata in 2006, the prevalence of Parkinsonism was found to be 45.82 per 100,000. In the state of Kashmir, the prevalence was 14.1 per 100,000, while

the age adjusted prevalence was 134 per 100,000. A survey, done in Parsi community in Mumbai, a small stable community, showed a prevalence of 192 per 100,000, which was higher compared to rest of the population. In a surveillance in old age homes in a Bangalore, there was very high prevalence of 17.6% (109/612 residents) of Parkinsonism. This may be due to unawareness of this disorder among the general population who do not avail the medical facilities at the appropriate time.

The prevalence rate in Europe is 1.8 per 100 in population above 65 years. As the prevalence of PD in non-white population is lower, the normal human brains from India and United Kingdom were compared, to determine the cause. The neurons which were melanized in substantia nigra (pars compacta), as the age increases, were counted and compared. Indian brains, though had low number of melanized neurons, the decrease in these number of neurons with age was much less compared to the whites. But a similar study comparing Nigerian and British brains did not show any difference between these two ethnic groups. In one clinical study comparing the prevalence of PD in different ethnic groups in India, Anglo- Indians, who had admixed ethnicity with Indian population had nearly five times lesser prevalence of PD compared to general Indian population. In summary, there may be several other factors causing the difference in prevalence in varied populations and this need to be explored.

In most studies, it has been found that women have lesser prevalence of PD. A study at All India Institute of Medical Sciences (AIIMS) by Yadav *et al.*, showed that women with higher number and longer duration of pregnancies had lesser prevalence of PD. This may indicate a protective role of estrogen.

Most of the prevalence studies have used different methods to screen the population. Tanner, devised one of the widely used validated screening questionnaires for PD. Sarangmath, from National Institute of Mental Health and Neurosciences (NIMHANS) have modified the screening questionnaire by Tanner *et al.*, to cater to the varying literacy levels and different languages used in India. A few changes were made from the prior standard screening questionnaire and this was found to have good sensitivity even when administered by a nonmedical assistant.

The epidemiology of PD in India is different compared to populations from different countries and also among different ethnicities in India. The prevalence studies in India are done among small population from specific areas probably with a particular type of cultural background. So, to generalize it to the rest of the population in a country

where there is a huge cultural, ethnic, and social difference may not be correct. Hence, large multicenter studies for evaluating prevalence at different parts of the country are necessary.

Population-based studies	State	Crude prevalence	Age-specific prevalence (>50 years)
Bharucha <i>et al.</i> , 1988 ^[7]	Mumbai-Western India		148/100,000
Razdan <i>et al.</i> , 1994 ^[8]	Kashmir-Northern	14.2/100,000	134/100,000
Gourie-Devi <i>et al.</i> , 2004 ^[4]	Karnataka-Southern	33/100,000	76/100,000
Das <i>et al.</i> , 2006 ^[9]	Kolkata-Eastern	45.82/100,000	

Fig:2.1 Prevalence of Parkinson's disease

2.5 Importance of machine learning in healthcare

The importance of machine learning in healthcare is increasing because of its ability to process huge datasets efficiently beyond the range of human capability, and then dependably convert analysis of that data into clinical insights that assist physicians in planning and providing care, which ultimately leads to better outcomes, reduces the costs of care, and increases patient's satisfaction. Using these types of advanced analytics, we can provide better information to doctors at the point of patient care.

2.6 Implementation of machine learning using Python

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

- web development (server-side),
- software development,
- mathematics, □ system scripting.

The most recent major version of Python is Python 3. However, Python 2, although not being updated with anything other than security updates, is still quite popular.

It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse, Anaconda which are particularly useful when managing larger collections of Python files.

Python was designed for its readability. Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly- brackets for this purpose.

In the older days, people used to perform Machine Learning tasks manually by coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reason is its vast collection of libraries. Python libraries that used in Machine Learning are:

- Numpy
- Scipy
- Scikit-learn
- Theano
- TensorFlow
- Keras
- PyTorch
- Pandas
- Matplotlib

NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical

functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number

capabilities. High-end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

SciPy is a very popular library among Machine Learning enthusiasts as it contains different modules for optimization, linear algebra, integration and statistics. There is a difference between the SciPy library and the SciPy stack. The SciPy is one of the core packages that make up the SciPy stack. SciPy is also very useful for image manipulation.

Skikit-learn is one of the most popular Machine Learning libraries for classical Machine

Learning algorithms. It is built on top of two basic Python libraries, NumPy and SciPy. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikit-learn can also be used for data-mining and data-analysis, which makes it a great tool who is starting out with Machine Learning.

Theano is a popular python library that is used to define, evaluate and optimize mathematical expressions involving multi-dimensional arrays in an efficient manner. It is achieved by optimizing the utilization of CPU and GPU. It is extensively used for unit-testing and self-verification to detect and diagnose different types of errors. Theano is a very powerful library that has been used in large-scale computationally intensive scientific projects for a long time but is simple and approachable enough to be used by individuals for their own projects.

TensorFlow is a very popular open-source library for high performance numerical computation developed by the Google Brain team in Google. As the name suggests, Tensorflow is a framework that involves defining and running computations involving tensors. It can train and run deep neural networks that can be used to develop several AI applications. TensorFlow is widely used in the field of deep learning research and application.

Keras is a very popular Machine Learning library for Python. It is a high-level neural networks API capable of running on top of TensorFlow, CNTK, or Theano. It can run seamlessly on both CPU and GPU. Keras makes it really for ML beginners to build and design a Neural Network. One of the best things about Keras is that it

allows for easy and fast prototyping.

PyTorch is a popular open-source Machine Learning library for Python based on Torch, which is an open-source Machine Learning library which is implemented in C with a wrapper in Lua. It has an extensive choice of tools and libraries that supports

on Computer Vision, Natural Language Processing (NLP) and many more ML programs. It allows developers to perform computations on Tensors with GPU acceleration and also helps in creating computational graphs.

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides

high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for grouping, combining and filtering data.

Matplotlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, histogram, error charts, bar charts, etc.

Some of the previous papers are as follows:

Shubham Jha used many approaches for detecting parkinson's disease by means of feature sets and Feature extraction using a hybrid of k-means and SVM, LR Neural Network and Decision tree. Highest accuracy is 100% for support vector machine

Adrian used classical computer vision techniques like *Histogram of Oriented Gradients (HOG)* for quantifying the features of the images and used them to train a *Random Forest Classifier*. He got an accuracy of **83.33%**.

3. SYSTEM ANALYSIS

3.1 Scope of the project

This project mainly focuses on the how the Parkinson's disease can be predicted by using some related information. It mainly depends on the symptoms. The scope of this site is the user can able to calculate the risk of Parkinson's disease.

3.2 Analysis

The dataset was created by Max Little of the University of Oxford, in collaboration with

National Centre for Voice and Speech, Denver, Colorado, who recorded the speech signals. The original study published the feature extraction methods for general voice disorders. The dataset contains 24 attributes which are used to predict Parkinson's disease the such as

name - ASCII subject name and recording number

MDVP:Fo(Hz) - Average vocal fundamental frequency

MDVP:Fhi(Hz) - Maximum vocal fundamental frequency

MDVP:Flo(Hz) - Minimum vocal fundamental frequency

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

MDVP:Shimmer,MDVP:Shimmer(dB),Shimmer:APQ3,Shimmer:APQ5,MDVP:APQ,Shimmer:DDA - Several measures of variation in amplitude

NHR,HNR - Two measures of ratio of noise to tonal components in the voice status - Health status of the subject (one) - Parkinson's, (zero) - healthy

RPDE,D2 - Two nonlinear dynamical complexity measures DFA - Signal fractal scaling exponent

name	MDVP:F0	MDVP:F1	MDVP:F0	MDVP:J1tt	MDVP:J1tt	MDVP:RAI	MDVP:PP1	Jitter:C	MDVP:Shi	MDV:Shimmer	Shimmer:	MDVP:AP1	Shimmer:	NHR	HNR	stat	RPDE	DFA	spread1	spread2	D2	PPE	
phon_R01	119.992	157.302	74.997	0.00784	0.00007	0.0037	0.00554	0.011	0.04374	0.43	0.02182	0.0313	0.02971	0.06545	0.0221	21	1	0.415	0.8153	-4.813	0.26648	2.3	0.2847
phon_R01	122.4	148.65	113.819	0.00968	0.00008	0.00465	0.00696	0.014	0.06134	0.63	0.03134	0.04518	0.04368	0.09403	0.0193	19.1	1	0.458	0.8195	-4.0752	0.33559	2.49	0.3687
phon_R01	116.682	131.111	111.555	0.0105	0.00009	0.00544	0.00781	0.016	0.05233	0.48	0.02757	0.03858	0.0359	0.0827	0.0131	20.7	1	0.43	0.8253	-4.4432	0.31117	2.34	0.3326
phon_R01	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.015	0.05492	0.52	0.02924	0.04005	0.03772	0.08771	0.0135	20.6	1	0.435	0.8192	-4.1175	0.33415	2.41	0.369
phon_R01	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.02	0.06425	0.58	0.0349	0.04825	0.04465	0.1047	0.0177	19.6	1	0.417	0.8235	-3.7478	0.23451	2.33	0.4103
phon_R01	120.552	131.162	113.787	0.00968	0.00008	0.00463	0.0075	0.014	0.04701	0.46	0.02328	0.03526	0.03243	0.06985	0.0122	21.4	1	0.416	0.8251	-4.2429	0.29911	2.19	0.3578
phon_R01	120.267	137.244	114.82	0.00333	0.00003	0.00155	0.00202	0.005	0.01608	0.14	0.00779	0.00937	0.01351	0.02337	0.0061	24.9	1	0.596	0.7641	-5.6343	0.25768	1.85	0.2118
phon_R01	107.332	113.84	104.315	0.0029	0.00003	0.00144	0.00182	0.004	0.01567	0.13	0.00829	0.00946	0.01256	0.02487	0.0034	26.9	1	0.637	0.7633	-6.1676	0.18372	2.06	0.1638
phon_R01	95.73	132.068	91.754	0.00551	0.00006	0.00293	0.00332	0.009	0.02093	0.19	0.01073	0.01277	0.01717	0.03218	0.0107	21.8	1	0.616	0.7736	-5.4987	0.32777	2.32	0.2316
phon_R01	95.056	120.103	91.226	0.00532	0.00006	0.00268	0.00332	0.008	0.02838	0.26	0.01441	0.01725	0.02444	0.04324	0.0102	21.9	1	0.547	0.7985	-5.0119	0.326	2.43	0.2714
phon_R01	88.333	112.24	84.072	0.00505	0.00006	0.00254	0.0033	0.008	0.02143	0.2	0.01079	0.01342	0.01892	0.03237	0.0117	21.1	1	0.611	0.7762	-5.2498	0.391	2.41	0.2497
phon_R01	91.904	115.871	86.292	0.0054	0.00006	0.00281	0.00336	0.008	0.02752	0.25	0.01424	0.01641	0.02214	0.04272	0.0114	21.4	1	0.583	0.7925	-4.9602	0.36357	2.64	0.2759
phon_R01	136.926	159.866	131.276	0.00293	0.00002	0.00118	0.00153	0.004	0.01259	0.11	0.00656	0.00717	0.0114	0.01968	0.0058	25.7	1	0.461	0.6468	-6.5471	0.15281	2.04	0.1385
phon_R01	139.173	179.139	76.556	0.0039	0.00003	0.00165	0.00208	0.005	0.01642	0.15	0.00728	0.00932	0.01797	0.02184	0.0104	24.9	1	0.43	0.6658	-5.6602	0.25499	2.52	0.1999
phon_R01	152.845	163.305	75.836	0.00294	0.00002	0.00121	0.00149	0.004	0.01828	0.16	0.01064	0.00972	0.01246	0.03191	0.0061	24.9	1	0.475	0.654	-6.105			

Fig:3.1 Dataset

Name	Subject Name
MDVP:FO (Hz)	Average vocal fundamental frequency
MDVP:Fhi (Hz)	Maximum vocal fundamental frequency
MDVP:Flo (Hz)	Minimum vocal fundamental frequency
MDVP:Jitter (%) MDVP:Jitter (Abs), MDVP:RAP,MDVP:PPQ, Jitter:DDP	variation in fundamental frequency
MDVP:Shimmer, MDVP:Shimmer (dB), Shimmer:APQ3, Shimmer:APQ5, MDVP:APQ, Shimmer:DDA	variation in amplitude
NHR,HNR	ratio of noise to tonal components in the voice
RPDE,D2	nonlinear dynamical complexity measures
DFA	Signal fractal scaling exponent
Spread1,spread2,PP	nonlinear measures of fundamental frequency variation

Figure : 3.2 Dataset Description

3.3 Data Pre-processing

Before feeding data to an algorithm we have to apply transformations to our data which is referred as pre-processing. By performing pre-processing the raw data which is not feasible for analysis is converted into clean data. In-order to achieve better results using a model in Machine Learning, data format has to be in a proper manner. The data should be in a particular format for different algorithms. For example, if we consider Random Forest algorithm it does not support null values. So that those null values have to be managed using raw data.

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

Pre-processing is the first step while creating the machine learning model. It is the process of converting raw dataset into cleaned dataset. Raw data contains noise, missing values, duplicate values which is not suitable for machine learning model. So, pre-processing is required for cleaning the data and making it suitable for machine learning model.

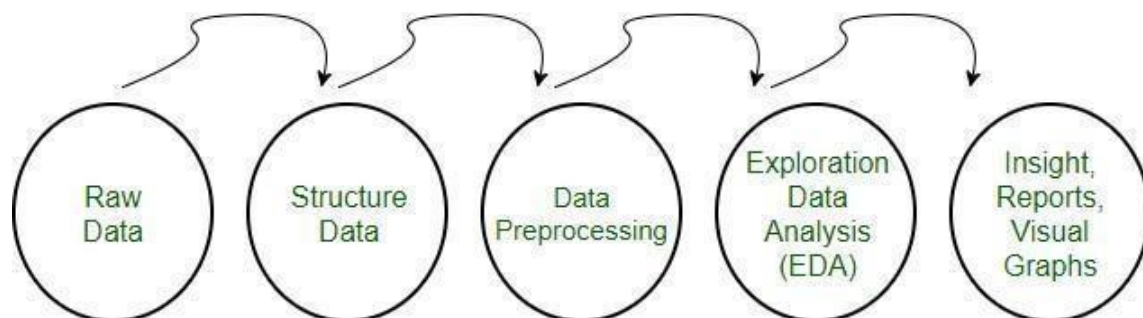


Fig:3.3 Data Pre-processing

3.3.1 Missing values

Filling missing values is one of the pre-processing techniques. The missing values in the dataset is represented as '?' but it a non-standard missing value and it has to be converted into a standard missing value NaN. So that pandas can detect the missing values. In Parkinson's disease dataset there are no missing values.

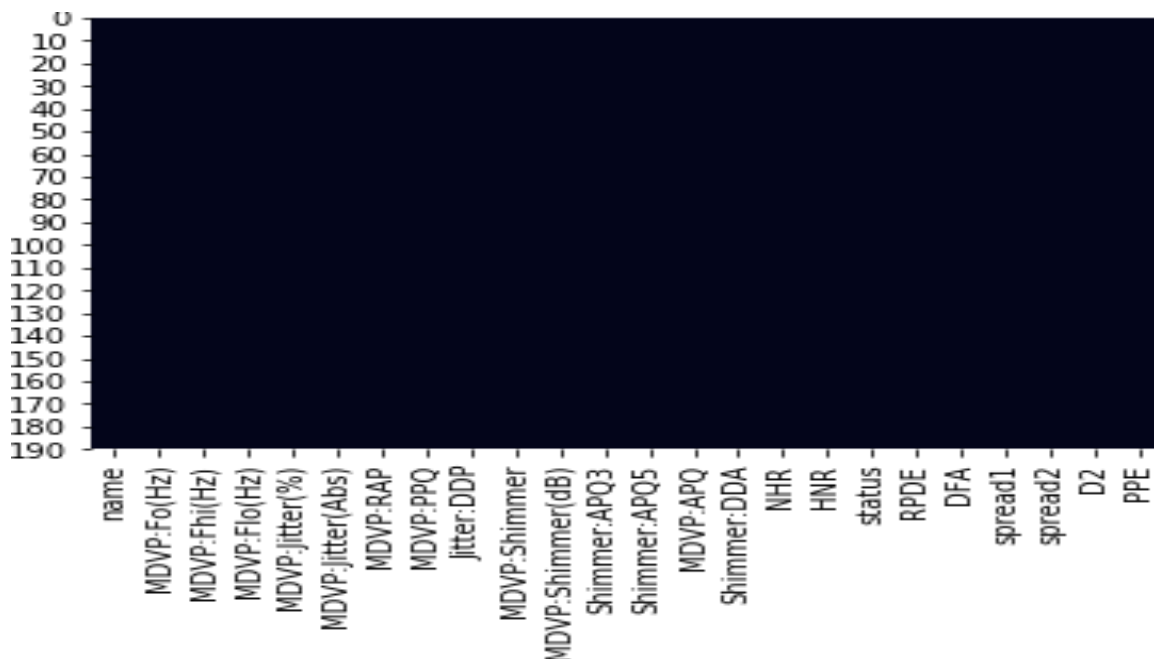


Fig :3.4 Dataset

3.4 Removing of outliers:

An **outlier** is an object that deviates significantly from the rest of the objects. They can be caused by measurement or execution error. The analysis of outlier data is referred to as outlier analysis or outlier mining.

Why outlier analysis?

Most data mining methods discard outliers' noise or exceptions, however, in some applications such as fraud detection, the rare events can be more interesting than the more regularly occurring one and hence, the outlier analysis becomes important in such case.

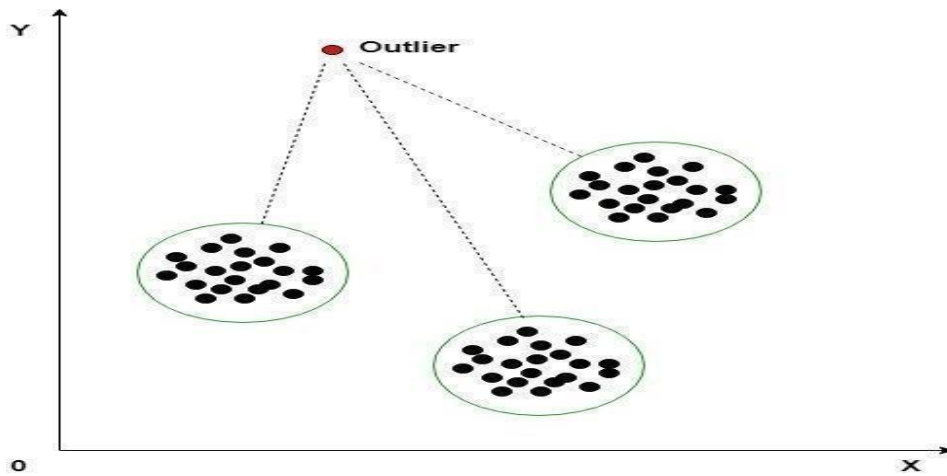


Fig: 3.5: Representation of Outliers

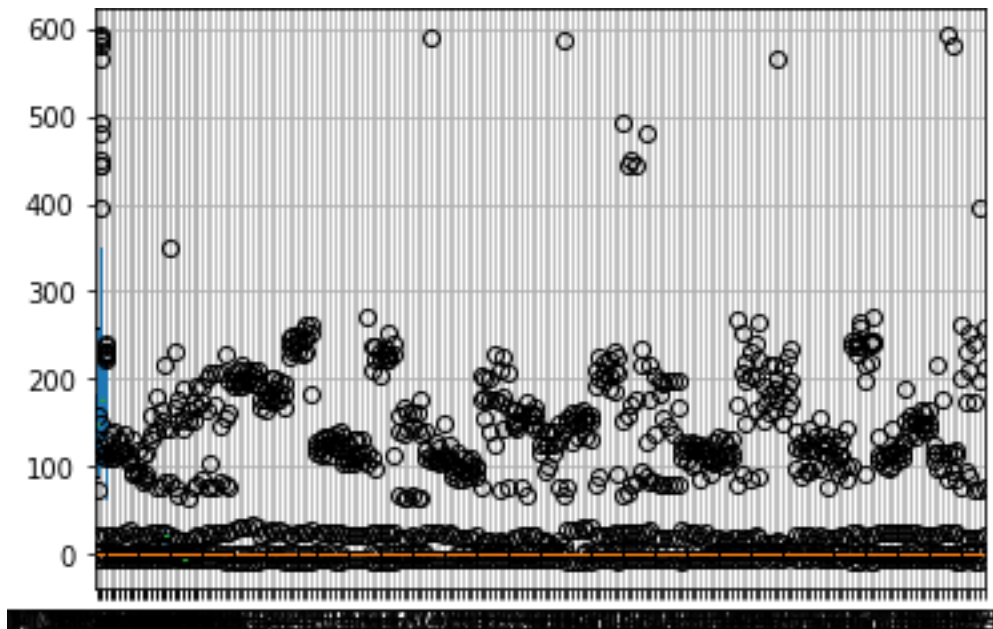


Fig:3.6 Before removing of outliers

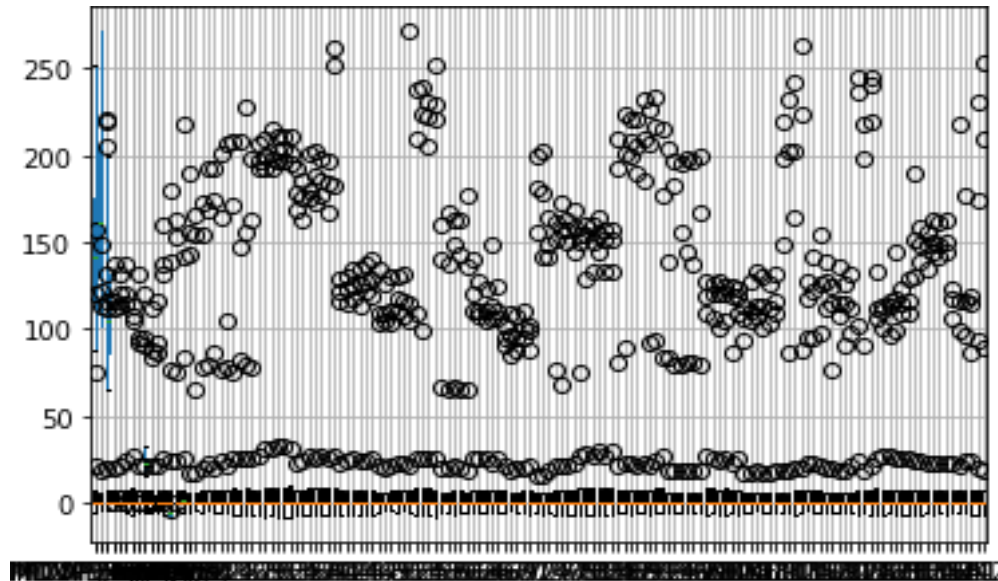


Fig:3.7 After removing of outliers

In the K-Means clustering technique, each cluster has a mean value. Objects belong to the cluster whose mean value is closest to it. In order to identify the Outlier, firstly we need to initialize the threshold value such that any distance of any data point greater than it from its nearest cluster identifies it as an outlier for our purpose. Then we need to find the distance of the test data to each cluster mean. Now, if the distance between the test data and the closest cluster to it is greater than the threshold value then we will classify the test data as an outlier.

3.5 FEATURE SELECTION: Feature Selection is process of reducing the number of input variables when developing a predictive model. To reduce the computational cost of modelling and in some of the cases to increase or improve the performance of the model it is desirable to reduce the number of input variables. Feature Selection method used in this project is correlation.

3.5.1 CORRELATION: Correlation describes the linear relationship between the two continuous variables. Correlation is used when there is no identified response variable. It is a statistical measure that indicates the extent to which two or more variables fluctuate together. A positive correlation indicates extent to which those variables increase or decrease in parallel. A negative correlation indicates extent to which one variable increases as the other decreases.

improve speed and model performance. It has recently been dominating in applied

machine learning. XGBoost models majorly dominate in many Kaggle Competitions. In this algorithm, decision trees are created in sequential form. Weights

play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results.

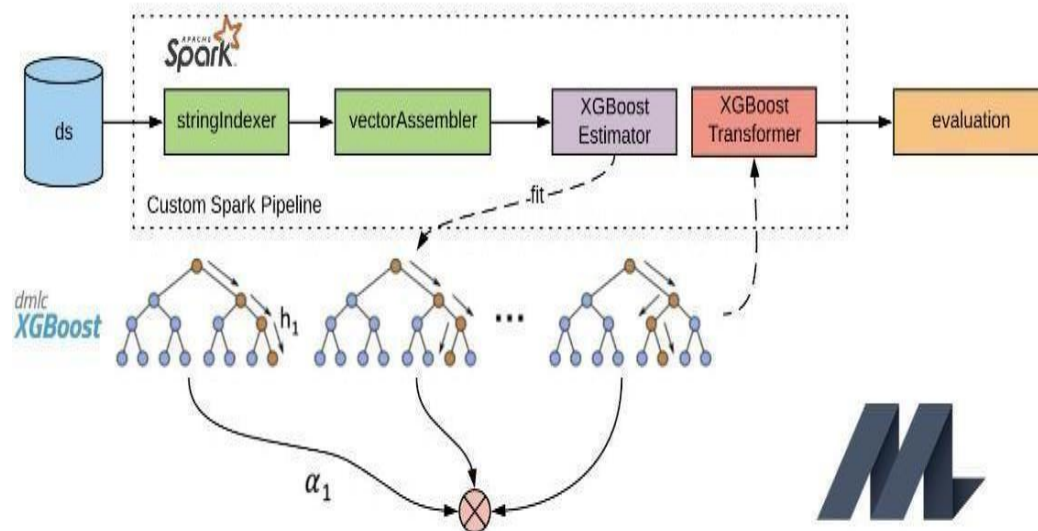


Fig: 3.9 XG Boost

1. **Extra Trees classifier: (*Extremely Randomized Trees Classifier*)** is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a “forest” to output it’s classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest. Each Decision Tree in the Extra Trees Forest is constructed from the original training sample. Then, at each test node, each tree is provided with a random sample of k features from the feature-set from which each decision tree must select the best feature to split the data based on some mathematical criteria (typically the Gini Index).

This random sample of features leads to the creation of multiple de-correlated decision trees.

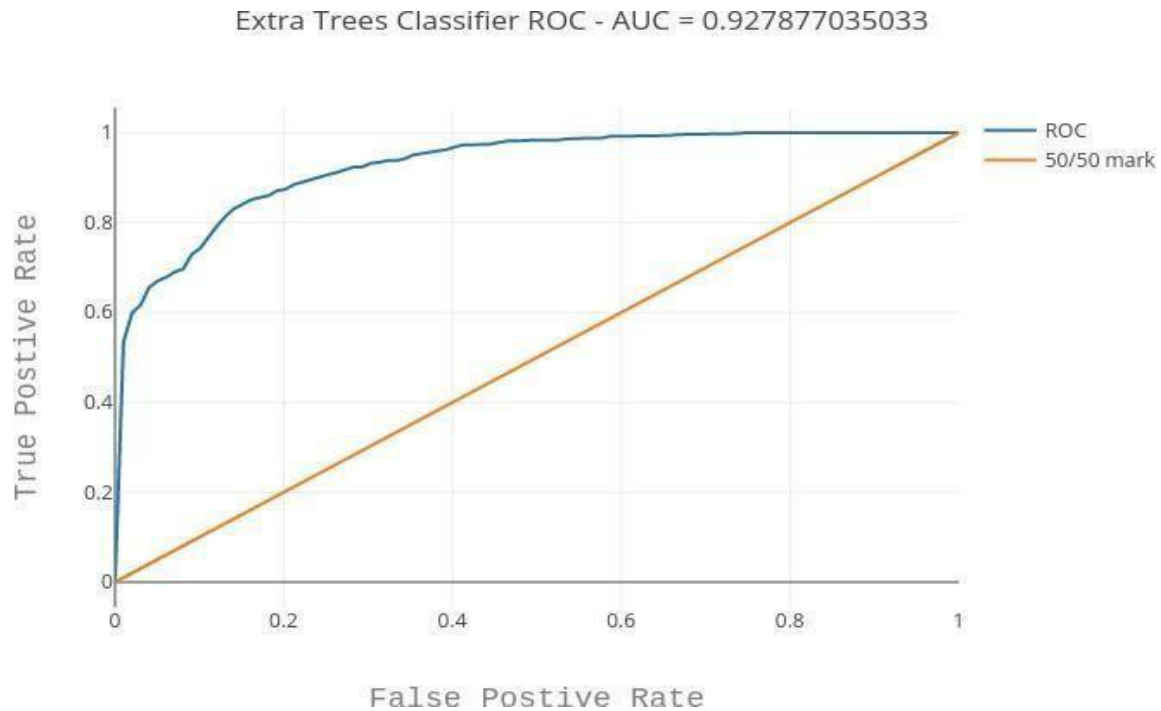


Fig: 3.10 Extra Trees Classifier

2. **Gradient Boosting classifier:** Gradient Boosting classifier is an ensemble modelling technique which attempts to build a strong classifier from the number of weak classifiers. It is done building a model by using weak models in series. Firstly, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models are added.

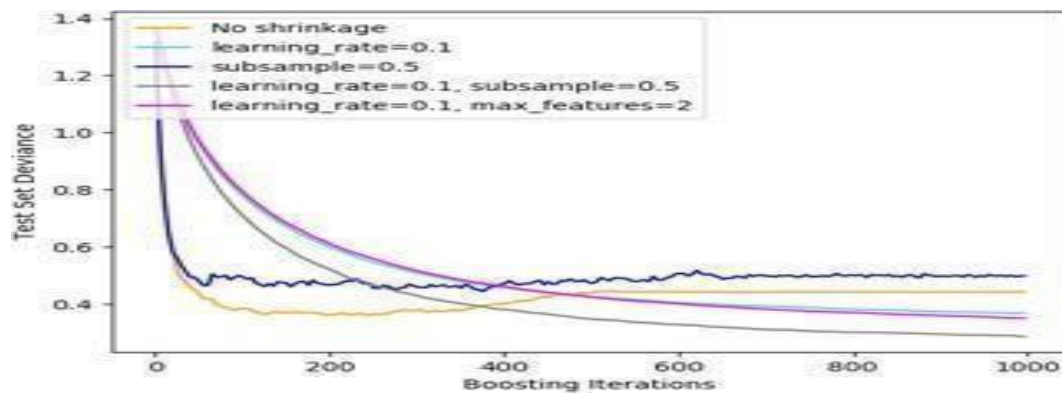


Fig: 3.11 Gradient Boosting

3. **XGBRF Classifier** : XGBoost is normally used to train gradient-boosted decision trees and other gradient boosted models. Random forests use the same model representation and inference, as gradient-boosted decision trees, but a different training algorithm. One can use XGBoost to train a standalone random forest or use random forest as a base model for gradient boosting. Here we focus on training standalone random forest.

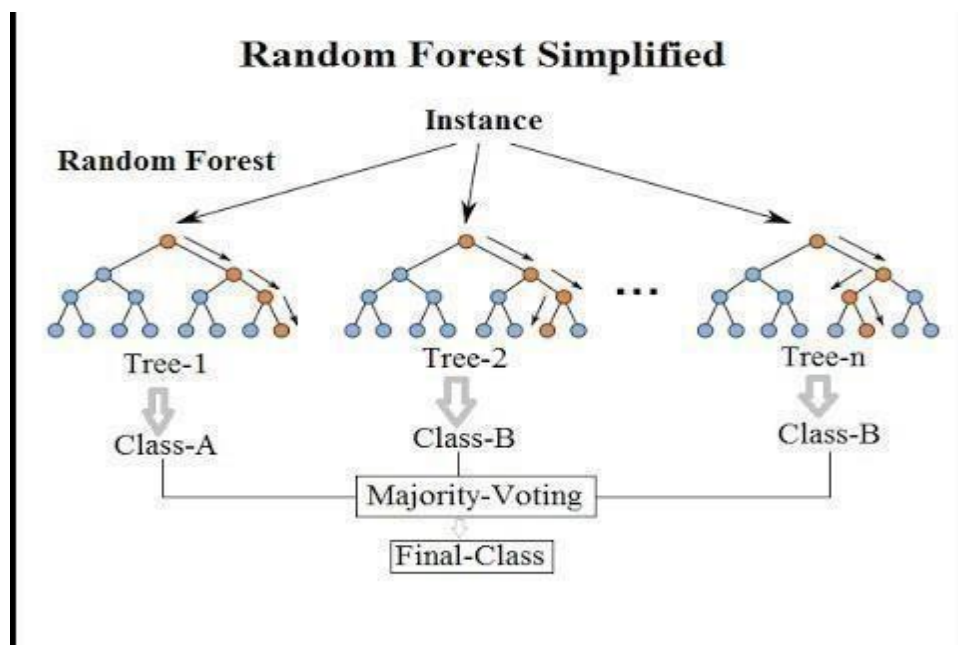


Fig: 3.12 Random Forest

3.7 CONFUSION MATRIX

A confusion matrix is a table that is often used to **describe the performance of a classification model** (or "classifier") on a set of test data for which the true values are known.

		Predicted class	
		<i>P</i>	<i>N</i>
Actual Class	<i>P</i>	True Positives (TP)	False Negatives (FN)
	<i>N</i>	False Positives (FP)	True Negatives (TN)

Fig:3.13 Confusion Matrix

A true positive (tp) is a result where the model predicts the positive class correctly. Similarly, a true negative (tn) is an outcome where the model correctly predicts the negative class.

A false positive (fp) is an outcome where the model incorrectly predicts the positive class. And a false negative (fn) is an outcome where the model incorrectly predicts the negative class

Sensitivity or Recall or hit rate or true positive rate (TPR)

It is the proportion of individuals who actually have the disease were identified as having the disease.

$$\text{TPR} = \text{tp} / (\text{tp} + \text{fn})$$

Specificity, selectivity or true negative rate (TNR)

It is the proportion of individuals who actually do not have the disease were identified as not having the disease.

$$\text{TNR} = \text{tn} / (\text{tn} + \text{fp}) = 1 - \text{FPR}$$

Precision or positive predictive value (PPV)

If the test result is positive what is the probability that the patient actually has the disease.

$$\text{PPV} = \text{tp} / (\text{tp} + \text{fp})$$

Negative predictive value (NPV)

If the test result is negative what is the probability that the patient does not have disease.

$$\text{NPV} = \text{tn} / (\text{tn} + \text{fn})$$

Miss rate or false negative rate (FNR)

It is the proportion of the individuals with a known positive condition for which the test result is negative.

$$\text{FNR} = \text{fn} / (\text{fp} + \text{tn})$$

Fall-out or false positive rate (FPR)

It is the proportion of all the people who do not have the disease who will be identified as having the disease.

$$\text{FPR} = \text{fp} / (\text{fp} + \text{tn})$$

False discovery rate (FDR)

It is the proportion of all the people identified as having the disease who do not have the disease.

$$\text{FDR} = \text{fp} / \text{fp} + \text{tp}$$

False omission rate (FOR)

It is the proportion of the individuals with a negative test result for which the true condition is positive.

$$\text{FOR} = \text{fn} / (\text{fn} + \text{tn})$$

Accuracy

The accuracy reflects the total proportion of individuals that are correctly classified.

$$\text{ACC} = (\text{tp} + \text{tn}) / (\text{tp} + \text{tn} + \text{fp} + \text{fn})$$

F1 score

It is the harmonic mean of precision and

$$\text{sensitivity } F1 = 2\text{tp} / (2\text{tp} + \text{fp} + \text{fn})$$

RMSE Score

Root mean square of the error that has occurred between the test values and the predicted values. It indicates the absolute fit of the model to the data—how close the observed data points are to the model's predicted values. Whereas R-squared is a relative measure of fit, RMSE is an absolute measure of fit. As the square root of a variance, RMSE can be interpreted as the standard deviation of the unexplained variance, and has the useful property of being in the same units as the response variable. Lower values of RMSE indicate better fit. RMSE is a good measure of how accurately the model predicts the response, and it is the most important criterion for fit if the main purpose of the model is prediction.

4. Implementation code

Extra tree classifier after removing of outliers:

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

from sklearn.preprocessing import MinMaxScaler

from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import accuracy_score, mean_squared_error
from sklearn.linear_model import LogisticRegression

from xgboost import XGBRFClassifier, XGBClassifier

from sklearn.svm import SVC

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import AdaBoostClassifier, BaggingClassifier, ExtraTreesClassifier, GradientBoostingClassifier, RandomForestClassifier

pkdata1 = pd.read_csv('C:/Users/ALEKHYA/Desktop/parkinsons.csv')

def remove_outlier(df_in, col_name):
    q1 = df_in[col_name].quantile(0.25)
    q3 = df_in[col_name].quantile(0.75)
    iqr = q3 - q1 #Interquartile range
    fence_low = q1 - 1.5 * iqr
    fence_high = q3 + 1.5 * iqr
    df_out = df_in.loc[(df_in[col_name] > fence_low) & (df_in[col_name] < fence_high)]
    return df_out

pkdata1 = remove_outlier(pkdata1, 'MDVP:Flo(Hz)')
pkdata1 = remove_outlier(pkdata1, 'MDVP:Fhi(Hz)')
pkdata1 = remove_outlier(pkdata1, 'MDVP:Jitter(%)')
```

```
pkdata1=remove_outlier(pkdata1,'MDVP:Jitter(Abs)')  
pkdata1=remove_outlier(pkdata1,'MDVP:Jitter(%)')  
pkdata1=remove_outlier(pkdata1,'MDVP:Shimmer')
```

```

pkdata1=remove_outlier(pkdata1,'MDVP:APQ')
pkdata1=remove_outlier(pkdata1,'NHR')
pkdata1=remove_outlier(pkdata1,'HNR')
pkdata1=remove_outlier(pkdata1,'spread1')
pkdata1=remove_outlier(pkdata1,'spread2')
pkdata1=remove_outlier(pkdata1,'D2')
pkdata1=remove_outlier(pkdata1,'PPE')
features=pkdata1.drop(['status','name'],axis=1)
labels=pkdata1['status']
scaler=MinMaxScaler((-1,1))
x=scaler.fit_transform(features)
y=labels
x_train,x_test,y_train,y_test=train_test_split(x, y, test_size=0.2, random_state=5)
lr=cross_val_score(LogisticRegression(),x_train,y_train)
xgbc=cross_val_score(XGBRFClassifier(),x_train,y_train)
xgb=cross_val_score(XGBClassifier(),x_train,y_train)
svm=cross_val_score(SVC(),x_train,y_train)
#nb=cross_val_score(MultinomialNB(),x_train,y_train)
dtc=cross_val_score(DecisionTreeClassifier(),x_train,y_train)
adb=cross_val_score(AdaBoostClassifier(),x_train,y_train)
bbc=cross_val_score(BaggingClassifier(),x_train,y_train)
etc=cross_val_score(ExtraTreesClassifier(),x_train,y_train)
gbc=cross_val_score(GradientBoostingClassifier(),x_train,y_train)
rfc=cross_val_score(RandomForestClassifier(),x_train,y_train)
#vc=cross_val_score(VotingClassifier(estimators),x_train,y_train)
print('log reg',lr,lr.mean())
print('xgbd',xgbc,xgbc.mean())
print('xgb',xgb,xgb.mean())
print('svm',svm,svm.mean())
#print('nb',nb,nb.mean())

```

```
print('dtc',dtc,dtc.mean())
```

```
print('adb',adb,adb.mean())  
print('bbc',bbc,bbc.mean())  
print('etc',etc,etc.mean())
```

```
print('gbc',gbc,gbc.mean())  
print('rfc',rfc,rfc.mean())  
#print('vc',vc,vc.mean())  
model=ExtraTreesClassifier()  
model.fit(x_train,y_train)  
y_pred=model.predict(x_test)  
print(accuracy_score(y_test, y_pred)*100)  
y_predtr=model.predict(x_train)  
print(accuracy_score(y_train,y_predtr)*100)
```

5. Result Analysis

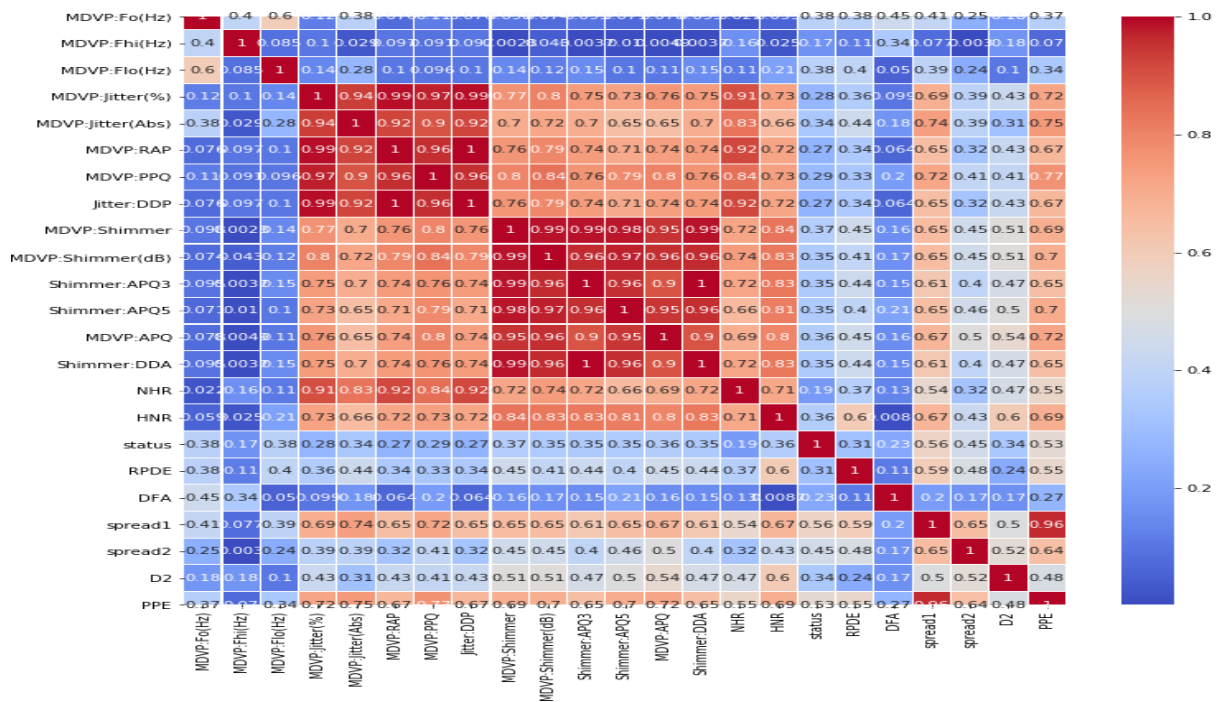


Fig:5.1 before applying correlation

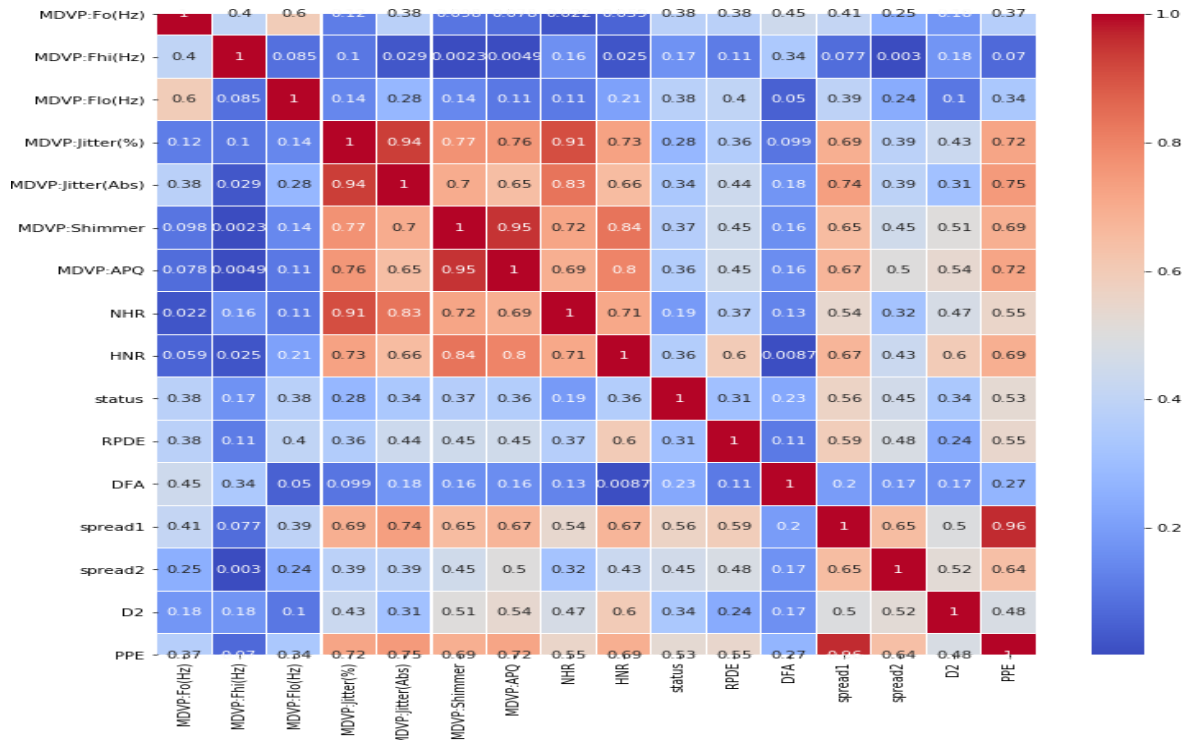


Fig:5.2 after applying correlation

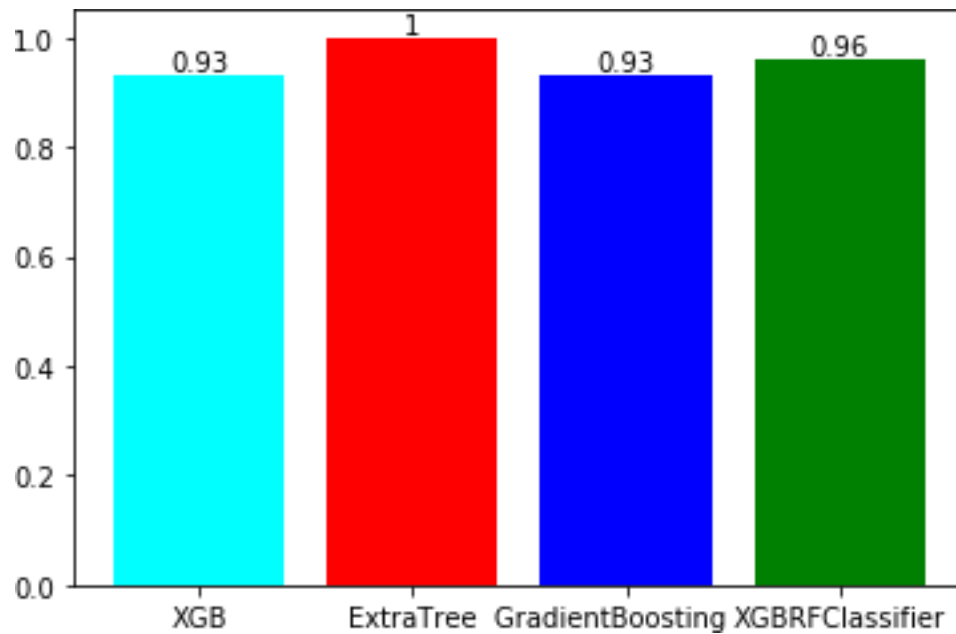


Fig: 5.3: comparison of algorithms after removal of outliers

Fig:5.3 shows accuracy score of algorithms after removing outliers from the dataset an outlier is an observation that diverges from an overall pattern on a sample.

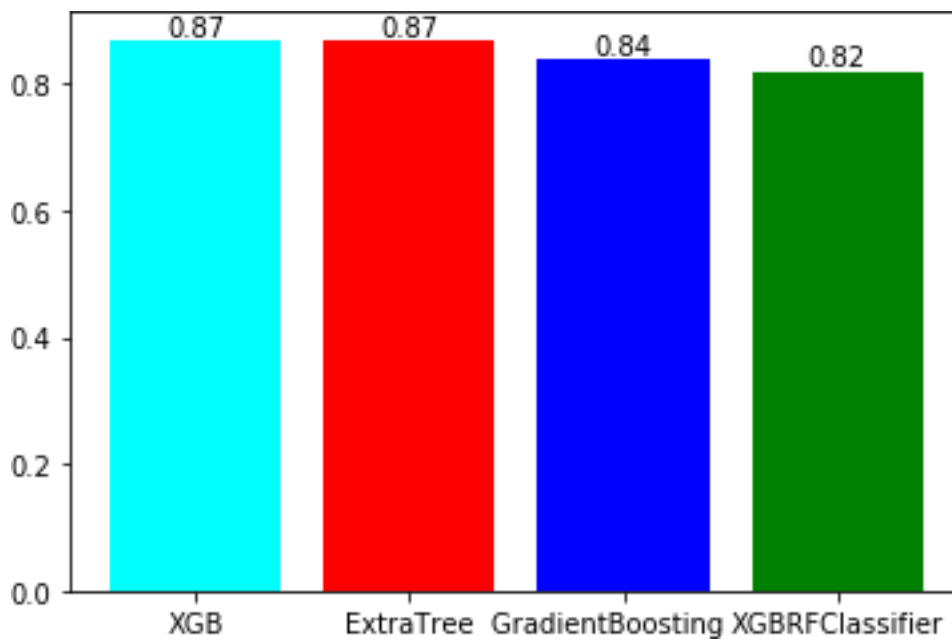


Fig:5.4 Comparison of algorithms after pca

Fig:5.4 shows accuracy score of algorithms after applying PCA. Principal Component Analysis (PCA) is a statistical procedure that uses an orthogonal transformation which converts a set of correlated variables to a set of uncorrelated variables.

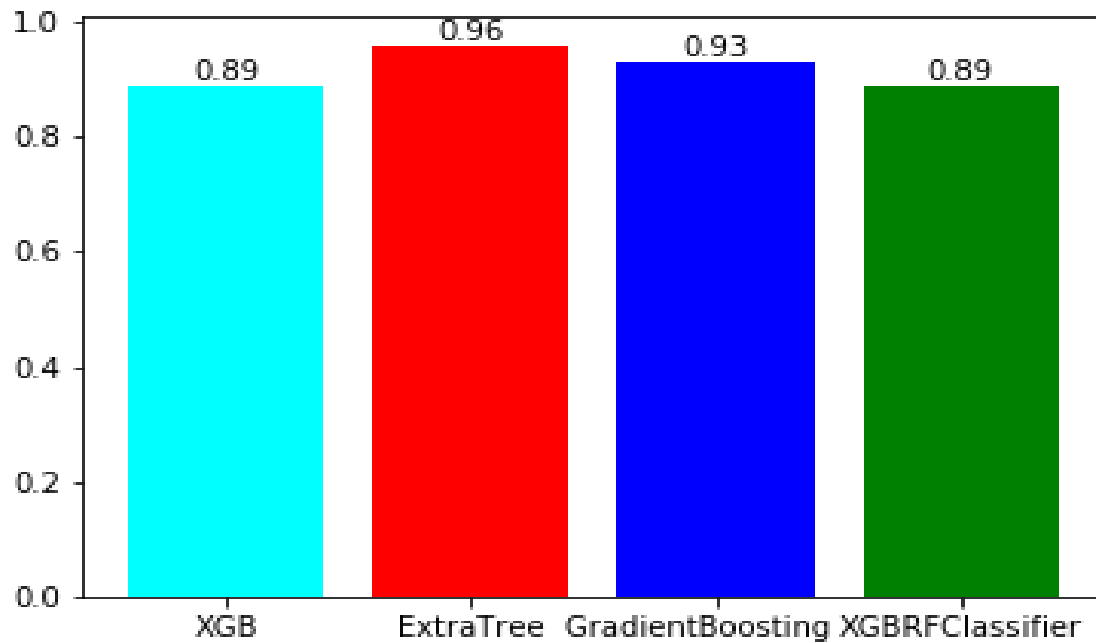


Fig:5.5 comparison of algorithms with correlation

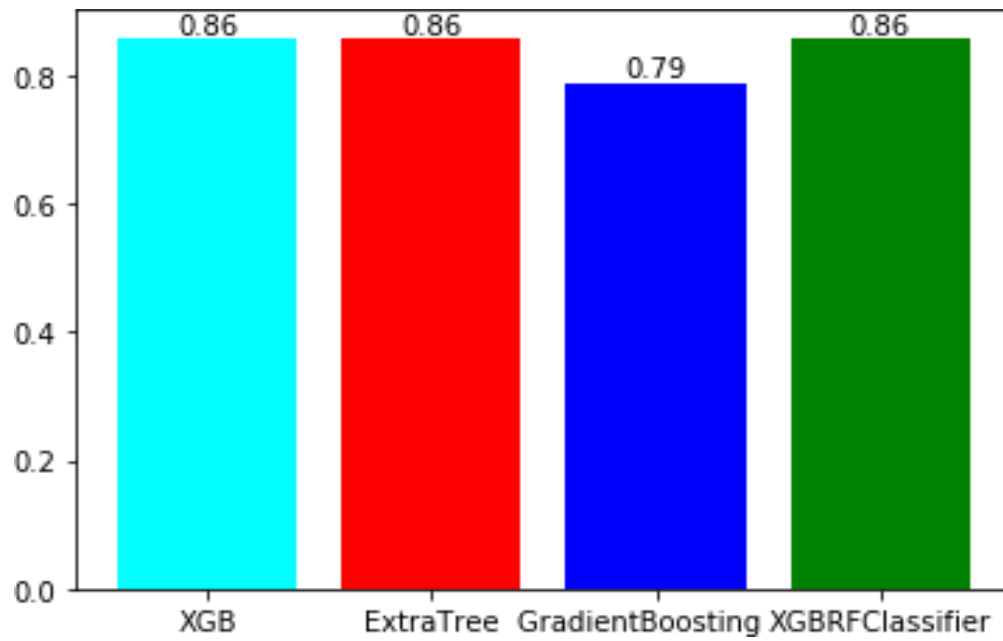


Fig:5.6 Comparison of algorithm after correlation and pca

Fig:5.6 shows accuracy score of algorithms after removing highly correlated attributes.

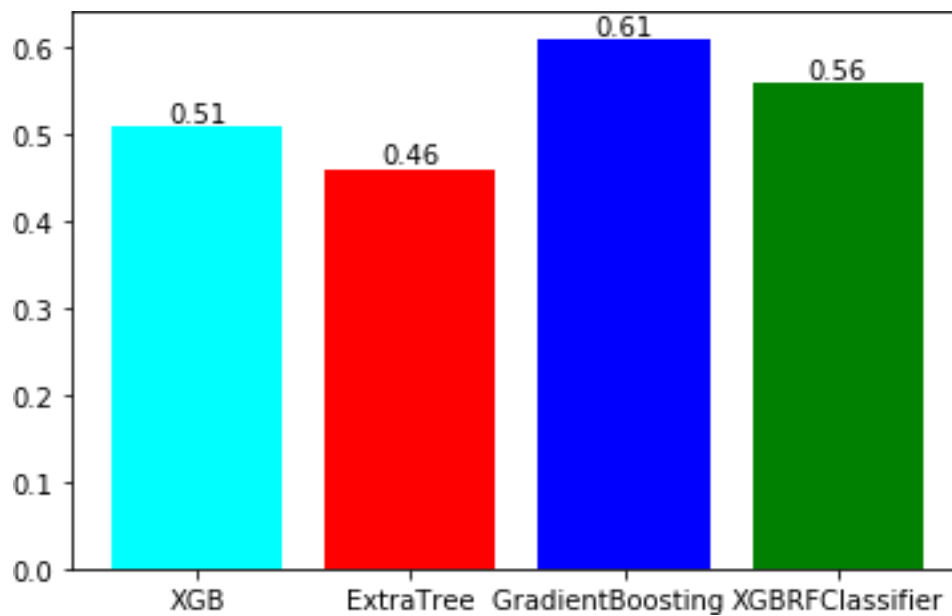


Fig:5.7 Comparison of algorithm after applying k-fold cross validation

Fig:5.7 shows accuracy score of algorithms after applying k-fold cross validation. It divides the dataset into k sets for every iteration kth set will be used as test set. So that every set will be equally treated and the model of an algorithm developed.

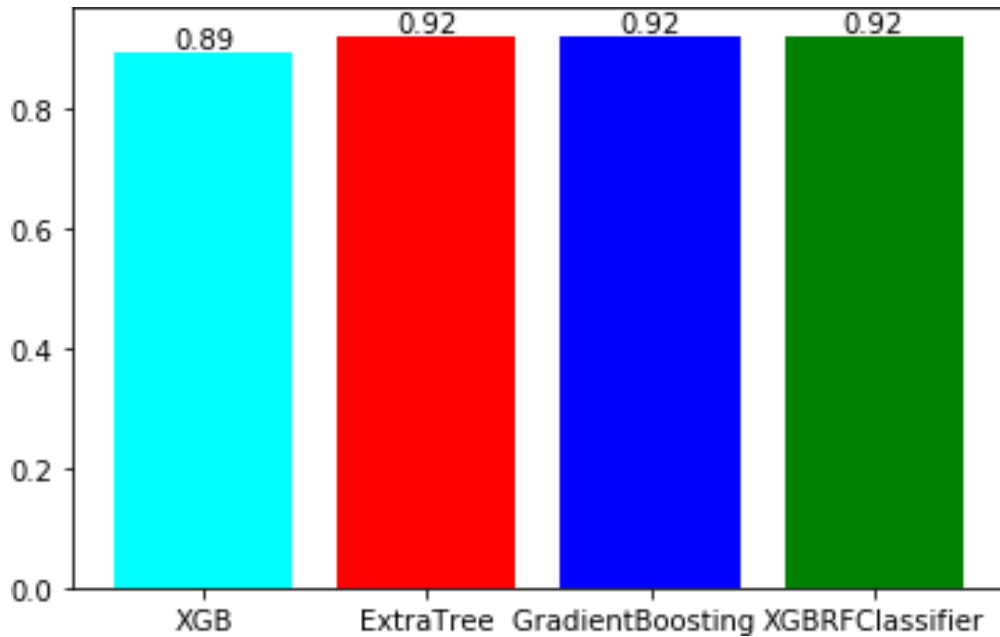


Fig 5.8: Comparison of algorithm after applying information gain

Fig:5.8 shows accuracy score of algorithms after finding information gain and removing attributes which have least information gain values

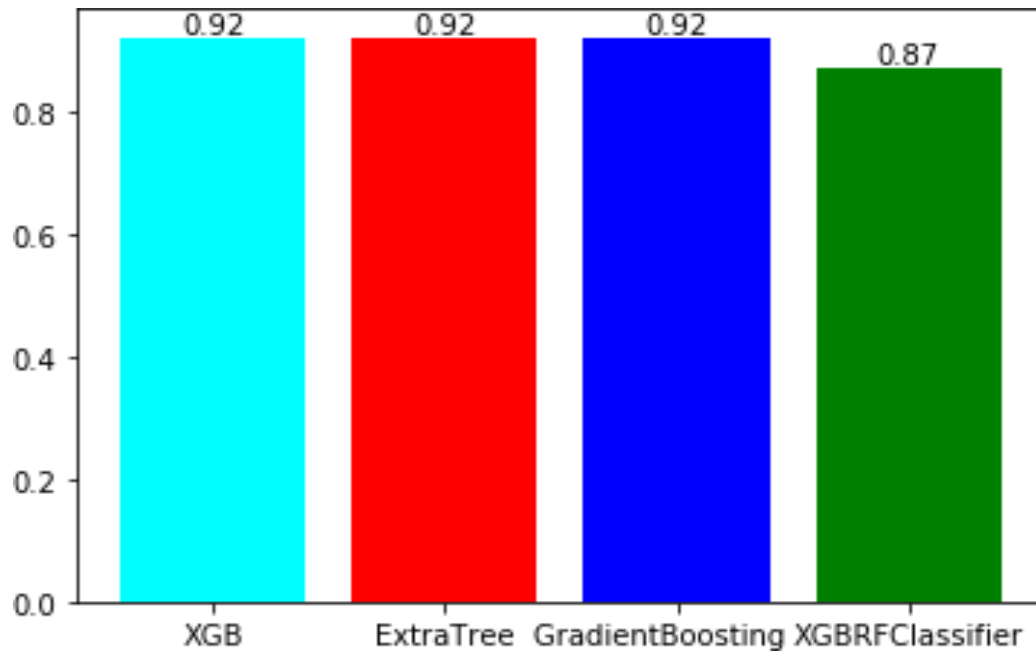
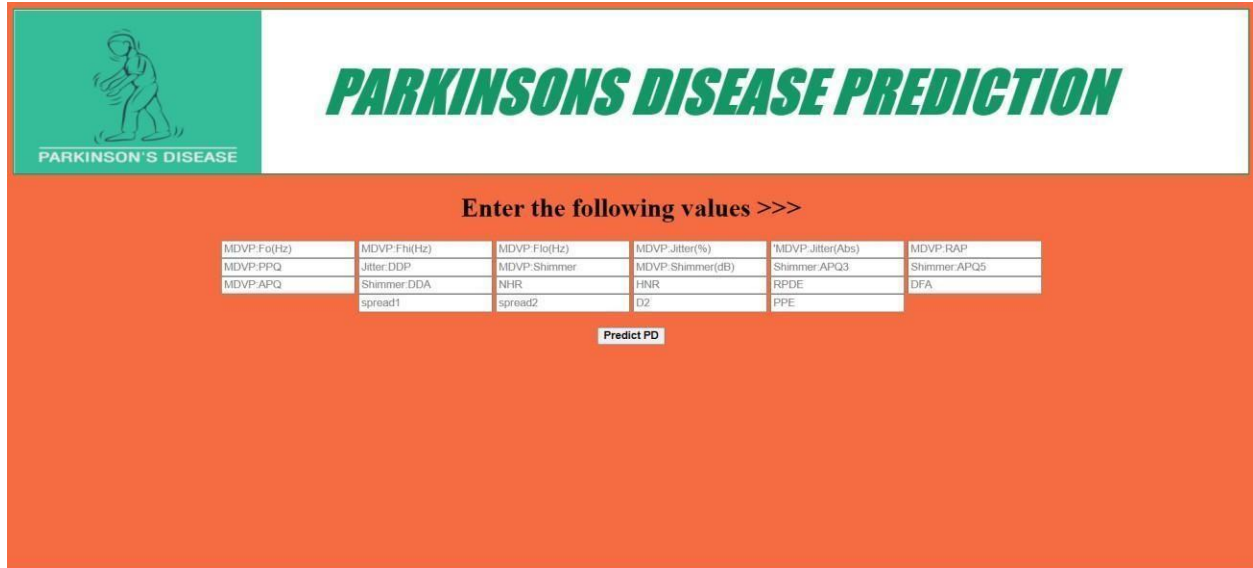


Fig 5.9: comparison of algorithms after applying smote

Fig:5.9 shows accuracy score of algorithms after applying SMOTE. SMOTE stands for Synthetic Minority Oversampling Technique. This is a statistical technique for increasing the number of cases in your dataset in a balanced way. This implementation of SMOTE does not change the number of majority cases

6. SCREENSHOTS

HOME PAGE:



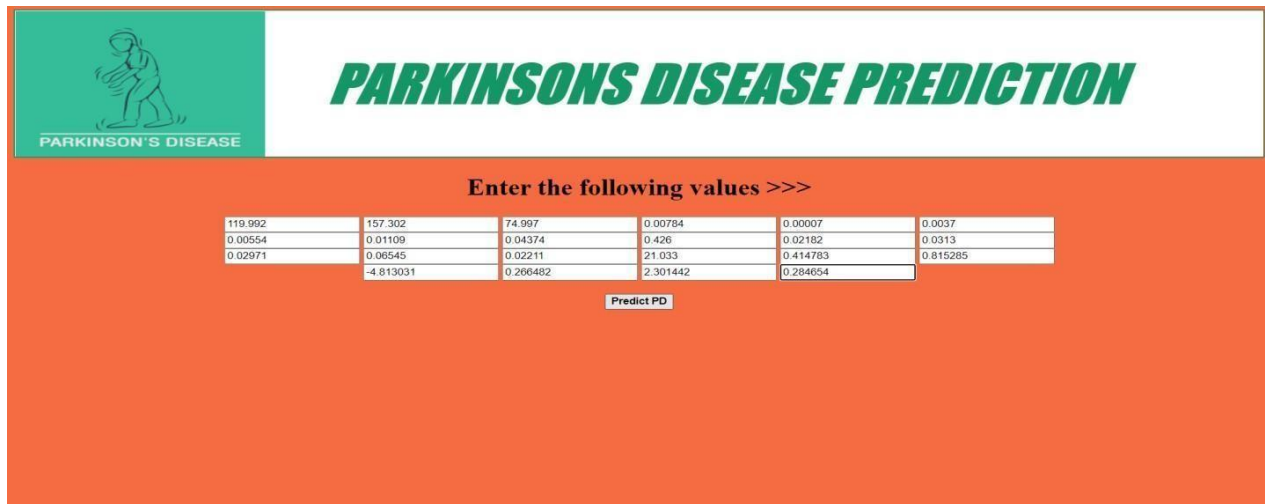
The screenshot shows the home page of the Parkinson's Disease Prediction application. It features a green header with a logo of a person walking and the text "PARKINSON'S DISEASE". The main title "PARKINSONS DISEASE PREDICTION" is displayed in large, bold, green letters. Below the title, the instruction "Enter the following values >>>" is shown. A table of input fields is provided, with labels for various acoustic features. The fields are arranged in a grid, and a "Predict PD" button is located at the bottom center.

MDVP:F0(Hz)	MDVP:F1(Hz)	MDVP:F0(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP
MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	MDVP:Shimmer(dB)	Shimmer:APQ3	Shimmer:APQ5
MDVP:APQ	Shimmer:DDA	NHR	HNR	RPDE	DFA
spread1	spread2	D2	PPE		

Predict PD

Fig: 6.1 Home page

ENTER VALUES:



The screenshot shows the input page of the Parkinson's Disease Prediction application. It features the same green header and main title as the home page. Below the title, the instruction "Enter the following values >>>" is shown. A table of input fields is provided, with numerical values entered into each field. The fields are arranged in a grid, and a "Predict PD" button is located at the bottom center.

119.992	157.302	74.997	0.00764	0.00007	0.0037
0.00554	0.01109	0.04374	0.426	0.02182	0.0313
0.02971	0.06545	0.02211	21.033	0.414783	0.815285
	-4.813031	0.266482	2.301442	0.284654	

Predict PD

Fig: 6.2 Input1

OUTPUT:



PARKINSONS DISEASE PREDICTION

Enter the following values >>>

MDVP:F0(Hz)	MDVP:F1(Hz)	MDVP:F1o(Hz)	MDVP:Jitter(%)	%MDVP:Jitter(Abs)	MDVP:RAP
MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	MDVP:Shimmer(dB)	Shimmer:APQ3	Shimmer:APQ5
MDVP:APQ	Shimmer:DDA	NHR	HNR	RPDE	DFA
	spread1	spread2	D2	PPE	

Predict PD

Patient has ** parkinsons disease **

Fig: 6.3 Has Disease

ENTER VALUES:



PARKINSONS DISEASE PREDICTION

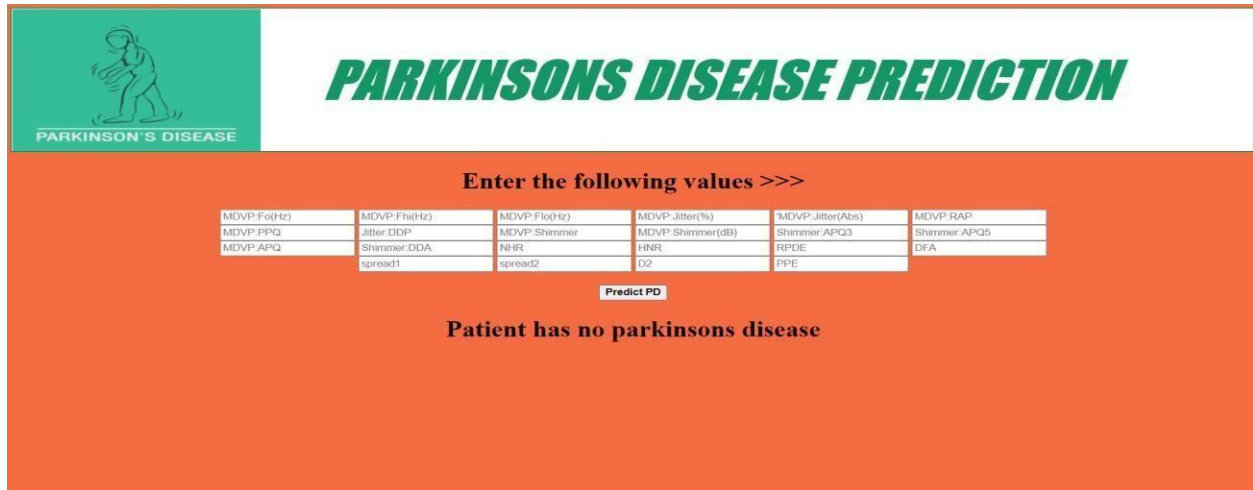
Enter the following values >>>

197.076	211.526	192.055	0.00289	0.00001	0.00166
0.00168	0.00498	0.01098	0.097	0.00468	0.0068
0.00802	0.01689	0.00339	21.209	0.340068	0.741367
	-7.3483	0.177551	1.743867	0.085569	

Predict PD

Fig: 6.4 Input 2

OUTPUT:



The image shows a web-based interface for Parkinson's Disease Prediction. It features a header with a logo and the title 'PARKINSONS DISEASE PREDICTION'. Below the header, there is a section titled 'Enter the following values >>>' containing a grid of input fields for various acoustic features. A 'Predict PD' button is located below the grid. The output of the prediction is displayed as 'Patient has no parkinsons disease'.

Enter the following values >>>					
MDVP-F0(Hz)	MDVP-F1(Hz)	MDVP-F2(Hz)	MDVP-Jitter(%)	MDVP-Jitter(Abs)	MDVP-RAP
MDVP-FPQ	Jitter-DDP	MDVP-Shimmer	MDVP-Shimmer(dB)	Shimmer-APQ3	Shimmer-APQ5
MDVP-APQ	Shimmer-DDA	HNR	HNR	RPDE	DFA
	spread1	spread2	D2	PPE	

Predict PD

Patient has no parkinsons disease

Fig: 6.5 Has no disease

we implemented extra tree classifier for detection of Parkinson's disease for prediction purposes which are collected from Kaggle.

The main motive of this project is to show how to tackle with such giant dataset as well as regularities.

CONCLUSION

- In this project, we implemented an extra tree classifier for detection of Parkinson's disease for prediction purposes which are collected from Kaggle.
- The main motive of this project is to show how to tackle such giant dataset as well as regularities.

FUTURE SCOPE

- In future, we can implement the website for this project.
- Also developing an android app or IOS app to provide users in an easy manner.

REFERENCES

- [1] <https://www.google.com/>
- [2] <https://scikit-learn.org/stable/>
- [3] <https://www.geeksforgeeks.org/machine-learning/>
- [4] <https://www.kaggle.com/>
- [5] <https://archive.ics.uci.edu/ml/index.php>