A Project Report

on

Early diagnosis of Parkinson's disease using machine learning

Submitted By:

Name: Sharmin Sultana Nimu

ID: 202011056061

Supervised By:

Umme Salma

Assistant Professor,

Department of CSE,

Bangladesh University

Dedicated to my parents

APPROVAL OF ACCEPTANCE

A project report written by Sharmin Sultana Nimu (ID: 202011056061) entitled "**Early diagnosis of Parkinson's disease using machine learning**" is submitted to the department of CSE, Bangladesh University. The project is done under supervision of Umme Salma, Assistant Professor, department of CSE, Bangladesh University.

We have examined this report and recommended its acceptance

Supervisor

Umme Salma,

Assistant Professor

Dept. of Computer Science and Engineering, Bangladesh University

Accepted by:

……..

Md. Sadiq Iqbal

Associate Professor

Head of the department

Computer Science and Engineering,

Bangladesh University

ACKNOWLEDGEMENT

First of all, I would like to express my gratitude to the almighty Allah for enabling me to complete this project. Later on, I would like to express my gratitude to my project supervisor Umme Salma, Assistant Professor, Computer Science & Engineering, Bangladesh University, for approving my project topic and imperative supervision. Her continuous encouragement, important suggestions, and necessary corrections helped me to complete this work successfully.

I would also like to acknowledge my teachers and friends. It would be impossible to continue my project without their valuable suggestions and ideas. I also thank my family members, especially my parents for their encouragement and support.

DECLARATION

I certify that the project report entitled, "**Early diagnosis of Parkinson's disease using machine learning**", submitted as a partial requirement for the degree of Bachelor of Science (Honors) in Computer Science and Engineering is the result of my research. I am here declaring that this project report has not been submitted elsewhere for the requirement of our kind of degree or publication.

Name: Sharmin Sultana Nimu
ID: 202011056061
Batch: 58
Department of CSE
Bangladesh University

Abstract

Parkinson's disease (PD) is a neurodegenerative disorder affecting 60% of people over the age of 50 years. Patients with Parkinson's face mobility challenges and speech difficulties, making physical visits for treatment and monitoring a hurdle. PD can be treated through early detection, thus enabling patients to lead a normal life. The rise of an aging population over the world emphasizes the need to detect PD early, remotely and accurately. This report highlights the use of machine learning techniques in telemedicine to detect PD in its early stages. Research has been carried out on the MDVP audio data of 31 people, 23 people with PD and healthy people during training of six ML models. Comparison of results of classification by Support Vector Machine (SVM), Random Forest, K-Nearest Neighbors (KNN), Logistic Regression models, Decision tree and XGBoost, yield Random Forest classifier as the ideal Machine Learning (ML) technique for detection of PD. Random Forest classifier model has a detection accuracy of 95%. We applied SHAP to see the performance of features. Through the findings of this report, we aim to promote the use of ML in telemedicine, thereby providing a new lease of life to patients suffering from Parkinson's disease.

List of Tables

List of Figures

Table of Contents

Chapter 1

Introduction

Parkinson’s disease was first described by Dr. James Parkinson in 1817, who called it "Shaking Palsy" because of the tremors it causes. Among brain-related diseases like epilepsy, Alzheimer's, and others, Parkinson’s disease is the second most common.

Diagnosing PD in its later stages is easier, but by then, it's hard to slow down the disease. Early diagnosis is challenging because its symptoms can resemble those of other neurological diseases. About 75% of Parkinson's cases are idiopathic, meaning there's no clear cause. Because of this complexity, new technologies like machine learning are being explored to help doctors diagnose PD earlier and more accurately, which could improve treatment and quality of life for patients.

Although there is no cure for Parkinson's, treatments like medications and physical therapy can help manage the symptoms. Research is ongoing, and scientists are exploring ways to diagnose the disease earlier and develop better treatments. Early detection and management can help improve the quality of life for people living with Parkinson's.

1.1: What is Parkinson’s disease?

Parkinson’s disease (PD) is a long-term, progressive neurological disorder that affects movement and coordination. Although it is most commonly associated with motor issues, the impact of PD goes beyond physical symptoms, influencing a person's daily life, emotions, and overall well-being.

Progression**:** Parkinson's disease typically progresses slowly, with symptoms gradually worsening over time. While every patient’s experience is unique, the disease generally moves through five stages, starting with mild symptoms and advancing to more severe impairment that can affect independence and daily functioning.

Diagnosis**:** There is no single test to diagnose Parkinson’s. Doctors usually rely on a combination of medical history, symptoms, and neurological exams. Imaging tests like MRI or dopamine transporter scans can help rule out other conditions, but diagnosing PD often requires time and observation of symptom progression.

Management**:** While there's no cure for Parkinson's, various treatments aim to manage symptoms and maintain quality of life. Medications like levodopa are commonly used to replace or mimic dopamine in the brain, easing movement-related problems. Beyond medication, physical therapy, occupational therapy, and speech therapy play vital roles in helping patients maintain mobility, strength, and communication skills.
Everyone experiences Parkinson's differently. For some, the symptoms might start out mild and get worse slowly. For others, the changes happen more quickly.

1.2: Causes of Parkinson's disease

It happens when certain cells in the brain, which make a chemical called dopamine, start to die. Dopamine helps control movements, so when there's less of it, a person might notice shaking in their hands or other parts of the body. But the exact cause of Parkinson's disease (PD) is not fully understood, but it is believed to be a combination of genetic and environmental factors.

Genetics play a role in some cases. Certain gene mutations can increase the risk of developing Parkinson's, especially if there's a family history of the disease. However, most people with Parkinson's do not have a strong family link.

Environmental factors may also contribute. Long-term exposure to toxins like pesticides, herbicides, or certain chemicals has been linked to an increased risk of Parkinson's. Head injuries and living in rural areas are also considered potential risk factors.
While aging is the biggest risk factor, with most cases developing after age 60, younger people can also get the disease. Researchers continue to study what triggers the death of dopamine-producing cells to better understand and hopefully prevent Parkinson's disease in the future.

1.3: Symptoms of Parkinson's disease

Parkinson's disease (PD) symptoms can be grouped into **motor** and **non-motor** categories. While the progression and combination of symptoms vary from person to person, the exact cause of the

disease isn't always known, a combination of genetics and environmental factors is believed to play a role.

Here is an overview of the common symptoms in a general order of appearance:

Motor Symptoms (related to movement):

1. **Tremor**:
    - The first noticeable symptom, often beginning as a slight shaking in the hands, fingers, or chin. It usually occurs at rest and improves with movement.
2. **Bradykinesia (Slowness of Movement)**:
    - This is a hallmark symptom where movements become slower and harder to initiate. It can make everyday tasks like walking, dressing, and writing difficult.
3. **Rigidity**:
    - Muscle stiffness in the arms, legs, or neck. This can limit range of motion and cause discomfort or pain.
4. **Postural Instability**:
    - Problems with balance and coordination, often leading to unsteadiness, stooped posture, and an increased risk of falling.
5. **Gait Changes**:
    - A shuffling walk with small steps, difficulty starting movement, or freezing episodes (where the person feels "stuck" in place) may develop.
6. **Reduced Facial Expression (Hypomimia)**:
    - A lack of facial movement, often referred to as a "masked face," where expressions become less animated.
7. **Speech Changes**:
    - Speech may become soft (hypophonia), monotone, slurred, or mumbled. Some people may speak too quickly or too slowly.
8. **Micrographic**:
    - Handwriting becomes smaller and more cramped over time.

Non-Motor Symptoms (affecting other systems in the body):

1. **Loss of Smell (Hyposmia)**:
    - A reduced ability to smell or taste, often one of the earliest signs.

2. **Sleep Problems**:
    - These can include insomnia, restless legs syndrome, or REM sleep behavior disorder (acting out dreams).
3. **Depression and Anxiety**:
    - Mood changes, such as feeling sad or anxious, can occur early in the disease and may worsen over time.
4. **Fatigue**:
    - A sense of extreme tiredness, unrelated to physical activity, is common.
5. **Cognitive Changes**:
    - Some people experience memory problems, difficulty concentrating, or slower thinking (mild cognitive impairment). In advanced stages, dementia can develop.
6. **Autonomic Dysfunction**:
    - Problems with automatic bodily functions, like constipation, urinary urgency or retention, low blood pressure (orthostatic hypotension), and excessive sweating.
7. **Pain**:
    - Many individuals experience muscle cramps, joint pain, or a general aching feeling.
8. **Swallowing and Drooling**:
    - Difficulty swallowing (dysphagia) can occur, along with excessive saliva or drooling (sialorrhea) due to reduced swallowing frequency.
9. **Vision Problems**:
    - Blurred vision, dry eyes, or difficulty with eye movements can appear.
10. **Cognitive and Behavioral Symptoms**:
    - Memory problems, confusion, hallucinations, and impulsive behavior may emerge in the later stages.

1.3.1: What lifestyle changes can help manage Parkinson's symptoms?

Lifestyle changes play a crucial role in managing Parkinson's disease (PD) and improving overall well-being. Regular exercise is one of the most effective ways to maintain mobility, balance, and strength. Activities like walking, swimming, or yoga can help reduce stiffness, improve flexibility, and promote better posture. Exercise also stimulates the brain, which may help slow the progression of motor symptoms.

A healthy diet is essential for maintaining energy levels, reducing inflammation, and managing weight. Eating a balanced diet rich in fruits, vegetables, whole grains, and lean proteins can also help support brain health. Staying hydrated and managing constipation, which is common in PD, is important too.

Remaining socially and mentally active helps combat non-motor symptoms such as depression, anxiety, and cognitive decline. Engaging in social activities, hobbies, or cognitive exercises like puzzles or reading can boost mood and cognitive function.

Physical therapy is often recommended to maintain muscle function and improve movement, while occupational therapy helps patients adapt their environment to make daily tasks easier. Mindfulness practices like meditation or deep breathing can reduce stress and improve emotional well-being.

Together, these lifestyle changes help individuals with Parkinson's maintain independence, manage symptoms more effectively, and enhance their overall quality of life.

1.4: Scenario in Bangladesh

In Bangladesh, the scenario regarding Parkinson's disease (PD) is becoming increasingly recognized, but the country faces several challenges in diagnosis, treatment, and awareness.

Prevalence and Awareness:

Parkinson's disease is not widely known in Bangladesh, especially in rural areas, where access to healthcare and education about neurological diseases is limited. The prevalence of PD in Bangladesh is not well-documented, but like other countries, it is more common in older

populations. As the country's elderly population grows, so does the number of people affected by neurodegenerative diseases, including Parkinson's.

Diagnosis and Treatment:

In Bangladesh, diagnosing PD can be difficult, especially in remote or underdeveloped regions. There is a lack of specialized healthcare professionals, such as neurologists, who are essential for proper diagnosis and treatment. Misdiagnosis is common, and many people only seek medical help in the later stages of the disease when symptoms become severe.

Treatment options are also limited. Although medications like levodopa (the standard treatment for Parkinson's) are available, they can be expensive and difficult to access for many. Furthermore, advanced treatment methods like deep brain stimulation (DBS) are not widely available, limiting the options for patients with advanced PD.

Healthcare Infrastructure:

Bangladesh's healthcare infrastructure is improving but remains under strain due to limited resources, especially in the field of neurology. While major cities like Dhaka have some neurological centers, rural areas still lack the facilities and trained staff to handle cases like PD effectively. Rehabilitation services, which are crucial for managing PD symptoms, such as physiotherapy and speech therapy, are also scarce.

Research and Support:

Research on Parkinson's disease in Bangladesh is minimal, and public awareness campaigns are rare. There are few support groups or community resources for people with PD, which leaves patients and their families isolated in managing the disease. Additionally, many people rely on traditional medicine or may not seek treatment due to stigma or a lack of knowledge about the disease.

1.5: Contribution of this project work

1. **Early Detection of Parkinson's Disease (PD)**: To use machine learning techniques to detect Parkinson's disease in its early stages, enabling timely treatment and improving patient outcomes.
2. **Promote Remote Diagnosis via Telemedicine**: To develop a telemedicine solution that allows for the remote and accurate diagnosis of Parkinson's disease, reducing the need for physical visits, especially for patients with mobility challenges.
3. **Comparison of Machine Learning Models**: To train and compare the performance of six machine learning models (SVM, Random Forest, K-Nearest Neighbors, Logistic Regression, Decision Tree, and XGBoost) for PD detection based on MDVP audio data.
4. **Identify the Best Performing Model**: To determine the most effective machine learning model for Parkinson's detection.
5. **Feature Importance Analysis using SHAP**: To apply SHAP (Shapley Additive explanations) to analyze the performance and importance of features used in the machine learning models for a more interpretable and explainable diagnosis.
6. **Support Aging Populations**: To address the growing needs of the aging population by improving early diagnosis and treatment accessibility through advanced technology.
7. **Raise Awareness for ML in Healthcare**: To highlight the potential of machine learning in revolutionizing healthcare, especially in managing neurodegenerative disorders like Parkinson's disease.

Chapter-2

Literature review

Several works have investigated the diagnosis of PD, in which many machine learning methods were applied such as Support Vector Machine, neural network, Naïve Bayes, K-nearest neighbor and Random Forests. In this report, several datasets were used to search for related studies on Parkinson's disease, including Scopus, IEEE Xplore, Science Direct and Google Scholar.

Machine learning (ML) is frequently used for medical disease diagnosis recently because of its implementation convenience and high accuracy. [3]

PD detection from speech can be regarded as a two-step task. The first step is to transfer the input speech signal to speech feature vectors or tensors that can be analyzed by DL models. Regarding the speech features of PD patients, several dimensions of speech are included, such as articulation, phonation, prosody, etc. [4] Mostafa et al. [5] tried to enhance the diagnoses of PD by using several methods of feature evaluation and classification. They used a multi-agent system to evaluate multiple features by using five classification methods, namely DT, NB, NN, RF, and SVM. To evaluate the proposed method, they conducted several experiments using original and filtered datasets. The results depicted that this method enhanced the performance of ML methods used by finding the best set of features. In this paper a novel approach has been proposed to diagnose PD using the gait analysis, that consists of the gait cycle, which can be broken down into various phases and periods to determine normative and abnormal gait. An average accuracy of 92.7% is achieved for the diagnosis of PD from gait analysis and tremor analysis is used for knowing the severity of PD.[7] The recent diagnosis methodology to Parkinson’s disease relies on voice disorders analysis. This methodology entails extracting feature sets of a recorded person’s voice then utilizing a machine learning technique to identify the healthy and Parkinson’s cases from the voice.[8] In a study by Sztaho et al., linear regression models based on the acoustic characteristic from the middle of the vowels are used to classify phrases and continuous talks. [9]

2.1: Comparison of prediction models of related works

Table 2.1: Comparison of prediction models of relate works

Ref	Objective	Methodology	Dataset	Key findings
[1]	enhance the prediction of PD using several machine learning methods with different feature selection methods	NB, KNN, SVM, MLP, RF	dataset includes 240 recodes with 46 acoustic features extracted from 3 voice recording replications for 80 patients.	KNN: Accuracy 88.33%
[2]	machine learning based diagnosis of Parkinson's disease	ANN, SVM	Not mentioned	SVM: Accuracy 93.63%
[6]	discriminate between two groups of participants (patients with Parkinson's disease and healthy people) by analyzing 3 types of voice recordings.	SVM, KNN	multiple types of voice recording of three sustained vowels /a/, /o/ and /u/ at a comfortable level which was collected from the 40 participants (20 PD and 20 healthy)	87.5% using linear kernel of SVM

Chapter 3

Data & Variables

Information about the data source, sample profile of the study and a concise description of the variables used to carry on the analysis are described in this chapter.

3.1: Source of data & data description:

The dataset was created by Max Little of the University of Oxford, in collaboration with the National Centre for Voice and Speech, Denver, Colorado, who recorded the speech signals. The original study published the feature extraction methods for general voice disorders. This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). Each column in the table is a particular voice measure, and each row corresponds one of 195 voice recording from these individuals ("name" column). The main aim of the data is to discriminate healthy people from those with PD, according to "status" column which is set to 0 for healthy and 1 for PD.

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

Table 3.1: Dataset description

Variables	**Description**	**Categories**	
		High value	**Low value**
Avg_fre	(MDVP (Hz)) Average Frequency	Higher average frequency usually indicates normal voice, suggesting lower likelihood of Parkinson's disease.	Lower average frequency indicates reduced voice stability, which may be a symptom of Parkinson's disease.
max_fre	(MDVP (Hz)) Maximum Frequency	Higher maximum frequency indicates better voice stability, suggesting a lower likelihood of Parkinson's disease.	Lower maximum frequency indicates voice instability, which may be a sign of Parkinson's disease.

min_fre	(MDVP(Hz)) Minimum Frequency	Higher minimum frequency suggests a normal voice.	Very low minimum frequency indicates voice instability, which can be seen in Parkinson's disease.
var_fre1	(MDVP (%)) Frequency Variability	Frequency Variability Higher jitter (frequency variability) indicates voice instability, which can be a sign of Parkinson's disease.	Lower jitter indicates voice stability, suggesting healthiness.
var_fre2	(MDVP(Abs)) Frequency Variation	Higher variation indicates significant changes in voice frequency, which can be a symptom of Parkinson's disease.	Lower variation indicates a stable voice, suggesting healthiness.
var_fre3 and var_fre4	Frequency Variation	Higher RAP (Relative Amplitude Perturbation) and PPQ (Pitch Perturbation Quotient) indicate greater frequency changes, which can be a sign of Parkinson's disease.	Lower RAP and PPQ suggest less frequency change, indicating a healthier voice.
var_amp1	(MDVP) Frequency Variation	Higher shimmer (amplitude variation) indicates voice instability, which can be a symptom of Parkinson's disease.	Lower shimmer indicates stable voice amplitude, suggesting healthiness.
var_amp2 and var_amp3	Amplitude Variation	Higher amplitude variation can be a sign of Parkinson's disease.	Lower amplitude variation indicates a healthy voice.
NHR	Noise to Harmonic Ratio	Higher NHR indicates more noise in the voice, which can be a sign of Parkinson's disease.	Lower NHR suggests a voice with more harmonics, indicating healthiness.
HNR	Harmonic to Noise Ratio	Higher HNR indicates less noise in the voice, which suggests a healthy voice.	Lower HNR indicates more noise in the voice, which can be a symptom of Parkinson's disease.

RPDE	Recurrence Period Density Entropy	Higher RPDE indicates greater complexity in the voice, which may be a sign of Parkinson's disease.	Lower RPDE indicates less complexity, suggesting a healthier voice.
DFA	Detrended Fluctuation Analysis	Higher DFA indicates stable frequency changes, suggesting healthiness.	Lower DFA indicates more variability in frequency changes, which can be a sign of Parkinson's disease.
spread1 and spread2	Frequency Spread	Higher frequency spread indicates more changes in the voice, which can be a symptom of Parkinson's disease.	Lower frequency spread suggests a stable voice, indicating healthiness.
D2	Dynamic Complexity	Higher D2 indicates greater dynamic complexity, which may be a sign of Parkinson's disease.	Lower D2 suggests less complexity, indicating a healthier voice.
PPE	Particle Percent Energy	Higher PPE indicates more energy changes in the voice, which can be a symptom of Parkinson's disease.	Lower PPE suggests a stable voice energy, indicating healthiness.

3.2: Data preprocessing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. For this, we use data preprocessing task.

3.2.1: Why do we need Data Preprocessing?

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks

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.

3.2.2: Process of data pre-processing

- To understand the dataset, I renamed the features names in fig:3.1.

	name	avg_fre	max_fre	min_fre	var_fre1	var_fre2	var_fre3	var_fre4	var_fre5	var_amp1	...	var_amp6	NHR	HNR	status	RPDE	DFA	spread1	spread2	D2	
	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109	0.04374	...	0.06545	0.02211	21.033	1	0.414783	0.815285	-4.813031	0.266482	2.301442	0
	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	...	0.09403	0.01929	19.085	1	0.458359	0.819521	-4.075192	0.335590	2.486855	0
	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633	0.05233	...	0.08270	0.01309	20.651	1	0.429895	0.825288	-4.443179	0.311173	2.342259	0
	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	...	0.08771	0.01353	20.644	1	0.434969	0.819235	-4.117501	0.334147	2.405554	0
	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	...	0.10470	0.01767	19.649	1	0.417356	0.823484	-3.747787	0.234513	2.332180	0
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
190	phon_R01_S50_2	174.188	230.978	94.261	0.00459	0.00003	0.00263	0.00259	0.00790	0.04087	...	0.07008	0.02764	19.517	0	0.448439	0.657899	-6.538586	0.121952	2.657476	0
191	phon_R01_S50_3	209.516	253.017	89.488	0.00564	0.00003	0.00331	0.00292	0.00994	0.02751	...	0.04812	0.01810	19.147	0	0.431674	0.683244	-6.195325	0.129303	2.784312	0
192	phon_R01_S50_4	174.688	240.005	74.287	0.01360	0.00008	0.00624	0.00564	0.01873	0.02308	...	0.03804	0.10715	17.883	0	0.407567	0.655683	-6.787197	0.158453	2.679772	0
193	phon_R01_S50_5	198.764	396.961	74.904	0.00740	0.00004	0.00370	0.00390	0.01109	0.02296	...	0.03794	0.07223	19.020	0	0.451221	0.643956	-6.744577	0.207454	2.138608	0
194	phon_R01_S50_6	214.289	260.277	77.973	0.00567	0.00003	0.00295	0.00317	0.00885	0.01884	...	0.03078	0.04398	21.209	0	0.462803	0.664357	-5.724056	0.190667	2.555477	0

Fig 3.1: Renamed the dataset

- Check the data types and null values

  I have checked the data types of dataset and checked if there were any null values in fig:3.2.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 24 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   name      195 non-null    object
 1   avg_fre   195 non-null    float64
 2   max_fre   195 non-null    float64
 3   min_fre   195 non-null    float64
 4   var_fre1  195 non-null    float64
 5   var_fre2  195 non-null    float64
 6   var_fre3  195 non-null    float64
 7   var_fre4  195 non-null    float64
 8   var_fre5  195 non-null    float64
 9   var_amp1  195 non-null    float64
 10  var_amp2  195 non-null    float64
 11  var_amp3  195 non-null    float64
 12  var_amp4  195 non-null    float64
 13  var_amp5  195 non-null    float64
 14  var_amp6  195 non-null    float64
 15  NHR       195 non-null    float64
 16  HNR       195 non-null    float64
 17  status    195 non-null    int64
 18  RPDE      195 non-null    float64
 19  DFA       195 non-null    float64
 20  spread1   195 non-null    float64
 21  spread2   195 non-null    float64
 22  D2        195 non-null    float64
 23  PPE       195 non-null    float64
dtypes: float64(22), int64(1), object(1)
memory usage: 36.7+ KB
```

Fig 3.2: Data types and check null values

- Drop unnecessary ‘name’ column

  I dropped the ‘name’ column which was unnecessary column in fig:3.3.

```
data.drop(columns="name",axis=1,inplace=True)
data
```

	avg_fre	max_fre	min_fre	var_fre1	var_fre2	var_fre3
0	119.992	157.302	74.997	0.00784	0.00007	0.00370
1	122.400	148.650	113.819	0.00968	0.00008	0.00465
2	116.682	131.111	111.555	0.01050	0.00009	0.00544
3	116.676	137.871	111.366	0.00997	0.00009	0.00502
4	116.014	141.781	110.655	0.01284	0.00011	0.00655
...	...	...	...	...	...	...
190	174.188	230.978	94.261	0.00459	0.00003	0.00263
191	209.516	253.017	89.488	0.00564	0.00003	0.00331
192	174.688	240.005	74.287	0.01360	0.00008	0.00624
193	198.764	396.961	74.904	0.00740	0.00004	0.00370
194	214.289	260.277	77.973	0.00567	0.00003	0.00295

Fig 3.3: Drop unnecessary column

- Histogram plots for all features

  Here I have shown the lots for all input and output features in fig 3.4

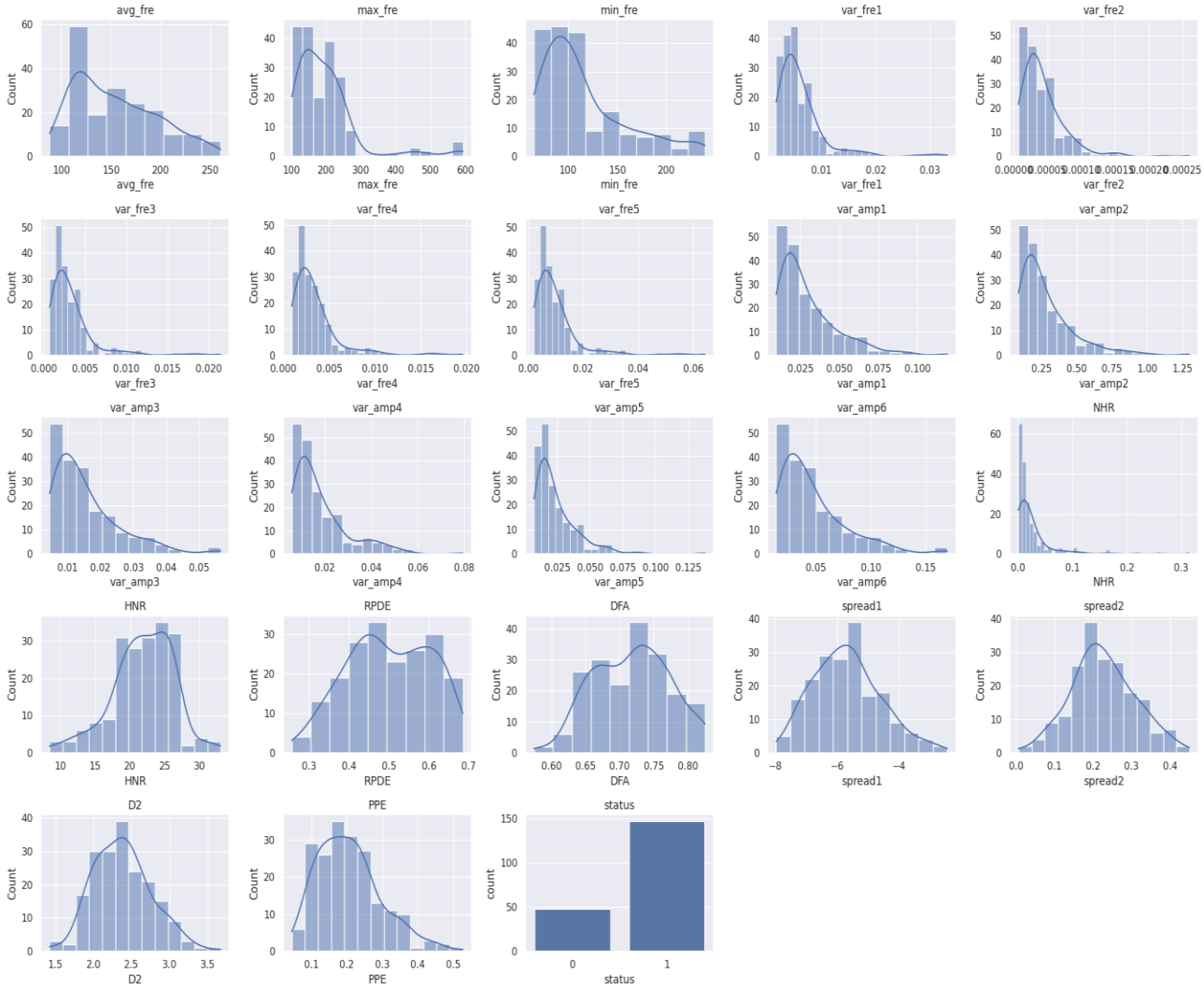


Fig 3.4: Plots shown for all features

- Calculate the mean and standard deviation
- Here the output feature is ‘status’ column, 0 means healthy and 1 means people with parkinson’s disease.

  Here I have shown the output feature plot in fig:3.5

Fig 3.5: Output feature

3.2.3: Splitting Data

At first splitting the dataset. Then dataset have two part

Y = dependent variable

X = independent variable

Train/Test is a method to measure the accuracy of my model. It is called Train/Test because I split the data set into two sets: a training set and a testing set. I select the dataset has 80% for training, and 20% for testing.

Chapter-4

Materials and Methods

4.1: Support vector machine (SVM):

Support Vector Machine (SVM) is a supervised machine learning algorithm primarily used for classification tasks, though it can also be applied to regression. Here's a detailed explanation of its concepts and workings:

- Hyperplane: In an n-dimensional space, a hyperplane is a flat affine subspace of dimension n−1n-1n−1. In the context of SVM, it is the decision boundary that separates different classes. The goal of SVM is to find the optimal hyperplane that maximizes the margin between the classes.
- Margin: The margin is the distance between the hyperplane and the nearest data points from either class. SVM aims to maximize this margin, as a larger margin is associated with better generalization to unseen data.
- Support Vectors**:** These are the data points that lie closest to the hyperplane and are crucial for defining it. If the support vectors were removed, the position of the hyperplane would change. The algorithm focuses on these points rather than the entire dataset, making it more efficient.

4.1.1: How SVM works

1. **Linear SVM**: In cases where the data is linearly separable, SVM finds a hyperplane that separates the two classes. It calculates the optimal hyperplane by maximizing the margin. The mathematical formulation involves finding weights and biases that define the hyperplane.

The equation of a hyperplane in an mmm-dimensional space can be written as:

$$w * x + b = 0$$

where:

- w is the weight vector (normal to the hyperplane).

- b is the bias term.

2. **Non-linear SVM**: Many real-world datasets are not linearly separable. To address this, SVM employs a technique called the **kernel trick**. This involves transforming the original feature space into a higher-dimensional space where a linear hyperplane can effectively separate the classes. Common kernels include:
    - **Polynomial Kernel**: Maps the data into a higher-dimensional space using polynomial functions.

$$K(x_i, x_j) = (x_i * x_j + c)d$$

    - **Radial Basis Function (RBF) Kernel**: Maps data into an infinite-dimensional space, which is effective for many non-linear problems.

$$\mathrm{K}(x_i, x_j) = \exp(-2\sigma 2 \parallel \mathrm{x}_i - \mathrm{x}_j \parallel 2)$$

3. **Soft Margin**: In practice, some data points may be misclassified due to noise or overlap between classes. To handle this, SVM introduces a soft margin that allows some misclassification while still aiming to maximize the margin. This is controlled by a parameter CCC that balances the trade-off between maximizing the margin and minimizing classification errors.

4.2: Random forest:

Random Forest is an ensemble learning method primarily used for classification and regression tasks. It builds multiple decision trees using bootstrap sampling and random feature selection to improve classification and regression performance. By aggregating the predictions from multiple trees, it achieves high accuracy, robustness, and a reduction in overfitting, making it a popular choice in machine learning applications. The core idea behind Random Forest is to use many decision trees instead of just one. Each tree in the forest is trained on a random subset of the training data, and its predictions are combined to make the final prediction. This "wisdom of the crowd" approach helps to reduce overfitting and improve generalization.

It is often one of the most accurate machine learning algorithms due to its ability to reduce overfitting and handle complex data. Unlike individual decision trees, Random Forest is less likely to overfit because of the random sampling of data and features. Random Forest can handle missing

data by using surrogate splits, allowing it to make predictions even with incomplete datasets. Random Forest can calculate the importance of each feature based on how often it's used in splits across the trees, which can be useful for feature selection.

4.2.1: How random forest works

Let D be the training dataset with N samples. The process can be mathematically represented as follows:

1. **Bootstrap Sample Creation**: For each b=1,2,….,B;
   - Generate a bootstrap sample D_b from D.

2. **Training Trees**: For each bootstrap sample D_b :
   - Construct a decision tree T_b using a random subset of features at each split.

3. **Prediction Aggregation**: For a new instance x:
   - For classification:

$$y^{\wedge} = mode(T1(x), T2(x), \dots, TB(x))$$

   - For regression:

$$y^{\wedge} = B1b = 1\sum BTb(x)$$

4.3: K-nearest neighbor:

The K-Nearest Neighbors (KNN) algorithm is a simple, yet powerful, supervised learning algorithm used for both classification and regression tasks. It operates on the principle that similar data points are likely to be near each other in feature space. is based on the idea that data points that are close to each other are likely to belong to the same class or have similar values. To classify or predict for a new data point, KNN looks at the k closest points (neighbors) from the training dataset and uses them to make a decision.

- For classification, it assigns the class that is most frequent among the neighbors.
- For regression, it takes the average value of the neighbors.

4.3.1: Algorithm steps of k nearest neighbor

Here's how KNN works step-by-step:

Step 1: Choose k

- Choose the number of neighbors k. This is a hyperparameter that defines how many nearest neighbors to look at for making predictions.

Step 2: Compute Distance

- For each point in the dataset, compute the distance between the new data point and all training data points. The most common distance metrics are:
    - Euclidean distance:

$$d(x, x_j) = j = 1\sum m(x_j - x_{ij})2$$

where x is the new data point, x_i is a training point, and m is the number of features.

- Manhattan distance:

$$d(x, x_j) = j = 1\sum m \mid x_j - x_{ij} \mid$$

Step 3: Find Nearest Neighbors

- Identify the k nearest points in the training dataset based on the computed distances. These are the nearest neighbors.

Step 4: Make Predictions

- **For classification**: Take the majority class of the nearest neighbors. For instance, if k=5k = 5k=5 and 3 of the nearest neighbors are labeled as class A and 2 as class B, the new point will be classified as class A.
- **For regression**: Take the mean (or sometimes median) of the values of the nearest neighbors.

4.4: Logistic regression:

Logistic Regression is a supervised learning algorithm primarily used for **binary classification** problems, where the output variable is categorical with two possible outcomes (e.g., "yes/no", "0/1", "spam/ham"). Although it has "regression" in its name, logistic regression is fundamentally a classification technique. It is computationally efficient, even for large datasets. The coefficients

w have a straightforward interpretation; they represent the log-odds of the event happening. Logistic regression provides well-calibrated probabilities for class membership.

Logistic regression is a powerful algorithm for binary classification that models the probability of class membership using a linear combination of input features and the sigmoid function. It's simple, interpretable, and effective for many tasks where the relationship between features and the target is approximately linear. Regularization and careful feature selection are key to preventing overfitting and improving performance.

4.4.1: How logistic regression works:

In binary classification, the output variable y can take only two possible values:

- y=1y = 1y=1 (positive class)
- y=0y = 0y=0 (negative class)

Given a dataset of input features $x=(x_1,x_2,\ldots,x_m)$ we want to predict the probability that y=1 (i.e., the probability that the data point belongs to the positive class).

The first step in logistic regression is to create a linear combination of the input features:

$$z = wTx + b = w_1x_1 + w_2w_2 + \cdots + wmxm + b$$

Where:

- $w=(w_1,w_2,\ldots,w_m)$ is the weight vector (coefficients).
- b is the bias (intercept).
- z is called the logit or linear score.

In linear regression, z would directly be the output. However, in binary classification, we need the output to represent a probability between 0 and 1.

4.5: Decision Tree

A Decision Tree is a supervised learning algorithm used for both classification and regression tasks, but it is especially popular for binary classification problems, where the goal is to assign one of two possible labels to an input. Decision trees classify data by recursively splitting it into subsets based on feature values, creating a tree-like structure.

Decision trees are easy to understand and visualize. The decision-making process can be clearly seen from the tree structure. Unlike many other algorithms, decision trees do not require normalization or scaling of features. Decision trees can model complex, non-linear relationships between features and the target. Decision trees naturally provide insight into feature importance based on how often they are used in the splits.

4.5.1: How decision tree works:

In binary classification, the target variable y can take one of two values:

- y=1y = 1y=1 (positive class)
- y=0y = 0y=0 (negative class)

Given a dataset with features $x=(x_1,x_2,\ldots,x_m)$, the decision tree algorithm aims to split the data at each internal node based on a feature and threshold that results in the best separation of the two classes.

A decision tree consists of three types of nodes:

1. Root Node: The top node of the tree that contains the entire dataset.
2. Internal Nodes: Each internal node represents a decision based on a feature and its value.
3. Leaf Nodes: The terminal nodes where the classification decision (class label) is made.

Each internal node splits the data into two or more subsets, based on certain criteria, and this process is repeated recursively, forming a hierarchical tree structure.

At each node, the decision tree algorithm must choose a feature and a corresponding threshold value that best splits the data into subsets. The goal of the split is to increase the **purity** of the subsets, meaning each subset should contain mostly samples from a single class (either 0 or 1).

Common criteria to measure the quality of a split are:

A. Gini Impurity

Gini impurity measures how "mixed" the classes are within a node. A lower Gini impurity indicates a purer node (i.e., it contains mostly one class).

The Gini impurity at a node t is defined as:

$$Gini(t) = 1 - i = 1\sum CPi2$$

Where:

- P_i is the proportion of samples belonging to class i at node t.

- C is the number of classes (for binary classification, C=2).

The algorithm chooses the feature and threshold that minimize the weighted sum of Gini impurity across the child nodes after a split.

B. Information Gain (Entropy)

Entropy measures the amount of uncertainty or disorder in a node. A node with only one class has an entropy of 0 (pure), and a node with mixed classes has higher entropy.

The entropy at a node t is given by

$$H(t) = -i = 1\sum CPilog2(Pi)$$

The algorithm selects the split that maximizes the information gain, which is the reduction in entropy after the split.

Information Gain:

$$IG = H(parent) - \sum(NparentNchild \times H(child))$$

Where:

- H(parent)H(parent)H(parent) is the entropy of the parent node.
- H(child)H(child)H(child) is the entropy of the child nodes.
- NNN represents the number of samples in each node.

4.6: XGBoost:

XGBoost, like other boosting algorithms, builds an ensemble of decision trees. Unlike Random Forest, which trains trees independently, XGBoost trains trees sequentially. Each new tree focuses on reducing the errors (residuals) of the previous model. The algorithm optimizes its performance using gradient descent to minimize a specific loss function.

XGBoost outperforms many other algorithms due to several reasons:

- XGBoost automatically handles missing values by learning the best way to split data with missing entries, making it suitable for real-world datasets.
- By adding L1 and L2 regularization, XGBoost prevents overfitting, which is a common problem in many tree-based methods.
- XGBoost allows weights to be applied to individual classes or data points, which is useful when dealing with imbalanced datasets.

- XGBoost is optimized for parallel computation, making it faster than traditional boosting algorithms. It parallelizes tree construction by sorting features and finding the best splits more efficiently.
- XGBoost is optimized to handle sparse data (e.g., from one-hot encoding), which makes it efficient when working with high-dimensional datasets.

XGBoost is known for delivering top performance in both classification and regression tasks. The built-in regularization mechanisms help to avoid overfitting. It is highly scalable, making it suitable for large datasets. Its optimized implementation allows for fast training even on large datasets. It provides information about feature importance, which can be used for feature selection.

4.6.1: How XGBoost works

Here's a step-by-step explanation of how XGBoost works:

Step 1: Initialize with a Base Prediction

XGBoost starts with an initial prediction, usually a constant value (such as the mean of the target variable for regression or a log-odds ratio for classification).

- Let the initial prediction for all data points be y^(0).

Step 2: Compute Residuals

For each iteration, XGBoost calculates the residuals (errors) between the true labels and the current predictions. These residuals represent how far off the model's predictions are from the true values.

$$r_i = y_i - y\hat{}i(t)$$

Where:

- y_i is the true value of the target for sample i.
- y^i(t) is the prediction of the model after ttt iterations (trees).
- r_i is the residual (error) for sample i.

Step 3: Fit a New Decision Tree to the Residuals

The next decision tree is fitted to these residuals. The goal of this tree is to minimize the residual errors by learning from them. This means that the tree attempts to predict the residuals (i.e., the amount by which the previous model's predictions were wrong).

Step 4: Add the New Tree to the Model

The predictions from the new tree are combined with the previous predictions. XGBoost updates the predictions by adding a fraction of the new tree's output (controlled by the learning rate, η) to the previous predictions.

$$y^{(t+1)} = y^{(t)} + \eta f t(x)$$

Where:

- ft(x) is the prediction of the t-th tree.
- η is the learning rate, controlling how much of the new prediction is added to the old one.

Step 5: Repeat the Process

This process is repeated for a pre-specified number of trees, or until the improvement in the model's accuracy becomes negligible.

4.7: Confusion matrix

A **confusion matrix** is a table used to evaluate the performance of a classification algorithm. It provides a clear picture of how well your model is predicting actual outcomes by comparing predicted and actual class labels.

The confusion matrix is primarily used for binary classification problems but can be extended to multiclass classification.

Structure of the Confusion Matrix:

For binary classification (e.g., predicting whether a person has a disease or not), the confusion matrix is a 2x2 table:

Table: 4.7.1: binary classification for confusion matrix

	Predicted Positive	**Predicted Negative**
Actual positive	True Positive (TP)	False Negative (FN)
Actual negative	False Positive (FP)	True Negative (TN)

Components:

1. **True Positives (TP):**
    - These are cases where the model predicted "positive" (e.g., disease present) and the actual outcome is also positive.

- Example: The model correctly predicts that a patient has a disease.

2. **False Positives (FP):**
    - These are cases where the model predicted "positive," but the actual outcome is negative.
    - Example: The model predicts that a patient has a disease, but the patient does not.
3. **True Negatives (TN):**
    - These are cases where the model predicted "negative" (e.g., disease absent), and the actual outcome is also negative.
    - Example: The model correctly predicts that a patient does not have a disease.
4. **False Negatives (FN):**
    - These are cases where the model predicted "negative," but the actual outcome is positive.
    - Example: The model predicts that a patient does not have a disease, but the patient actually does.

4.7.1: How confusion matrix works

Using the values from the confusion matrix, we can calculate several important performance metrics:

Accuracy:

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

Accuracy measures the proportion of correctly predicted instances (both positives and negatives) over the total instances.

Precision (Positive Predictive Value):

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

Precision indicates how many of the predicted positive cases are actually positive. It's useful when the cost of false positives is high.

Recall (Sensitivity or True Positive Rate):

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

Recall measures how well the model is identifying actual positive cases. It's important when missing positive cases (false negatives) is costly.

F1 Score:

$$F1\ score = 2 * \frac{Precision * Recall}{\text{Precision} + \text{Recall}}$$

The F1 score is the harmonic mean of precision and recall, providing a balance between the two, especially when the class distribution is imbalanced.

4.8: Receiver Operating Characteristic curve

The Receiver Operating Characteristic (ROC) curve is a graphical representation that illustrates the performance of a binary classification model at different classification thresholds. It helps to visualize the trade-offs between sensitivity (recall) and specificity, providing insight into how well the model distinguishes between the positive and negative classes.

The ROC curve is created by adjusting the classification threshold of the model. For binary classification, the model outputs a probability (or score) for each instance, which is compared against a threshold to decide whether the instance is classified as positive or negative.

By varying this threshold from 0 to 1, you get different TPR and FPR values.

- A high threshold will classify fewer positives, leading to a lower TPR and FPR.
- A low threshold will classify more positives, leading to a higher TPR and FPR.

For each threshold, we compute the TPR and FPR. These values are then plotted as points on the graph, and connecting these points forms the ROC curve.

4.8.1: Interpreting the ROC Curve:

1. The Diagonal Line**:**
    - The diagonal line from (0, 0) to (1, 1) represents a random classifier with no discriminatory power. A model whose ROC curve is near this line is no better than random guessing.
2. The Area Under the Curve (AUC)**:**
    - The AUC is the area under the ROC curve, and it is a single number summary of the model's performance.
    - The AUC ranges from 0 to 1:

      AUC = 0.5: The model is performing no better than random guessing.

      AUC = 1**:** The model is perfect, with no false positives and no false negatives.

      AUC between 0.7 and 0.9**:** The model performs reasonably well.

      AUC < 0.7: The model's performance is suboptimal.
3. Perfect Classifier**:**
    - A perfect classifier would have a point at (0, 1), meaning TPR = 1 and FPR = 0 (i.e., all positives are correctly identified, and there are no false positives).
4. Model Comparison**:**
    - When comparing models, the one with the higher ROC curve and AUC is generally considered better. The closer the curve is to the top-left corner, the better the model is at classifying positives and negatives.

4.9: Explainable AI (Shapley Additive Explanations)

SHAP (Shapley Additive Explanations) is a framework used to explain the predictions of machine learning models by attributing the contribution of each feature to the final prediction.

SHAP helps to answer questions like:

- Which features contributed most to this specific prediction?
- How does each feature affect the model's overall prediction?

4.9.1: How SHAP works

Coalitions of Features: To compute SHAP values, the algorithm considers all possible coalitions (combinations) of features and calculates each feature's contribution by comparing the model's output with and without the feature in these coalitions. This captures the interaction between features.

Marginal Contribution: The marginal contribution of a feature is calculated by looking at how the model's prediction changes when that feature is added to a set of other features.

Shapley Values: The Shapley value for each feature is the weighted average of its marginal contributions across all possible coalitions. This ensures that each feature's contribution is computed fairly, taking into account interactions with other features.
The mathematical formula for Shapley values is:

$$\boldsymbol{\phi i = S \subseteq N \setminus \{i\} \sum n! \mid S \mid ! \, (n - \mid S \mid - 1)! \, [f(S \cup \{i\}) - f(S)]}$$

where:
N is the set of all features,
S is a subset of features excluding i,
f(S) is the model's prediction using only the features in S,
f(S∪{i}) is the prediction when feature iii is added to S.

Global Interpretations: SHAP can be used to explain the overall importance of features across the entire dataset (global interpretation). This is done by averaging the SHAP values for each feature across all predictions.

4.9.2: Why SHAP is Useful:

1. **Model Transparency**: SHAP provides a clear explanation of why the model made a specific prediction. This is especially important for complex models like neural networks or ensemble methods (e.g., random forests, gradient boosting), which are often seen as "black boxes."

2. **Fairness**: SHAP ensures a fair distribution of the feature contributions. It takes into account the interaction between features, meaning that it provides a balanced attribution even when features depend on one another.

3. **Global and Local Explanations**:
    - SHAP can explain individual predictions (local explanation), helping you understand why a particular instance was classified in a certain way.
    - It can also explain the general behavior of the model by averaging SHAP values across the entire dataset (global explanation), showing which features are most important overall.

SHAP is a powerful tool for interpreting machine learning models, based on Shapley values from game theory. It fairly distributes the prediction of a model across the features, explaining how much each feature contributes to the output. SHAP can be used with any model and provides both global (dataset-level) and local (instance-level) explanations.

Chapter-5

Result and discussion

Here we applied six machine learning models which was support vector machine, random forest, k nearest neighbor, logistic regression, decision tree, and XGBoost. We have trained the data 80% and test the data 20%. In this table we can see the performance of each model. Overall classification report is also shown here. We can see which model is doing its best to predict the Parkinson's disease or healthy.

5.1: Accuracy and classification report of each model

Table:5.1: Result

Model	Accuracy	Class	Precision	Recall	F1 score
SVM	87%	0(healthy)	75%	43%	55%
		1(PD)	89%	97%	93%
Random Forest	95%	0	100%	71%	83%
		1	94%	100%	97%
KNN	82%	0	50%	43%	46%
		1	88%	91%	89%
Logistic Regression	90%	0	100%	43%	60%
		1	89%	100%	94%
Decision Tree	87%	0	62%	71%	67%
		1	94%	91%	92%
XGBoost	92%	0	83%	71%	77%
		1	94%	97%	95%

5.2: Confusion matrix for each model

SVM

In fig:5.1 I have shown the SVM model confusion matrix.

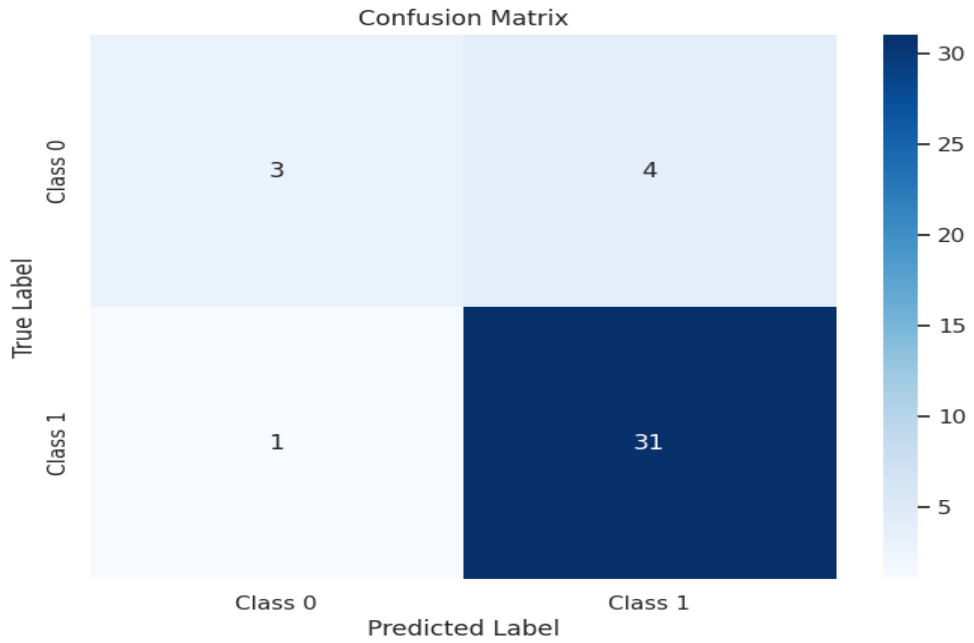


Fig 5.1: SVM

Interpretation of the values in the matrix:

- True Positives (TP): The number of data points that were correctly predicted as belonging to class 1. In this case, 31 data points were correctly classified as class 1.
- True Negatives (TN): The number of data points that were correctly predicted as belonging to class 0. In this case, 3 data points were correctly classified as class 0.
- False Positives (FP): The number of data points that were incorrectly predicted as belonging to class 1, but actually belonged to class 0. In this case, 4 data points were misclassified as class 1.
- False Negatives (FN): The number of data points that were incorrectly predicted as belonging to class 0, but actually belonged to class 1. In this case, 1 data point was misclassified as class 0.

Overall performance:

Based on the confusion matrix, the model seems to perform reasonably well, with a higher number of correct predictions (TP and TN) compared to incorrect predictions (FP and FN).

Random Forest

In fig:5.2 I have shown the Random forest model confusion matrix.

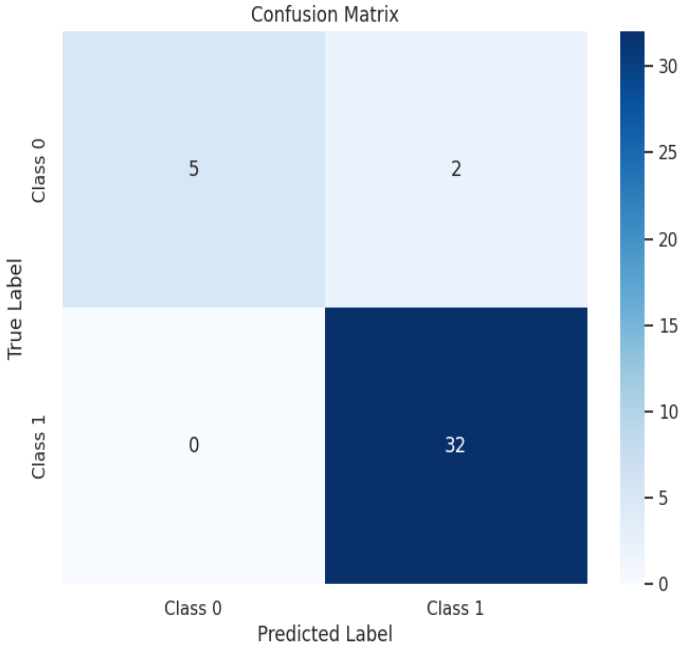


Fig 5.2: Random forest

Interpretation of the values in the matrix:

- True Positives (TP): The number of data points that were correctly predicted as belonging to class 1. In this case, 32 data points were correctly classified as class 1.
- True Negatives (TN): The number of data points that were correctly predicted as belonging to class 0. In this case, 5 data points were correctly classified as class 0.
- False Positives (FP): The number of data points that were incorrectly predicted as belonging to class 1, but actually belonged to class 0. In this case, 2 data points were misclassified as class 1.

- False Negatives (FN): The number of data points that were incorrectly predicted as belonging to class 0, but actually belonged to class 1. In this case, 0 data points were misclassified as class 0.

Overall performance:

Based on the confusion matrix, the model seems to perform very well, with a high number of correct predictions (TP and TN) and a low number of incorrect predictions (FP and FN). This indicates that the model is effectively distinguishing between the two classes.

KNN

In fig:5.3 I have shown the KNN model confusion matrix.

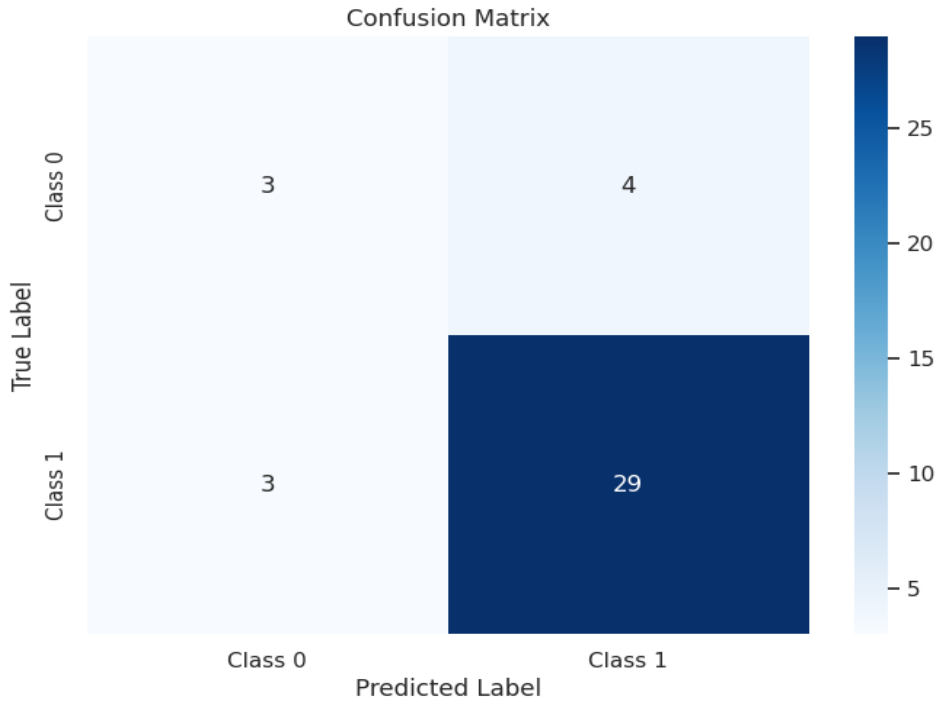


Fig 5.3: KNN

Interpretation of the values in the matrix:

- True Positives (TP): The number of data points that were correctly predicted as belonging to class 1. In this case, 29 data points were correctly classified as class 1.

- True Negatives (TN): The number of data points that were correctly predicted as belonging to class 0. In this case, 3 data points were correctly classified as class 0.
- False Positives (FP): The number of data points that were incorrectly predicted as belonging to class 1, but actually belonged to class 0. In this case, 4 data points were misclassified as class 1.
- False Negatives (FN): The number of data points that were incorrectly predicted as belonging to class 0, but actually belonged to class 1. In this case, 3 data points were misclassified as class 0.

Overall performance:

Based on the confusion matrix, the model seems to perform reasonably well, with a higher number of correct predictions (TP and TN) compared to incorrect predictions (FP and FN).

Logistic Regression

In fig:5.4 I have shown the Logistic regression model confusion matrix.

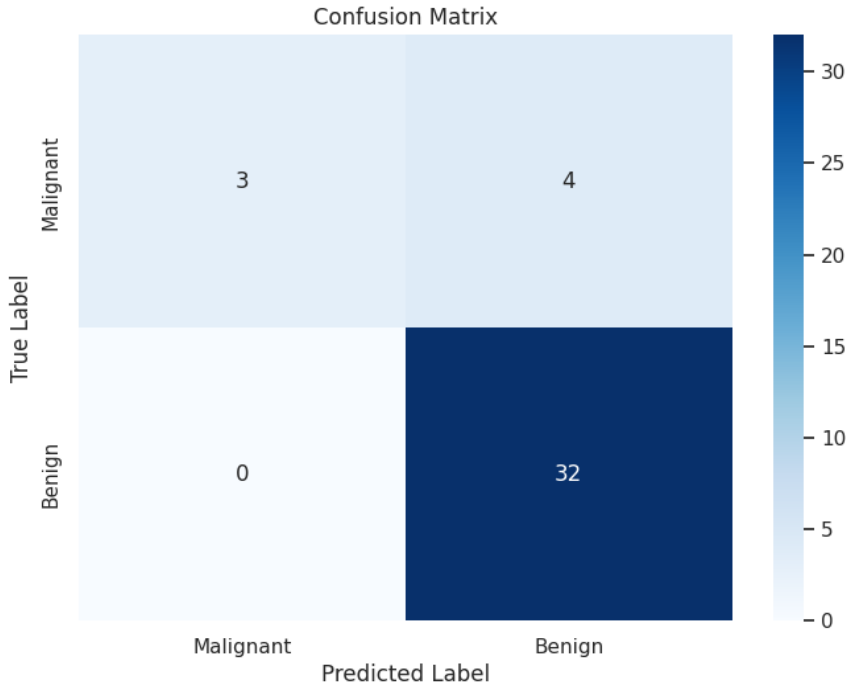


Fig 5.4: Logistic regression

Interpretation of the values:

- True Positives (TP): 32 data points were correctly predicted as class 1 (positive).
- True Negatives (TN): 3 data points were correctly predicted as class 0 (negative).
- False Positives (FP): 4 data points were incorrectly predicted as class 1 (positive), but they actually belonged to class 0 (negative).
- False Negatives (FN): 0 data points were incorrectly predicted as class 0 (negative), but they actually belonged to class 1 (positive).

Overall performance:

Based on the confusion matrix, the model seems to perform reasonably well, with a higher number of correct predictions (TP and TN) compared to incorrect predictions (FP and FN).

Decision Tree

In fig:5.5 I have shown the Decision tree model confusion matrix.

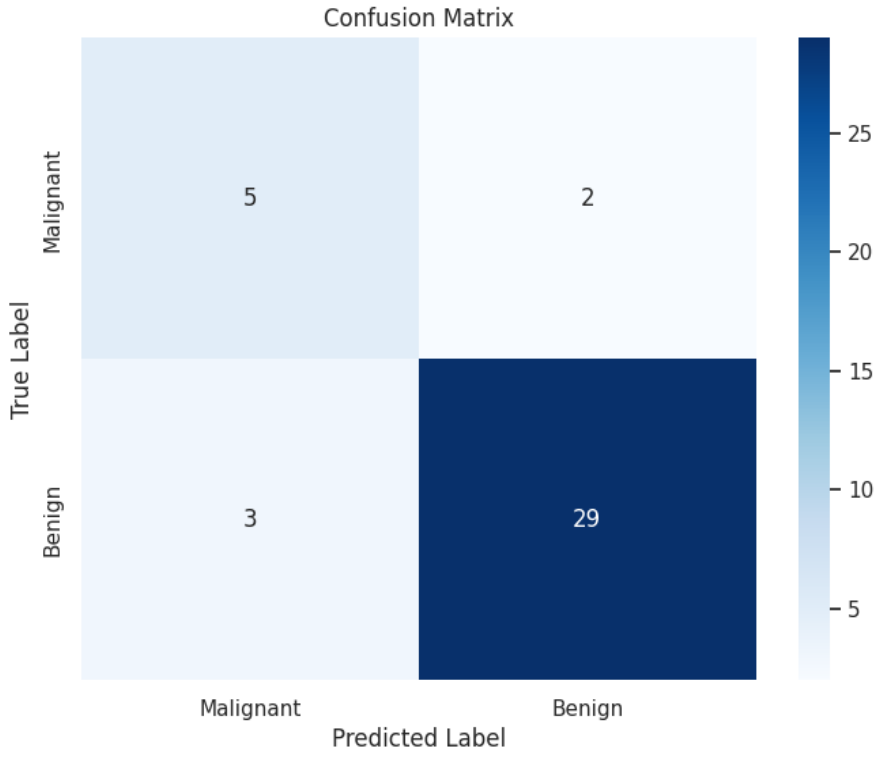


Fig:5.5: Decision Tree

Interpretation of the values:

- True Positives (TP): 29 data points were correctly predicted as class 1 (positive).
- True Negatives (TN): 5 data points were correctly predicted as class 0 (negative).
- False Positives (FP): 2 data points were incorrectly predicted as class 1 (positive), but they actually belonged to class 0 (negative).
- False Negatives (FN): 3 data points were incorrectly predicted as class 0 (negative), but they actually belonged to class 1 (positive).

Overall performance:

Based on the confusion matrix, the model seems to perform reasonably well, with a higher number of correct predictions (TP and TN) compared to incorrect predictions (FP and FN)

XGBoost

In fig:5.6 I have shown the XGBoost model confusion matrix.

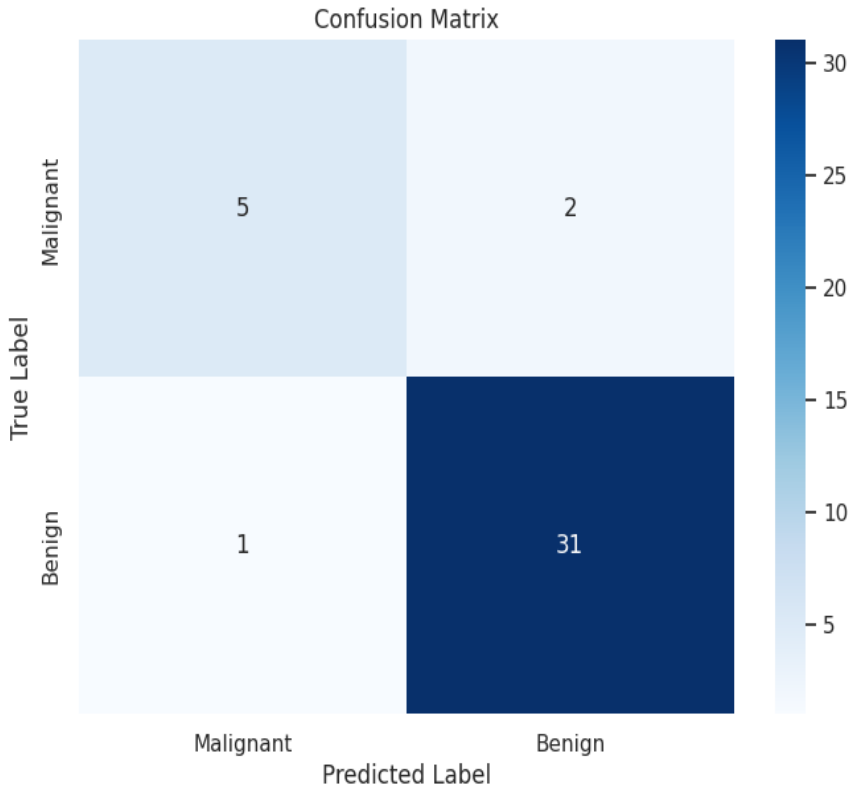


Fig 5.6:XGBoost

Interpretation of the values:

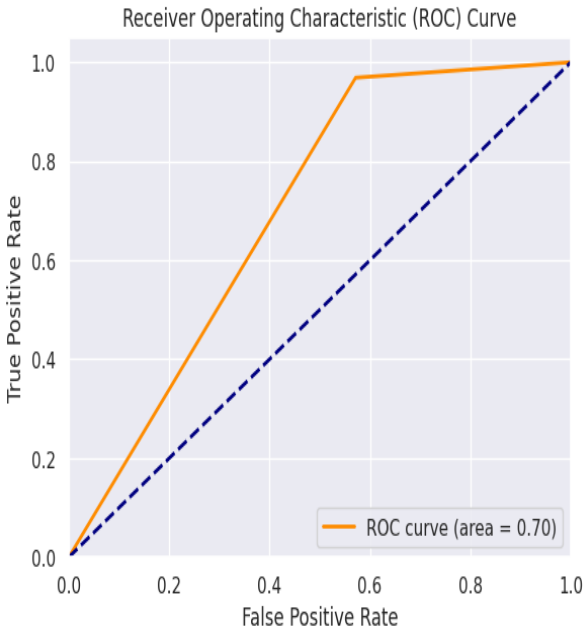
- True Positives (TP): 31 data points were correctly predicted as class 1 (positive).
- True Negatives (TN): 5 data points were correctly predicted as class 0 (negative).
- False Positives (FP): 1 data points were incorrectly predicted as class 1 (positive), but they actually belonged to class 0 (negative).
- False Negatives (FN): 1 data points were incorrectly predicted as class 0 (negative), but they actually belonged to class 1 (positive).

Overall performance:

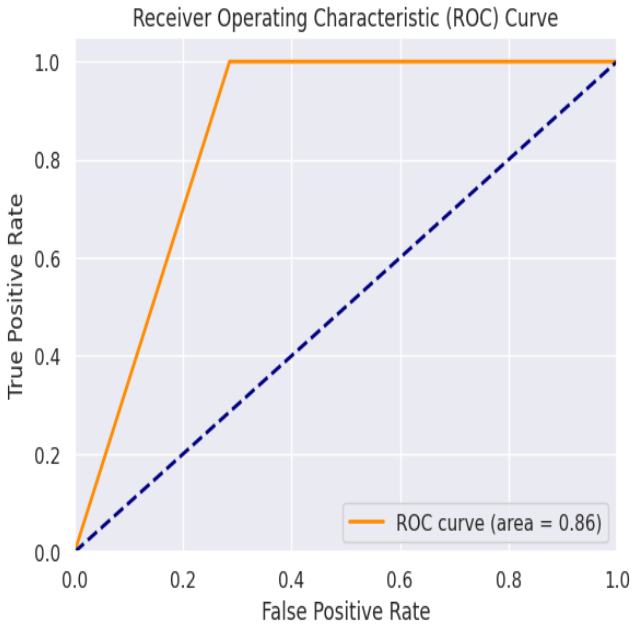
Based on the confusion matrix, the model seems to perform reasonably well, with a higher number of correct predictions (TP and TN) compared to incorrect predictions (FP and FN).

5.3 ROC for each model

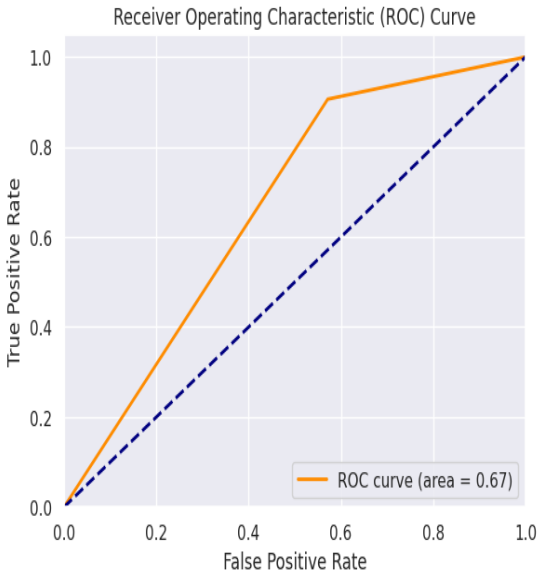
Here I have shown all the graphs of ROC curve in fig 5.7



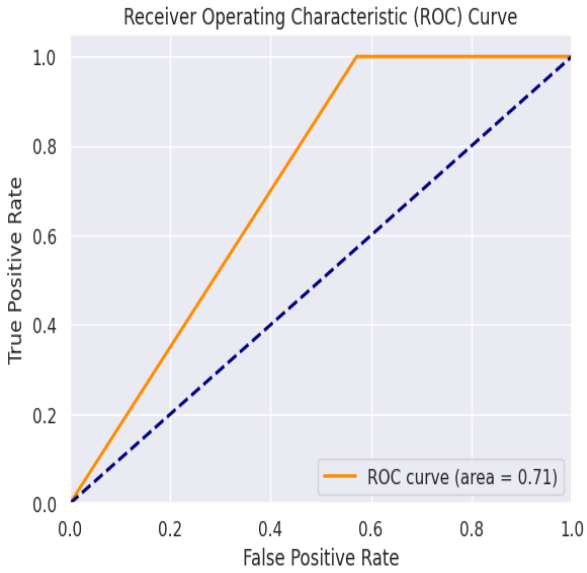
SVM



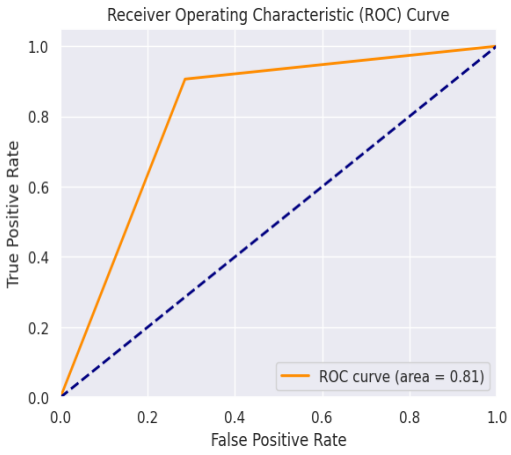
Random Forest



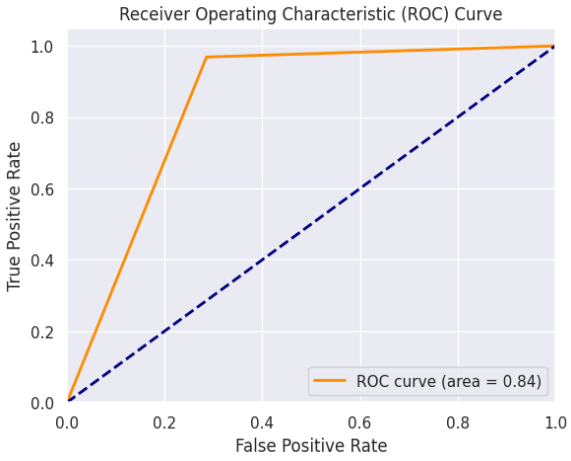
KNN



Logistic Regression



Decision tree



XGBoost

Fig 5.7: ROC curve shown for all models

After seeing the all classification result, confusion matrix and ROC curve we can say that Random forest is the best performer model for this prediction.

5.4: Apply SHAP

In fig:5.8 we have shown the SHAP summary plot

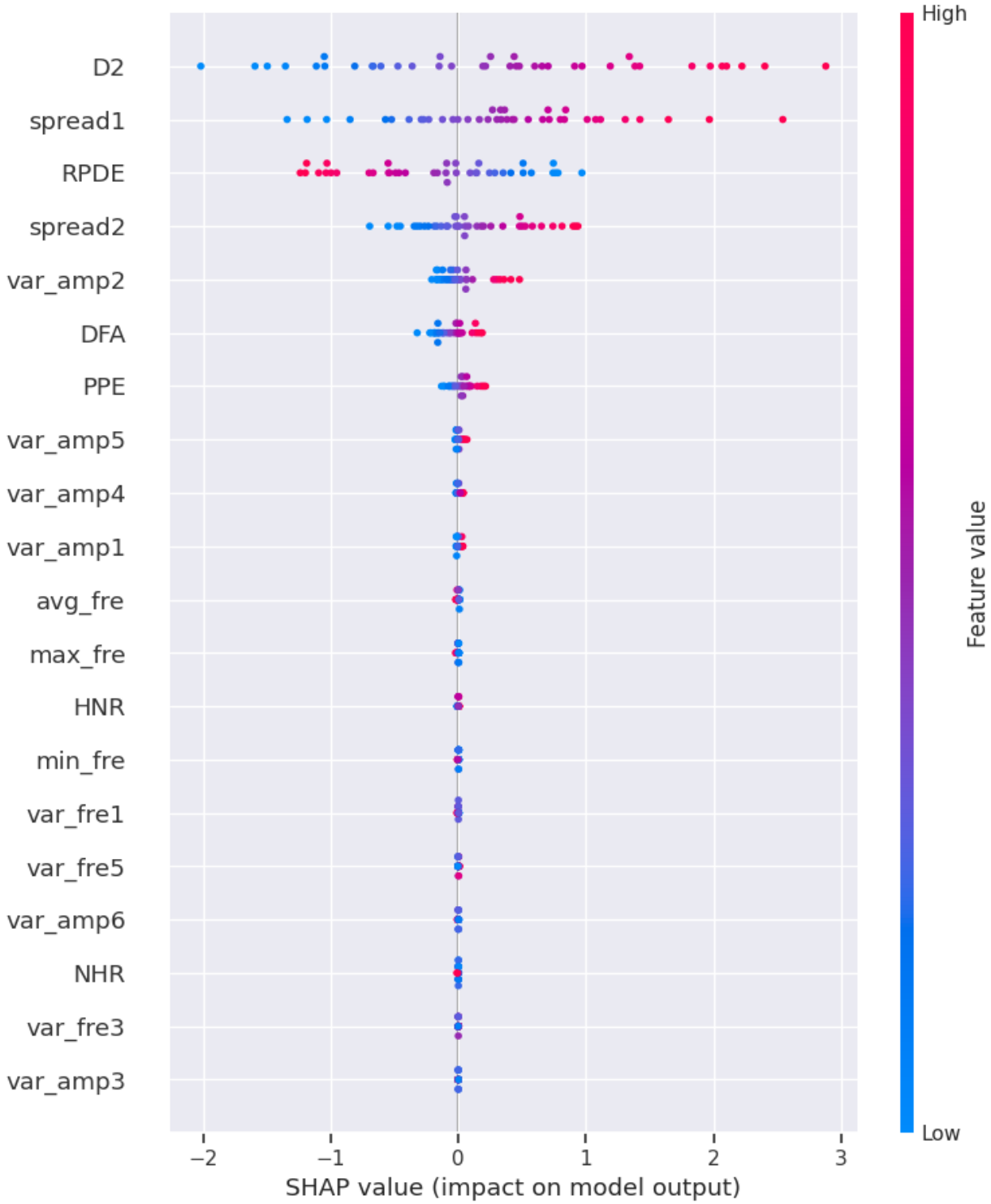


Fig 5.8: SHAP summary plot for Random Forest

The image is a SHAP summary plot. It visualizes the impact of individual feature values on the model's output for detecting Parkinson's disease.

Here's a breakdown of how to interpret this plot:

Key Points:

- Y-axis (Features**)**: The features are listed vertically, with the most important ones at the top. For example**,** D2**,** Spread1, and RPDE are the most influential features in the model.
- X-axis **(**SHAP value**)**: The SHAP value represents the impact of each feature on the model's prediction. A positive SHAP value indicates the feature increases the probability of predicting Parkinson's, while a negative value indicates it reduces the probability.
- Color**:** Each dot represents an individual data point (from your dataset). The color of the dot corresponds to the feature value, where red indicates a high feature value, and blue indicates a low feature value.

Interpretation:

1. **D2**:
    - Dots are spread across the SHAP value range, with higher feature values (in red) pushing predictions toward a higher likelihood of Parkinson's disease (positive SHAP values).
    - Lower feature values (in blue) tend to reduce the likelihood of predicting Parkinson's.
2. **Spread1**:
    - Similar to D2, high feature values (red) have a positive impact on predicting Parkinson's disease, whereas low values (blue) reduce the probability of detection.
3. **RPDE**:
    - High RPDE values (red) push predictions towards Parkinson's, while lower values (blue) have a slight negative impact on the prediction.
4. **Spread2** and **var_amp2**:
    - High values (in red) increase the likelihood of detecting Parkinson's, but the spread is more balanced, meaning these features don't always have a strong positive impact.

5. **DFA and PPE**:
    - Their dots are more concentrated near the center, indicating that these features don't have as much of a spread in SHAP values, but still influence the model.

High-impact features like D2, Spread1, and RPDE have a clear influence on the model's predictions, with high values pushing the model towards diagnosing Parkinson's.
Features lower down the plot, like HNR**,** min_fre**,** and var_fre3, have less influence as their SHAP values are centered closer to zero, indicating they have minimal impact on the model's output.
This summary plot gives a detailed view of how each feature influences the model's predictions, helping you understand the relationship between voice data and Parkinson's disease detection.

The bar plot is shown in fig 5.9

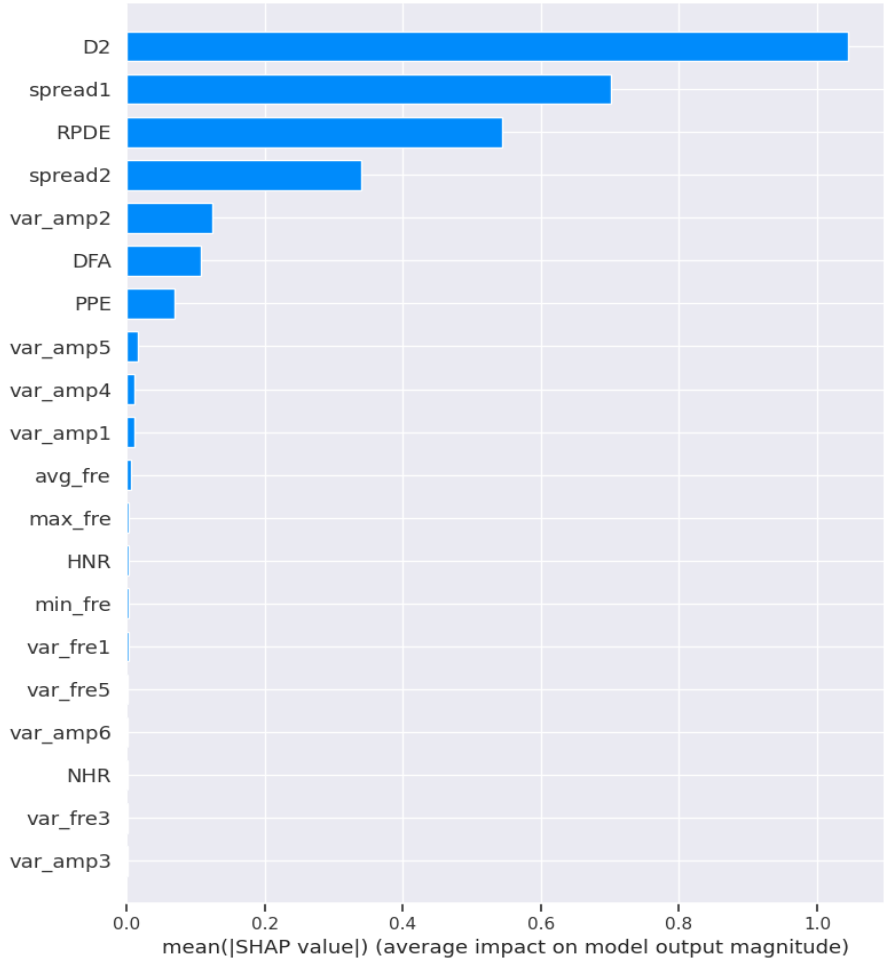


Fig 5.9: The bar plot

The chart is a SHAP (Shapley Additive Explanations) bar plot, which shows the importance of various features in predicting Parkinson's disease using a machine learning model, likely a Random Forest classifier. Each feature's impact on the model's output is represented by the mean absolute SHAP value (shown on the x-axis). Here's an explanation of the key features relevant to Parkinson's detection:

Top Important Features:

1. **D2**: This feature has the highest impact on the model's prediction, suggesting it plays a crucial role in distinguishing between individuals with and without Parkinson's disease. In the context of speech analysis, D2 often refers to a non-linear dynamic complexity measure, which could indicate irregularities in vocal performance associated with Parkinson's.
2. **Spread1**: This feature measures voice signal variation and spread. Higher values might indicate more instability in the voice, a known symptom of Parkinson's.
3. **RPDE (Recurrence Period Density Entropy)**: RPDE measures how unpredictable the vocal patterns are. Parkinson's disease can cause changes in speech regularity, making this a significant feature in detecting the condition.
4. **Spread2**: Another measure of signal spread, it further assesses variation in vocal attributes that are often affected by Parkinson's, such as pitch and tone variation.
5. **Var_amp2 (Variation in Amplitude 2)**: This refers to the variation in voice amplitude. Individuals with Parkinson's often have reduced loudness and variability in their speech, making this an important feature for the model.
6. **DFA (Detrended Fluctuation Analysis)**: DFA quantifies long-range temporal correlations in the speech signal. Changes in these correlations can be indicative of Parkinson's, which affects the coordination of speech.
7. **PPE (Pitch Period Entropy)**: PPE measures the regularity of pitch in speech. In Parkinson's disease, pitch can become more unstable or monotone, so this feature also contributes to the model's predictive power.

Less Important Features:

The lower-ranked features (such as HNR, NHR, min_fre, and var_amp6) have a much smaller impact on the model's output. While they may contribute some information, their influence on the final prediction is minimal compared to the top features.

Chapter 6

Conclusion

This project successfully demonstrates the potential of using machine learning to detect Parkinson's disease early by analyzing voice data. Out of the six models tested, the Random Forest classifier performed the best, achieving a high detection accuracy of 95%. The use of SHAP allowed us to understand which features of the voice data were most important in making predictions. This approach can help doctors remotely diagnose Parkinson's disease, making life easier for patients who have difficulty visiting hospitals. In the future, we can improve this project by using a larger and more diverse dataset to make the model more reliable, testing the system on real-world telemedicine platforms for remote diagnosis. enhancing the model to detect not just Parkinson's, but also other neurodegenerative diseases, continuously updating the model with new data to keep improving its accuracy and reliability, incorporating additional features like movement data to make the detection even more accurate and focusing on the most impactful features identified by SHAP and using them to create a more refined primary dataset for training future models, improving efficiency and predictive power.

Reference

1. **https://www.techscience.com/cmc/v71n3/46491/html#s1**
2. **https://www.sciencedirect.com/science/article/abs/pii/S0306987719314148**
3. **https://www.sciencedirect.com/science/article/pii/S1877050918316648**
4. **https://ieeexplore.ieee.org/abstract/document/9321410**
5. **https://www.sciencedirect.com/science/article/abs/pii/S1389041718308933**
6. **https://www.sciencedirect.com/science/article/abs/pii/S1959031817301136**
7. **https://www.sciencedirect.com/science/article/abs/pii/S0167739X1731631X**
8. **https://www.sciencedirect.com/science/article/abs/pii/S1389041718308933**
9. Sztahó, D., Kiss, G., & Vicsi, K., "Estimating the severity of parkinson's disease from speech using linear regression and database partitioning", In Sixteenth Annual Conference of the International Speech Communication Association, 2015.