PARKINSON'S DISEASE DETECTION

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

In

Computer Science and Engineering/Information Technology

By

LALIT YADAV (191523)

UNDER THE SUPERVISION OF

Dr. Emjee Puthooran and
Mr. Praveen Modi

to

Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report entitled **"Parkinson's Disease Detection"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat, is an authentic record of my own work carried out over a period from May 2022 to July 2023 under the supervision of **Dr. Emjee Puthooran** (Associate Professor in Electronics and Communication Department) and Co-Supervisor **Mr. Praveen Modi** (Assistant Professor (Grade 1) in CSE & IT Department.

I also authenticate that I have carried out the above-mentioned project work under the proficiency stream Data Science.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Lalit Yadav, 191523

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)
Dr. Emjee Puthooran
Associate Professor
ECE Department
Dated:

(Co-Supervisor Signature)
Mr. Praveen Modi
Assistant Professor (Grade-1)
(CSE & IT) Department
Dated:

PLAGIARISM CERTIFICATE

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ....

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: ________________________ **Department:** _______________ **Enrolment No** ________

Contact No. ___________________________ **E-mail.** __________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): ______________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

ACKNOWLEDGEMENT

With the guidance and assistance of numerous well-wishers, an endeavor over a lengthy period of time can be effective. We would like to take this time to let everyone know how much we appreciate them.

In the beginning, we'd like to express our gratitude to our supervisor, Dr. Emjee Puthooran, Associate Professor, Department of Electronics & Communication Engineering, and co-supervisor, Mr. Praveen Modi, Assistant Professor (Grade-I), Department of Computer Science & Engineering/Information Technology at Jaypee University of Information Technology (JUIT), for their invaluable support and direction throughout the project's implementation.

We wish to express our sincere thanks and gratitude to our project guide, **Dr. Emjee Puthooran**, Associate Professor, Department of Electronics and Communication Engineering, and **Mr. Praveen Modi**, Assistant Professor (Grade I), Department of Computer Science & Engineering/Information Technology at Jaypee University of Information Technology (JUIT), for the stimulating discussions, in analyzing problems associated with our project work, and for guiding us throughout the project. Project meetings were highly informative. We express our warm and sincere thanks for the encouragement, untiring guidance, and confidence she has shown in us. We are immensely indebted to her for her valuable guidance throughout our project.

TABLE OF CONTENT

ABSTRACT ---**VI**

INTRODUCTION --**1**

Introduction -- 1
Objectives---2
Language Used--3
Technical Requirements (Hardware)--------------------------------------3
Deliverables of the Major Project---4
Major Project SDLC---4
Design of Problem Statement ---4

LITERATURE SURVEY---**6**

SYSTEM DEVELOPMENT ---**8**

Requirements on Major Project --8
Use Case Diagram of the MajorProject ------------------------------------- 9
DFD Diagram of the Major Project -------------------------------------10
Implementation of the Major Project -----------------------------------11
Algorithm---15

RESULT AND EXPERIMENTAL ANALYSES -----------------------------**43**

Discussion on the Results Achieved -------------------------------------43

CONCLUSIONS ---**57**

Conclusions ---57
Applications of the Major Project---58
Limitations of the Major Project ---58
Future Work --59

REFERENCES ---**60**

ABSTRACT

Voice-based biomarkers can help diagnose symptoms of dementia such as Parkinson's disease, PD is a modern neurodegenerative disease affecting about 7 million people, people worldwide (usually adults), with about 150 thousand new scientific diagnoses performed each year . Historically, PD has been difficult to find and documents tend to focus on a few symptoms and even ignore some, depending on the scores of independent points. Due to the decline in motor manipulate which is a sign of illness, the term can be used as a means of detecting and diagnosing PD.

Common sense has meant that physicians often focus on the symptoms of PD while ignoring the other. By using independent measurement scales, the term can be used to diagnose and diagnose the disease. This paper presents evidence to support the concept of supervised classification, which can be used to diagnose individuals with diseases such as diabetes and pulmonary fibrosis. Through Linear Regression, Logistic Regression, Decision Trees, Support Vector Machine, Random Forest, XGBoost, we were able to achieve peak accuracy of 100% for diagnosing pathological conditions.

Project also uses various Evaluation Methods and Metrices such as Confusion Matrix, Classification Report, F1 - Score, Accuracy, Precision, Recall.

CHAPTER 1
INTRODUCTION

1.1) Introduction

Parkinson's disease (PD) shows loss of life of dopaminergic neurons in substantia nigra pars compacta in the midbrain. These neurodegeneration modifications end up with numbers and symptoms that include communication problems, voice adjustment, and allergies. Dysarthria is also seen in PD sufferers; its miles are characterized by weakness, paralysis, and a lack of communication within the motor vehicle: affecting breathing, crying, speech, and prosody.. The exact reason of Parkinson's disease is unknown, but it is concept to involve complex interactions between genetics, biology, and the environment, which not only leads to the ability heterogeneity of PD signs and symptoms, however also to their charge of progression through the years. This brings uncertainty to sufferers' destiny first-class of existence, and it also brings challenges to treatment trials in determining successful endpoints. More and more reports of changes in peripheral immune machine function in sufferers with Parkinson's sickness. In latest years, there's evidence that a protein referred to as α-synuclein, the enteric fearful machine, and the intestinal brain axis are associated with the etiology of Parkinson's disorder, which suggests that Parkinson's ailment can also even start from the outer edge. These studies can also provide an possibility to pick out markers that are expecting the progression of Parkinson's sickness. Since signs and the disorder path they vary, PD is usually now not available for decades. Therefore, there may be a need for more sophisticated diagnostic equipment to diagnose PD because, as the disease progresses, additional symptoms appear that make PD harder to deal with. The main shortcomings of PD speech are loss of depth, tone of voice and voice, reduction of pressure, hallucinations, short speech speeds, dynamic charging, blurred consonants, and a violent and breathing voice (dysphonia). The wide range of voice-related features promises a powerful tool because voice recording is non-invasive and can be done without difficulty with mobile gadgets. This business is evaluating the effectiveness of the use of controlled type algorithms, as well as Logistic Regression, Vector Support Machines, Decision Trees, Random Forest, XGBoost, Neural Network for better identification of people with the disease. Our 100% duration (in the database that has been used) provided using the model diagnostic gadget exceeds the accuracy of non-specialist medical diagnostics (73.8%) and the accuracy between dynamic pathologists (79.6% without follow-up, 83.9% after followup) and authentic pathological autopsy tests. .

1.2) Objective of the Minor Project

PD is difficult to detect early because of the first hidden symptoms. There is a huge burden on patients and the resilience system due to delays in diagnosis. The problem in early PD analysis has encouraged researchers to expand the test gear calculations in automated algorithms to separate healthy controls in people with PD. This binary diagnosis specializes in one-step verification of visible biomarkers in disease identification and control; now it does not offer the kind of alternative analysis where the version may also distinguish PD between a host of symptoms that manifest symptoms such as PD (e.g., Lewy-Body Dementia, Essential Tremor). Modern studies are a promising first step toward the long-term goal of introducing a preferred set of physician guidelines in screening patients with PD.

1.3) Language Used

Programming Language used is **PYTHON**

1.4) Technical Requirements (Hardware)

As we will be working with Jupiter Notebook (Anaconda Distribution) The minimum system requirements are:

- Memory: 4 GB
- Free Storage: 2 GB
- Screen Resolution: 1200 x 800
- OS: Windows 7/8/8.1/10 (64-bit)

The recommended system requirements are

- Memory: 8 GB RAM
- Free Storage: 4 GB (SSD Recommended)
- Screen Resolution: 1920 x 800
- OS: Windows 10 (64-bit)
- CPU: Intel Core i5-8400 3.0 GHz or better.

Requirement for doing Analysis. The Algorithm, Libraries used:

- NumPy, Matplotlib, Seaborn.
- Pandas, Scikit Learn, XGBoost.
- Linear Regression, Logistic Regression, Decision Trees.
- Support Vector Machine, Random Forest.
- XGBoost.

1.5) Deliverables of the Major Project

The project will predict whether the person is or is not having PD. Using the given data set we will analyze data using Machine Learning Algorithms (Linear_Regression, Logistic_Regression, Decision_Trees, Support_Vector_Machine, Random_Forest, XGBoost.) Our goal is to attain a 100% accuracy Model. We will also provide a Confusion Matrix, Classification Report, F1 - Score, Accuracy, Precision, Recall. For the better conclusion of our models.

1.6) Major Project SDLC

Feasibility Study on Major Project:

This project evaluates the effectiveness of using controlled classification algorithms, such as Logistic Regression, Vector Support Machines, Decision Trees, Random Forest, XGBoost, Neural Network to accurately identify people with the disease. Our 100% accuracy (in the database we used) provided by machine learning models exceeds the accuracy of clinical diagnostic tests for non-specialists (73.8%) and the accuracy among movement therapists (79.6% without follow-up, 83.9% after follow-up) with autopsy as a basic fact.

1.7) Design of Problem Statement

Parkinson’s disease is a neurological disorder that affects the motion of the body, the disease can lead to bad posture, shaky hands, stiffness in muscles, and hallucinations. The detection of disease is difficult in young people and disease is common in elderly groups. Medical statistics has developed a huge scale of quantity from distinct clinical regions such as fitness care services. To handle these statistics and accomplish insights from these statistics there is a want of Big Data evaluation through Machine Learning knowledge of that goal to solve various medical and clinical hassle. Already, many of the studies show that ML knowledge of algorithms has received meaningfully excessive overall performance in classification-primarily based medical troubles. However, supervised ML strategies are one of the handiest techniques for the studies community and actual-life programs in clinical fields. This work's important objective is to enhance the detection and analysis strategies of Parkinson's disease treatment. Parkinson's disease can not be cured; however, medicinal drugs can help manipulate your signs and symptoms, regularly dramatically. So, if it detects within the early stage, the cost of drugs will reduce. Therefore, our observation may be playing a vital position in detecting Parkinson's disease with Machine Learning algorithms. the detection of disease is important therefore the purpose of our major project is to develop a prediction model of Parkinson’s disease. We are going to achieve our goal with the help of the following parameters.

CHAPTER 2
LITERATURE SURVEY

1) Ali L., Khan S. U. (2019) [4] According to new research, patients with Parkinson's disease (PD) exhibit a variety of symptoms such as tight muscles, delayed movement, and tremor. However, dysphonia—changes in speech and articulation—is regarded as the most important precursor, as 90% of Parkinson's disease patients have speech difficulties. PD detection has made use of a variety of speech data formats. The many types of speech data contain several sorts of voice or speech samples for each subject, such as speech recordings of numbers, words, some short sentences, and vowel phonations. Previous research has found that numerical samples outperform vowel samples in terms of performance. As a result, it is currently unclear which kind of speech samples provide additional information about Parkinson's disease.

2) S. Aich, Hee-Cheol Kim (2019) [3] Among neurological diseases, Parkinson's disease is the second most common disease affecting the elderly over 65 years of age. It is also mentioned that the number of people affected by Parkinson's disease will continue to increase until 2050, which will be an increasing problem for many developed countries because of the high cost of health care for this disease. Parkinson's disease (PD) is one of the neurological disorders that directly affects brain cells and manifests itself in the form of movement, voice and other cognitive disorders. In recent years, researchers have worked to detect and monitor Parkinson's disease using data from speech analysis and gait analysis. Machine learning and artificial intelligence techniques are gaining popularity because these techniques are able to automate the pattern recognition process with high accuracy, but no one has yet compared the performance metrics using different feature sets by applying nonlinear and linear classification approaches based on voice data. Therefore,

in this paper, we proposed a new approach by comparing the performance metrics using different feature sets such as genetic algorithm-based feature sets as well as principal component analysis-based feature reduction techniques to select feature sets. 3) Md Nafiul Alam, Amanmeet Garg (2017)

[3] Patients with Parkinson's disease (PD) regularly show abnormal gait patterns. Automated discrimination between abnormal and normal gait can serve as a potential tool for early diagnosis and monitoring the effect of PD treatment. The aim of the current study is to distinguish PD patients from healthy controls using features derived from vertical ground reaction force (VGRF) data during walking at a normal pace. The current work represents a comprehensive study that demonstrates the effectiveness of various machine learning classifiers to develop an accurate prediction system. The selection of meaningful features based on sequential selection of forward features, swing time, stride time variability, and center of pressure features facilitated the successful classification of control and PD gaits. Support vector machine (SVM), K-nearest neighbor (KNN), random forest, and decision trees were used to build the prediction model. We found that SVM with cubic kernel outperformed the other classifiers with accuracy of 93.6%, sensitivity of 93.1%, and specificity of 94.1%.

CHAPTER 3
SYSTEM DEVELOPMENT

3.1) Requirements on Major Project

3.1.1) Functional Requirements

- Anaconda Distribution's (Jupyter Notebook)
- Programming language – Python
- Microsoft Windows 10
- Machine Learning Libraries: NumPy, Matplotlib, Seaborn, Pandas, Scikit Learn and XGBoost.
- Machine Learning Algorithms Used: Linear_Regression, Logistic_Regression, Decision_Trees, Support_Vector_Machine, Random_Forest, XGBooster.
- Evaluation Methods and Metrics: Confusion Matrix, Classification, Accuracy, Precision, and Recall.

3.1.2) Non-Functional Requirements

- ATLEAST: 2 GB memory
- Intel 8^{th} generation higher CPU's
- Nvidia 1650 min GPU

3.2) Use Case Diagram of the MajorProject

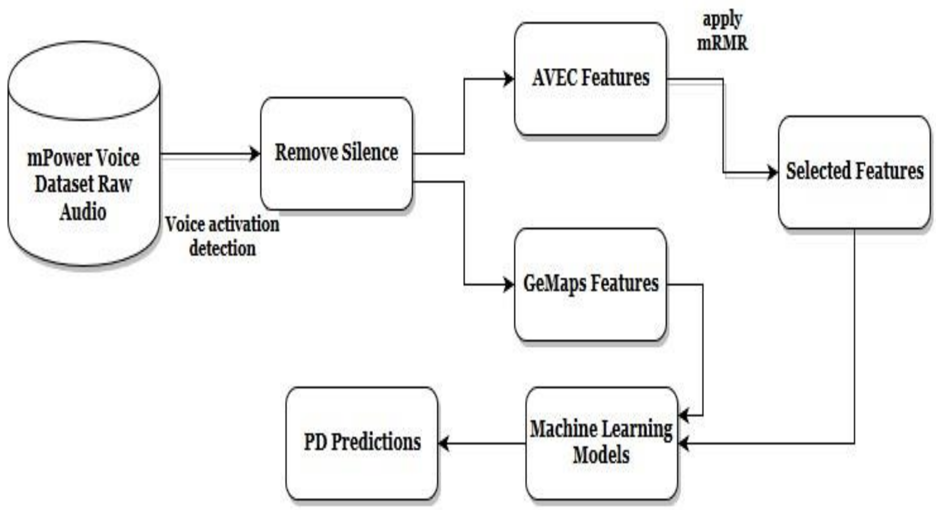


Fig1: Use Case Diagram

3.3) DFD Diagram of the Major Project

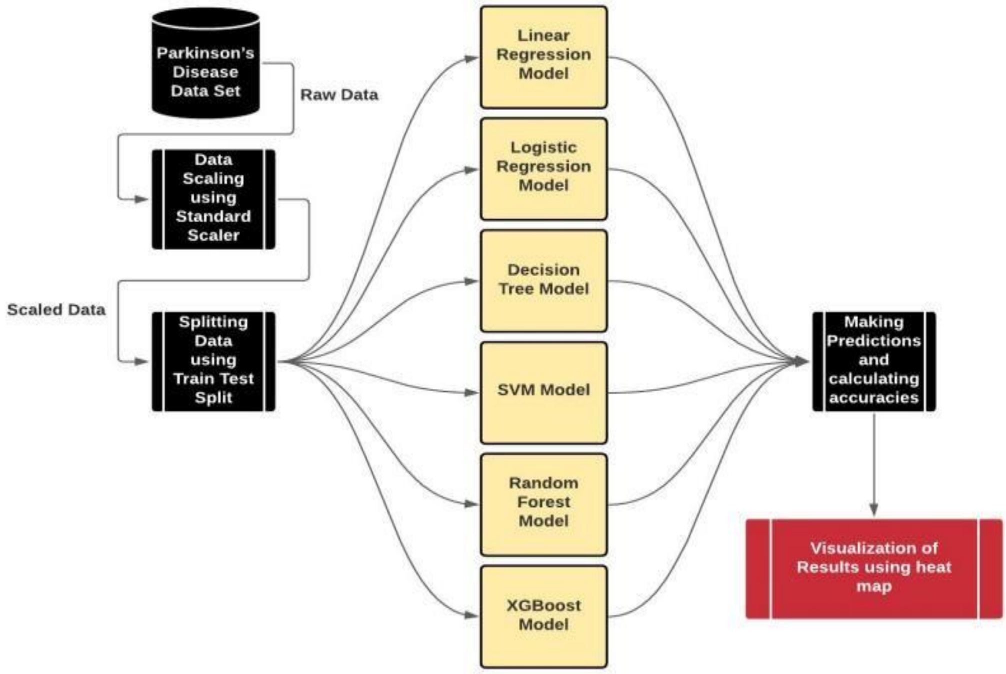


Fig2: DFD Diagram

3.4) Implementation of the Major Project

3.4.1) Dataset Used in the Major Project

This database contains a large number of biomedical voice measurements (Acoustic Analysis of voice) from 31, 23 people with Parkinson's disorder (PD). Each column on the desk is a measure of a certain word, and each line corresponds exactly to one of the 195 words recorded for those people (column "calls"). An important mathematical goal is to differentiate healthy people from people with PD, according to the "status" column of almost zero for good and 1 for PD. Information is in ASCII CSV format. CSV document lines include an example corresponding to a single voice recording. There are about six recordings of the affected person, the affected person's call is found in the first column. The dataset is created by means of 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 unique have a look at published characteristic extraction techniques for trendy voice problems.

	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109
1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01394
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966

MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	...	Shimmer:DDA	NHR	HNR	status	RPDE	DFA	spread1	spread2	D2	PPE
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	0.284654
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	0.368674
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	0.332634
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	0.368975
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	0.410335

3.4.2) Date Set Features

Types of Data Set

The Parkinson's Disease Data Set is a multivariate data set with 179 instances. The total no. of attributes in the given data set which is used for modeling is 23 the data set was denoted on 26th June 2008 this data does not contain any missing values and is used for classification.

Number of Attributes, fields, description of the dataset

The Parkinson's disease database will help us to figure out whether the respective target is or is not having the disease its a multivariate data set. The database is the voice samples which have been accumulated from 31 people out of which 23 are having the disease this data set is composed of the range of biomedical voice measurements and each column describes a particular voice measure and each row corresponds to one of the 195 voice recordings from these individuals. We with the help of machine learning techniques Machine Learning Techniques Will be creating a model which can be 100% accurate in the description of the patient the model will analyze the data in the given data set and will protect whether the patient is or is not having the Parkinson disease.

Qualification Information:

Matrix column entries (attributes):

- name - ASCII title name and recording number
- MDVP: Fo (Hz) - Basic voice frequency
- MDVP: Fhi (Hz) - The frequency of the basic voice
- MDVP: Flo (Hz) - Basic voice frequency
- MDVP: Jitter (%), MDVP: Jitter (Abs), MDVP: RAP, MDVP: PPQ, Jitter: DDP - Several

estimates of basic frequency variability

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

APQ5, MDVP: APQ,

Shimmer: DDA - A few steps for size variation

- NHR, HNR - Two levels of sound measurement and tone components in voice
- condition - The health condition of the subject (one) - Parkinson's (zero) - is healthy
- RPDE, D2 - Two steps of two indirect flexible complex
- DFA - Signal fractal measurement Exponent
- spread1, spread2, PPE - Three indirect measurements of basic frequency variation.

Algorithm / Pseudocode of the Project Problem with the screenshots of various stages of the project

1.) DATA CLEANING:

Before starting with the analysis part, we win modify our data for proper and correct analysis for that we will check the unique values in the name and status column after that we will remove the name variable from our data set because the name variable is not important for making predictions. After that, we have seeing the shape of our data set when the value of status is one and 0. After seeing the shape we will see that whether our data is having missing values or not if the data is having missing values, we will have two do appropriate calculations to remove it but, in our case, data didn't have any missing values therefore it was ready for further analysis.

Some screenshots of the process have been attached below:

```
[ ] len(df.name.unique())

195

[ ] len(df.status.unique())

2

[ ] #we dont need the 'name' variable for predictions
    df = df.drop(['name'], axis = 1)

[ ] df.columns
```

```
[ ] df.shape

(195, 23)

df[df['status']==1].shape

(147, 23)

[ ] df[df['status']==0].shape

(48, 23)
```

2.) Standard Scaling:

After the data cleaning process is completed, our next step is 2 standard scale our data. For making a model we will have to do a train test split and for that, it is important that data has been successfully transformed into X and Y arrays. We will drop the status column from the X array and transform X using a standard scaler meanwhile we will create another array that will only contain the value of status. Now the data is completely ready for train test split and further analysis. It measures features by subtracting meaning and measuring to unit variations.

A typical sample school x is calculated as:

z = (x - u) / s

where u is the method of training samples, and s the standard deviation of training samples.

Two of the most popular ways to measure numerical data before modeling are customization and configuration. The practice of measuring different inputs separately at a range of 0-1, which is a floating point value range when we have high accuracy. The measurement is different for each input by subtracting the meaning (called centering) and dividing it by standard deviation to change the distribution so that it has zero meaning and standard deviation for each.

MinMaxScaler (feature_range = (0, 1)) will convert each value into a column equally within the range [0,1]. Use this as the first scale option to change the feature, as it will maintain the database status (no distortion).

StandardScaler () will convert each value in a column to a range of 0 with a standard deviation of 1, that is, each value will be made normal by subtracting and dividing by standard deviation. Use StandardScaler if you know that data distribution is common.

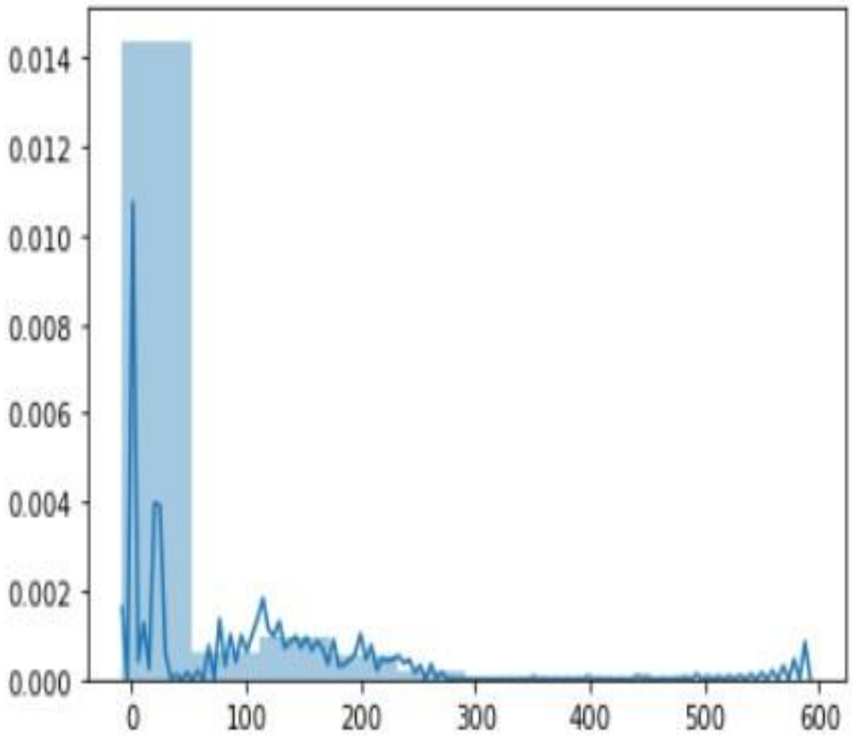
```
[ ] #scaling the inputs
    from sklearn.preprocessing import StandardScaler
```

```
[ ] x = df.drop(['status'], axis = 1)
    y = df['status']
```

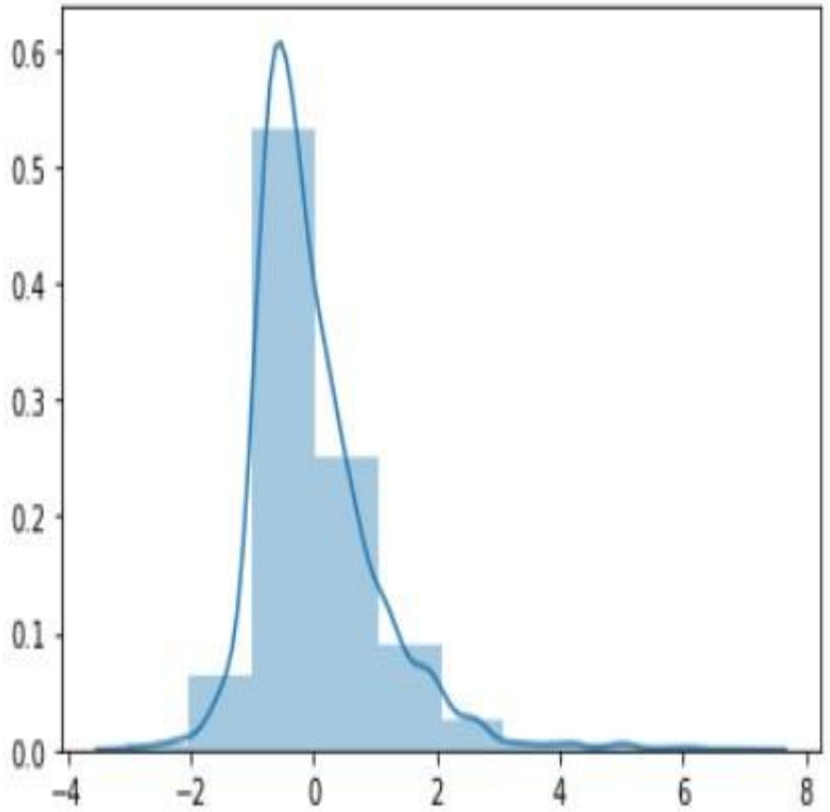
```
[ ] std = StandardScaler()
    X = np.array(std.fit_transform(x))
```

```
[ ] X
```

```
array([[-0.82929965, -0.43616456, -0.95203729, ...,  0.48047686,
        -0.21053082,  0.86888575],
       [-0.77097169, -0.53097409, -0.05772056, ...,  1.31118546,
         0.27507712,  1.80360503],
       [-0.90947638, -0.7231683 , -0.10987483, ...,  1.01768236,
        -0.10362861,  1.40266141],
       ...,
       [ 0.49557839,  0.47010361, -0.96839309, ..., -0.81807931,
         0.78033848, -0.83241014],
       [ 1.07876114,  2.19004398, -0.95417967, ..., -0.22906571,
        -0.63700298, -0.92610456],
       [ 1.45481664,  0.69224632, -0.88348115, ..., -0.43085284,
         0.45480231, -0.64505466]])
```


0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0
100
200
300
400
500
600

BEFORE

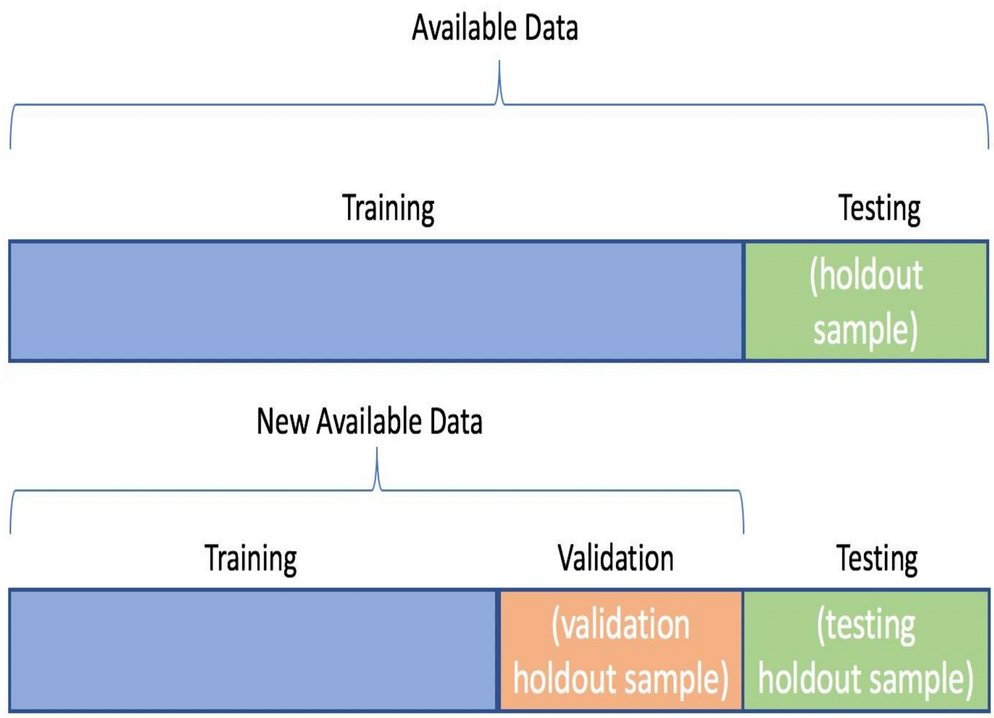


AFTER

3.) Train test split:

In order to obtain effective model calculation in machine learning, it is important to train and build an algorithm that can assist in data prediction. The data provided is usually categorized into data sets and reused for training and testing purposes which are usually training, validation, and test sets. The method is used to measure the overall performance of ML algorithms while it may be used to speculate on unspecified facts to teach the model. It is a fast and easy way to do it, the results of which will allow you to test the performance of ML algorithms to your predictive modeling complexity. Although easy to use and translate, there are instances when the process should not be used now, including when you have a small database and situations where additional configuration is required, including when used in class and the database is uneven. . The model was first included in the training data after the model was trained using a supervised learning method. The current model or model we are developing is used with a set of training data and will produce a result based on the result we can predict whether the model successfully predicts prices or not.

The embedded model is useful for predicting a confirmation data set that provides an unbiased evaluation of the model at the end of the data set provides an unbiased evaluation of the final model of the training data set. The separation of the train test will result in two trained databases and the test train data will be used to match the machine learning model and the test data set will be used for testing purposes. The average train ride to the test is 80% train and 30% inspection.



In our model, we will use a trains test split with a test size of 0.2 the method will result in two X_train, X_test, Y_train, and Y_test these values would be then stored in an array the value stored now would be used for further analysis.

```
[ ] from sklearn.model_selection import train_test_split
```

```
X_train, X_test, Y_train, Y_test = train_test_split(x,y,test_size=0.2,ra
```

```
[ ] print(X_train.shape)
    print(X_test.shape)
    print(Y_train.shape)
    print(Y_test.shape)
```

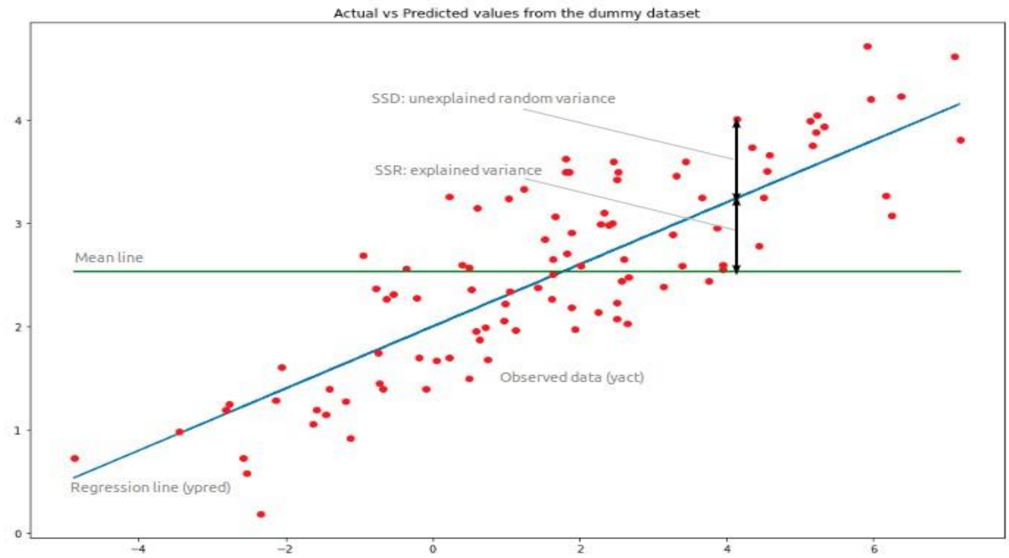
```
(156, 22)
(39, 22)
(156,)
(39,)
```

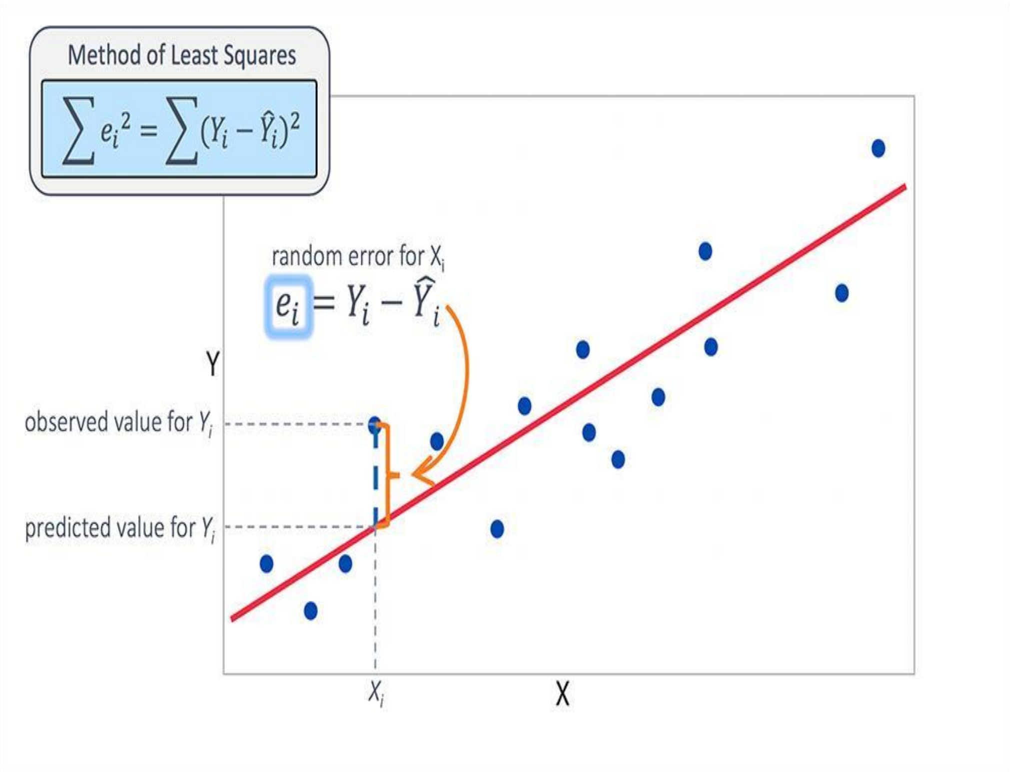
```
[ ] X_train = np.array(X_train)
```

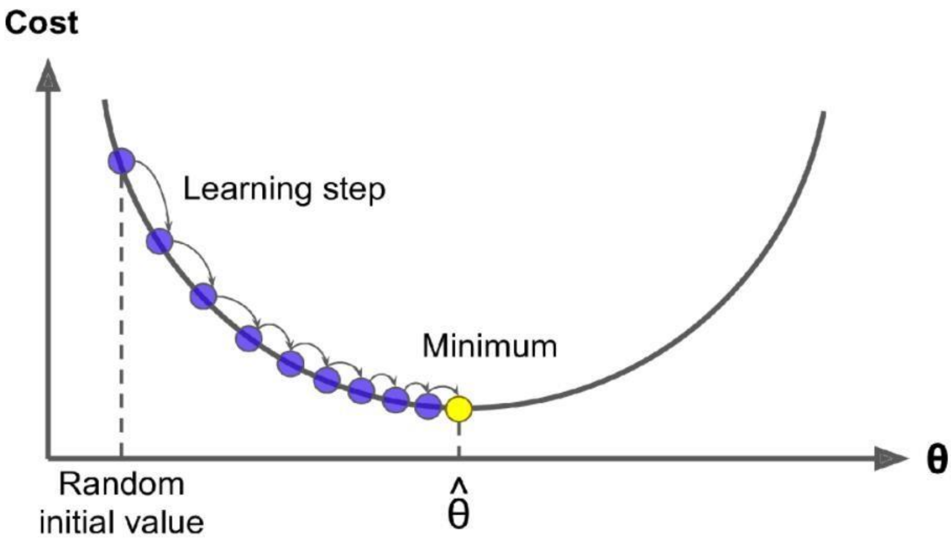
4.) Linear Regression Model:

The linear regression is helping mode model relationship between one or more than one variable B1 is an independent variable while the other is a dependent variable. This model is used to find relationships between two or more continuous variables.

The General equation is y = mx + c where x = independent variable y = dependent variable and m = slope between them and c = constant. The model will fit a straight line that minimizes the difference between predicted and original output. The implementation of linear regression is much easier than any other model and generally provides good accuracy scores. The most common and the best linear regression method is the least square method. the prediction made by the linear model can be easily fitted into the best model using gradient descent algo. Gradient descent algo is an optimization algorithm which after changing the value of coefficients provides the optimal solution. Linear Regression is easy to use and not so difficult to translate the output coefficients. If the connection between independent and dependent variants has a dating line, this algorithm is ideal for use due to its very small complexity compared to different algorithms. If the connection between the independent and dependent variables has a linear relationship, this set of rules is special to apply due to its very small complexity compared to other algorithms. On the other hand, within the line regression paths exits can have significant effects on retrospect, and the limits are straightforward in this process. In contrast, linear regression assumes a linear relationship between dependent variations and bias. Thus it is thought that there is an immediate courtship between them. It assumes independence between attributes, but the regression of the line is also reflected in the relationship between the definition of dependent variation and independent variation. Just as the definition is not always the complete definition of a single variance, the linear regression is not always the full definition of the relationship between variables.

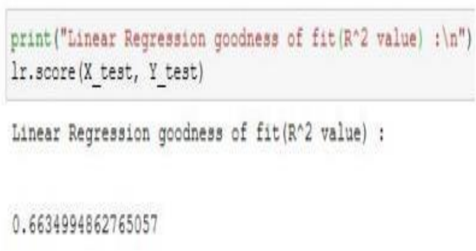

Actual vs Predicted values from the dummy dataset
SSD: unexplained random variance
SSR: explained variance
Mean line
Observed data (yact)
Regression line (ypred)


Method of Least Squares
$\sum e_i^2 = \sum (Y_i - \hat{Y}_i)^2$
random error for X_i
$e_i = Y_i - \hat{Y}_i$
Y
observed value for Y_i
predicted value for Y_i
X_i
X



In our model we have fitted X_train and Y_train in linear regression model after that we have applied gradient descent in order to find the best fit. followed, we have predicted the accuracy, and the accuracy we caught was 66%.

The implementation figures with the confusion matrix and accuracy score are been mentioned below:



```
print("Linear Regression goodness of fit(R^2 value) :\n")
lr.score(X_test, Y_test)
```

```
Linear Regression goodness of fit(R^2 value) :

0.6634994862765057
```

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
```

```
lr.fit(X_train, Y_train)
```

```
LinearRegression()
```

```
predictions_model1 = lr.predict(X_test)
predictions_model1
```

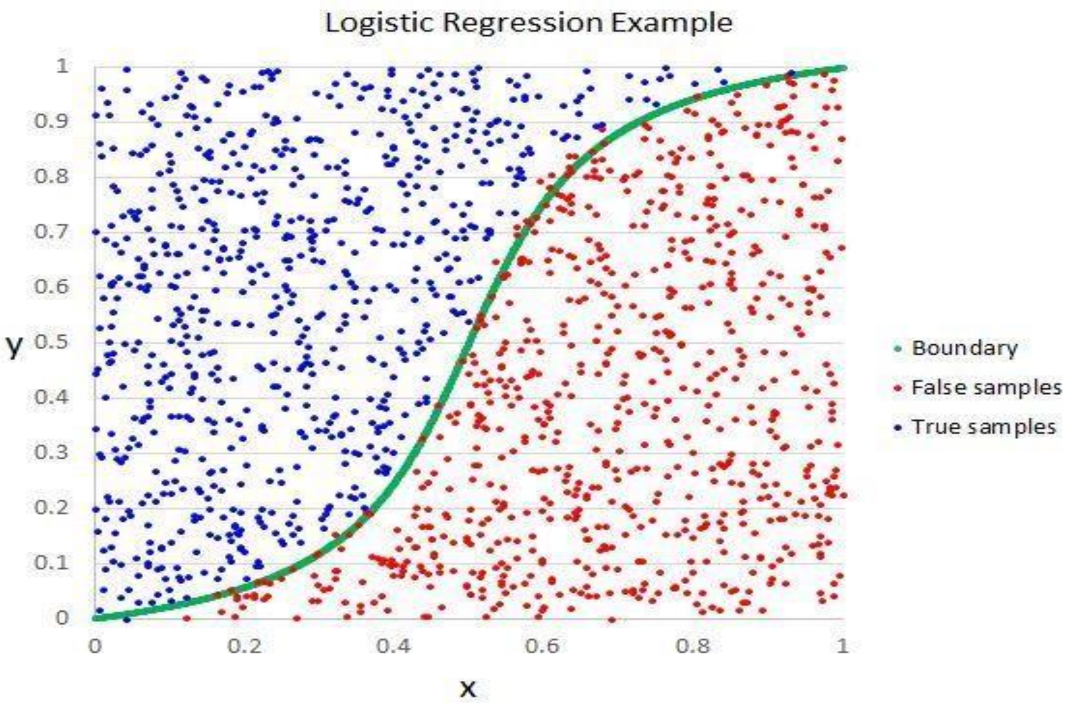
```
array([0.51464502, 0.92291805, 1.08895196, 0.93072429, 0.39722818,
       0.96450257, 0.51665718, 0.73428536, 0.96425774, 0.89648295,
       0.54933081, 1.04009737, 1.02683814, 1.02090193, 0.52986194,
       0.21476581, 1.07171961, 0.93419406, 0.55144217, 0.87538823,
       1.17667136, 0.53241189, 0.40486539, 0.77032915, 0.94973256,
       0.79196182, 0.95604616, 1.05056541, 0.02141046, 0.08635714,
       0.86404109, 0.73424635, 1.07673007, 0.16402396, 0.26651789,
       0.70248981, 0.5602687 , 0.96492844, 0.09659312])
```

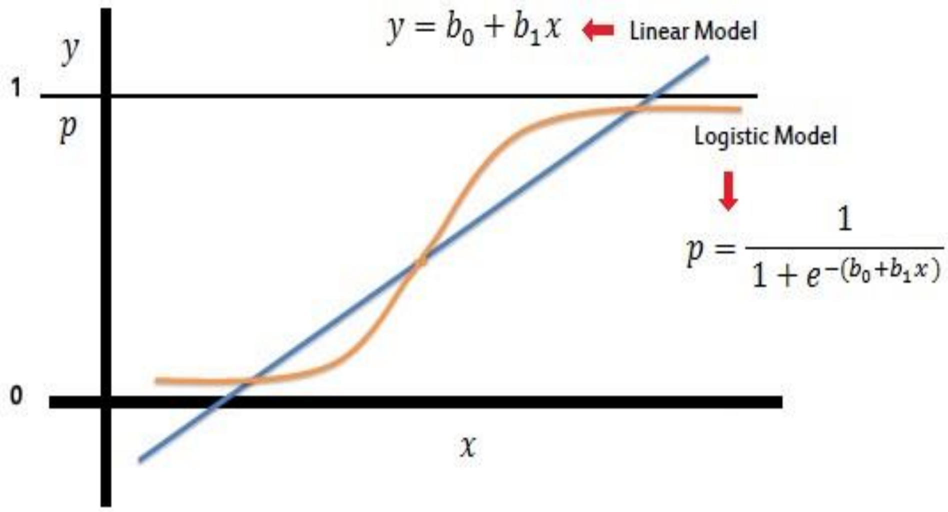
```
for i,j in enumerate(predictions_model1):
    if(j<0.5):
        predictions_model1[i]=0
    else:
        predictions_model1[i]=1
print(predictions_model1)
predictions_model1.shape
```

```
[1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 0. 1.
 1. 1. 1. 1. 0. 0. 1. 1. 1. 0. 0. 1. 1. 1. 0.]
```

5.) Logistic Regression Model:

The next model that we're going to use is the logistic regression model which is like linear regression model is another supervised learning technique. The logistic regression model generally gives the solution in a binary yes or no or true or false situation. The main difference between both models is that linear regression provides a continuous prediction whereas logistic regression provides predictions in categories. Logistic regression is simpler to put into effect, interpret, and very efficient to teach, It makes no assumptions approximately distributions of classes in characteristic space, It can without problems increase to multiple training(multinomial regression) and a herbal probabilistic view of sophistication predictions, It, not handiest presents a measure of the way appropriate a predictor(coefficient size)is, however additionally its path of association (positive or negative), The good accuracy of many mathematical sets is simple and playable when the database is subdivided along the line and the Logistic decline is slightly inclined to overequilibrium but may extend to large-scale datasets.. One can also don't forget Regularization (L1 and L2) strategies to avoid over-fitting in those scenarios. The logistic regression ML model is used for prediction of various diseases including heart disease, neurological disorders prediction of GDP and economy.

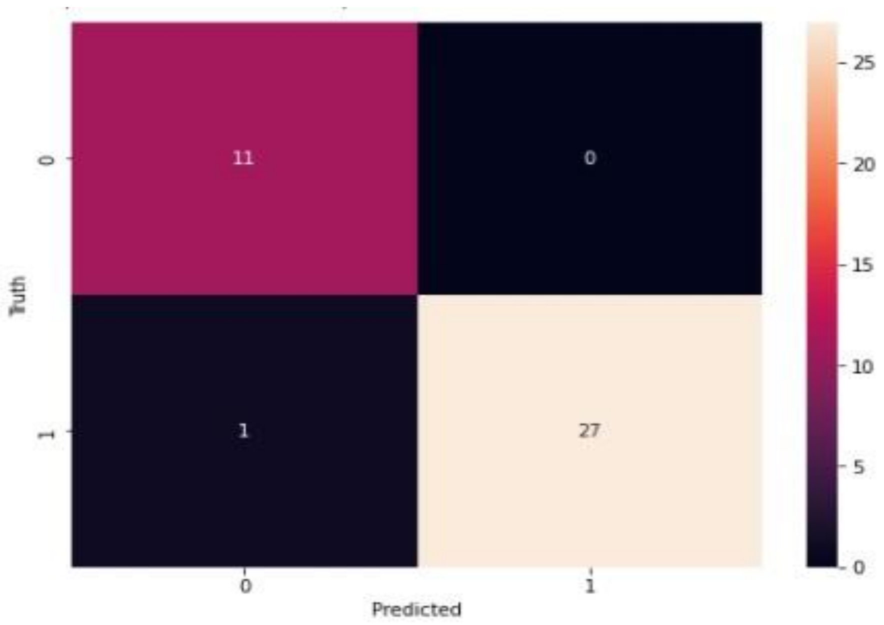

Logistic Regression Example
y
x
Boundary
False samples
True samples


$y = b_0 + b_1 x$
Linear Model
Logistic Model
$p = \frac{1}{1 + e^{-(b_0 + b_1 x)}}$
y
p
x
1
0

Like linear model uses gradient descent for the best fit logistical regression uses a maximum likelihood estimation method for best fit.

Our second model in this project is being made with the help of logistic regression we have fitted the values of X train and Y train in logistic regression model after death we have predicted the result and calculated the accuracy in this case we got a 97% accuracy a confusion matrix based on the same model is also constructed.

```
[ ] model2= LogisticRegression()

model2.fit(X_train,Y_train)

[ ] Y_predmod2 = model2.predict(X_test)

[ ] Y_predmod2

array([1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0,
       0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0], dtype=int64)

[ ] Y_test

array([1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
       0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0], dtype=int64)

[ ] print('Logistic Regression Accuracy:')
    model2.score(X_test,Y_test)

Logistic Regression Accuracy:
0.9743589743589743
```

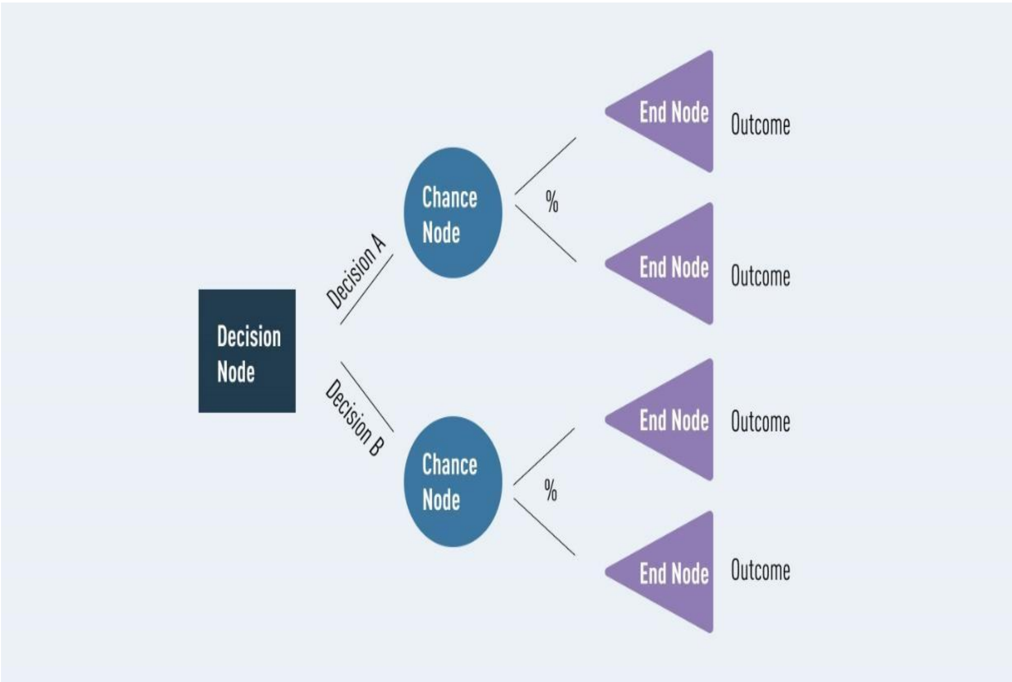


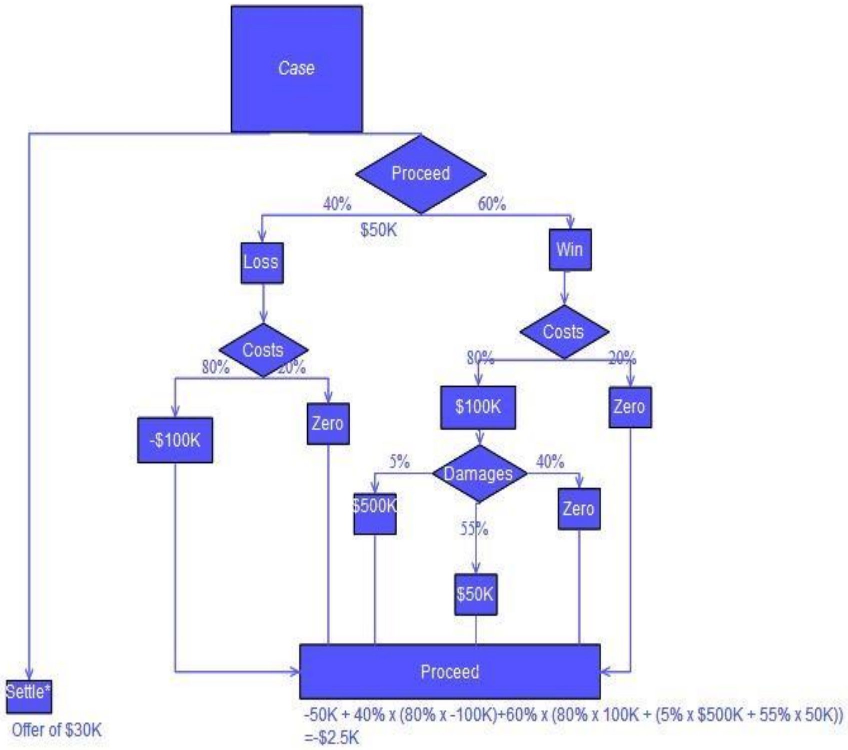
Implementation figures accuracy score and confusion matrix of logistic regression respectively.

6.) Decision Tree:

A decision tree organizes results in a flowchart manner with the help of conditional statements it classifieds data and then branches the data into two or more directions. Branches direct us to two different possible outcomes decision tree provides an optimal visual solution to the user and is extremely useful in data analysis. It is responsible for giving us solutions hey now in a loose fashion it breaks complex data and divides it into different parts. A decision tree is used for various predictions including disease prediction, vaccination predictions, economical predictions, electronics, and communication predictions with computer science predictions. A decision tree comprises three parts. Decision notes, chance notes, and end nodes are depicted in squares, circles, and triangles respectively and all these nodes are connected with branches

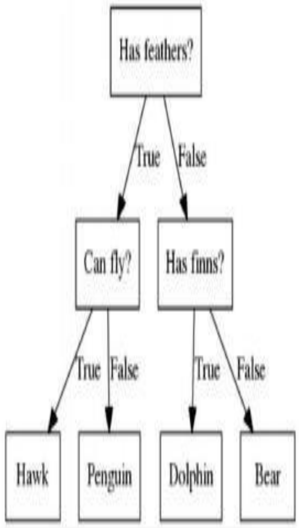
.



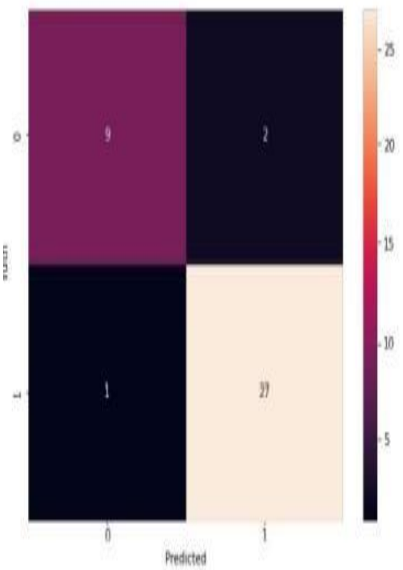


It also contains a root note which is the first node on the path, a leaf node that shows the end of the decision path, and internal nodes that are present between the root and the leaf nodes. The decision tree starts at a single control statement to classify data which then splits into two or more directions.

The third model in our project the pics they use of decision tree we have fitted the train data sets into the decision tree model after making predictions we have calculated the accuracy score of 92% And a confusion matrix is also made for the same.



	precision	recall	f1-score	support
0	0.90	0.82	0.86	11
1	0.93	0.96	0.95	28
accuracy			0.92	39
macro avg	0.92	0.89	0.90	39
weighted avg	0.92	0.92	0.92	39




```
from sklearn import tree

model3=tree.DecisionTreeClassifier()
```

In [47]:

```
model3.fit(X_train,Y_train)
```

Out[47]:

```
DecisionTreeClassifier()
```

In [48]:

```
Y_predmod3=model3.predict(X_test)
Y_predmod3
```

Out[48]:

```
array([1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
       0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0], dtype=int64)
```

In [49]:

```
Y_test
```

Out[49]:

```
array([1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
       0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0], dtype=int64)
```

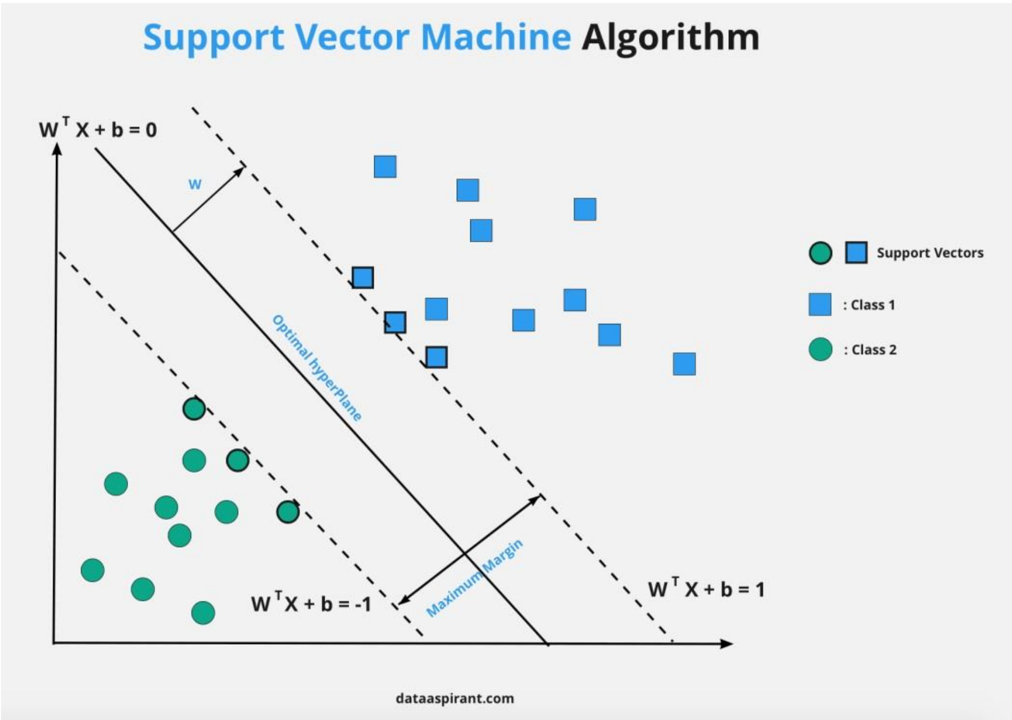
In [50]:

```
print('Decision tree Accuracy:')
model3.score(X_test,Y_test)
```

```
Decision tree Accuracy:
```

7.) Support Vector Machine Model:

The vector support machine is a supervised learning algorithm for regression and classification questions and scenarios. Creates a decision boundary that can divide N-dimensional spaces into classes this decision boundary is called a hyperplane. It selects excess points to create hyperplanes and is called supporting vectors. We have two types of vertical and indirect vector learning machine. Is Kim creating multiple decision boundaries to segregate data but we need to find out the best decision boundary to classify our data The best decision boundary is known as the hyperplane supporting vectors are the data points which are in the most proximity to the hyperplane.



In our 4th model, we have applied the support vector machine method for that we have imported SVM from SK learn then fitted it with our X train and Y train values after that we have predicted the accuracy of 84%, we have also constructed a confusion matrix for the same model.

The implementation and the accuracy with confusion matrix have been attached below:

MODEL 4 - SUPPORT VECTOR MACHINE

In [52]:

```
from sklearn.svm import SVC
model4 = SVC()
```

In [53]:

```
model4.fit(X_train,Y_train)
```

Out[53]:

```
SVC()
```

In [54]:

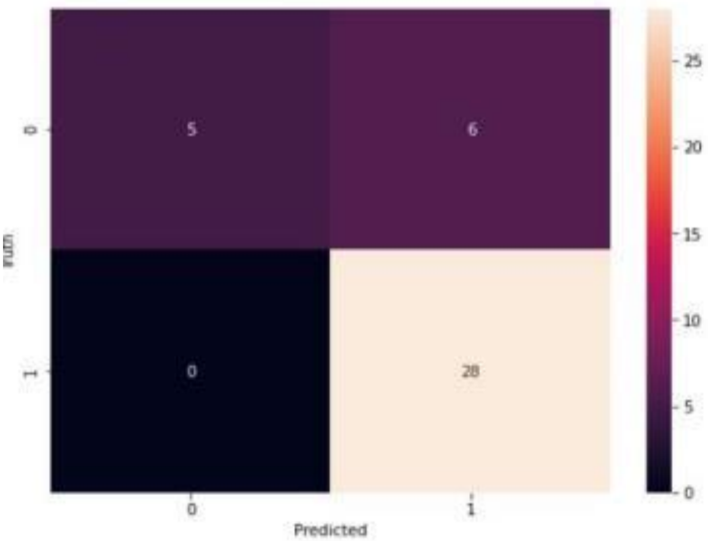
```
print('Support Vector MachinesAccuracy:')
model4.score(X_test,Y_test)
```

```
Support Vector MachinesAccuracy:
```

Out[54]:

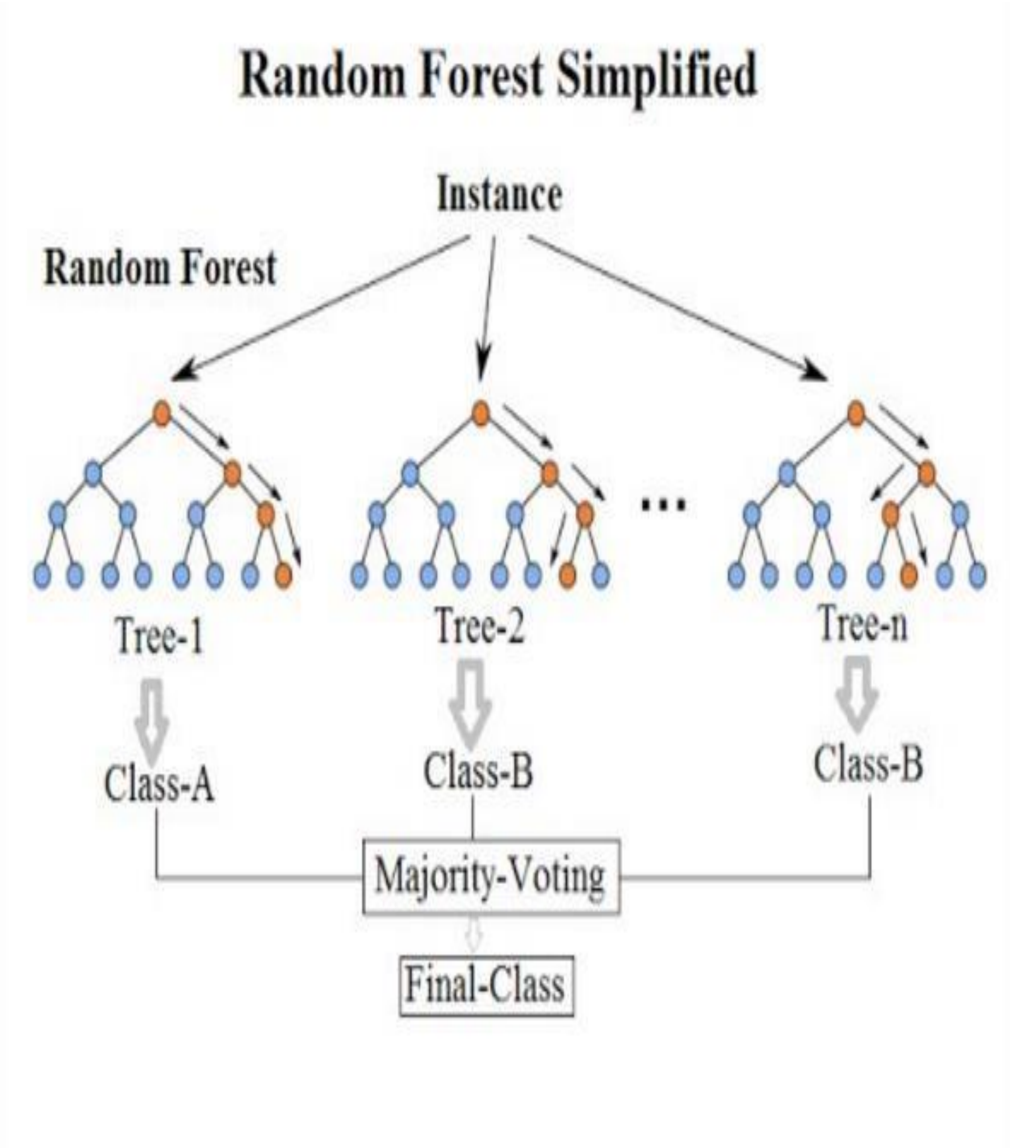
```
0.8461538461538461
```

	precision	recall	f1-score	support
0	1.00	0.45	0.62	11
1	0.82	1.00	0.90	28
accuracy			0.85	39
macro avg	0.91	0.73	0.76	39
weighted avg	0.87	0.85	0.82	39



8.) Random Forest Model:

Another supervised learning technique that we have used in our model is random forest classifier it is also among the most popular algos used for classification or regression problems random forest classifier predicts data by merging multiple decision trees with the help of randomness it enhances the prediction accuracy and provides the better result as compared to the decision tree random forest classifier can be used in various disease prediction, various economical prediction, and prediction in electronics and communication sector. If a data set will have N features it will select some random features which are less than the value of total features and will calculate the root node among those fewer selected features by picking the node with the highest information then the algorithm splits the node into child node and repeats the procedure. The random forest algorithm is more reliable than the decision tree divider will not deal with any overfill problems and can be used for both regression and classification problems. missing values also does not cause any issue while using random forest realize this is slower than other algorithms because of multiple decision tree formation and some time is unfit for fast predictions.



In our 5th model, we have applied the Random Forest Classifier for that we have imported RandomforestClassifier from SK learn then fitted it with our X train and Y train values after that we have predicted the accuracy of 97%, we have also constructed a confusion matrix for the same model.

```
from sklearn.ensemble import RandomForestClassifier
```

In [59]:

```
model5 =RandomForestClassifier()
model5.fit(X_train,Y_train)
```

Out[59]:

```
RandomForestClassifier()
```

In [60]:

```
print('Random forest Accuracy :')
model5.score(X_test,Y_test)
```

```
Random forest Accuracy :
```

Out[60]:

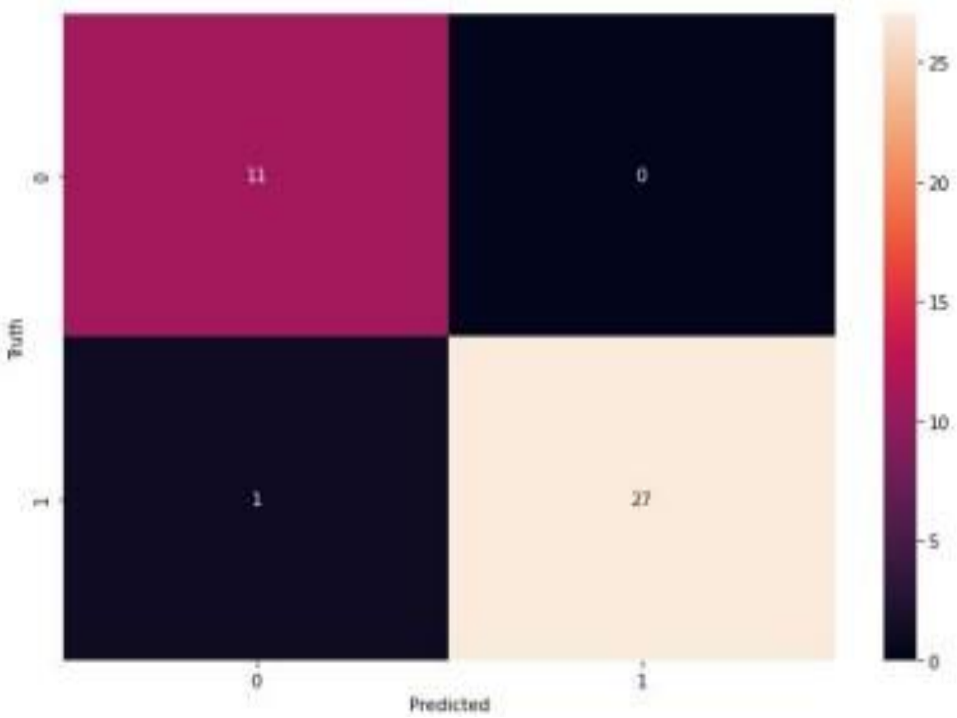
```
0.9743589743589743
```

In [61]:

```
Y_predmod5=model5.predict(X_test)
Y_predmod5
```

Out[61]:

```
array([1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
       0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0], dtype=int64)
```



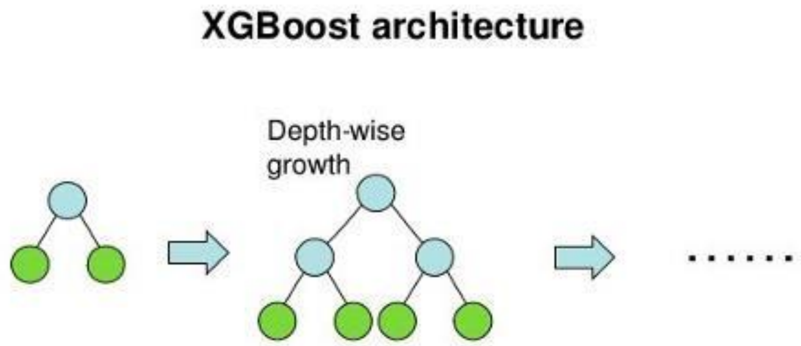
9.) XGBOOST:

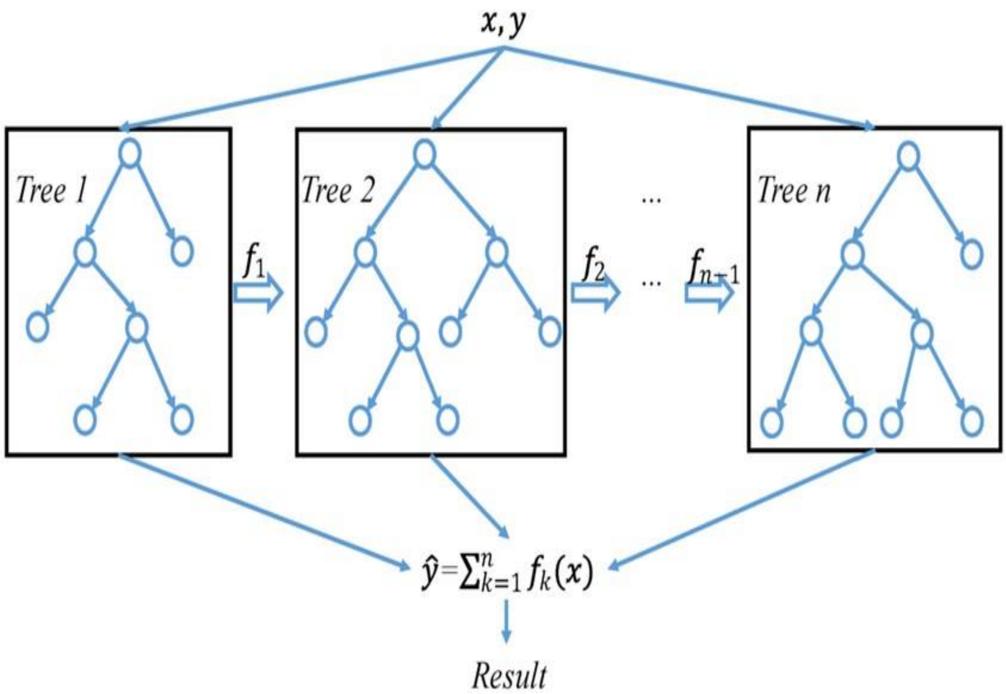
XG boost or extreme gradient boost is designed for being highly and portable and use machine learning algorithms under the framework of gradient solving data science problems in a fast and accurate way. It is the extension of gradient boosted decision tree and it's much faster than the GBM it supports regularized learning, gradient tree boosting and shrinkage, and column subsampling. It has a remarkably fast speed as compared to other algorithms and the model performance is also far better than any other model used for supervision and prediction. The algorithm works on a decision tree and constructs a graph after examining various if statements this algorithm will add more and more if to the tree for stronger model construction.

XGBoost is a boosting-based ensemble learning method. In XGBoosting, the trees are arranged in such a way that each subsequent tree aims to reduce the flaws of the previous tree. Each tree learns from its predecessors and reviews the remaining errors. Therefore, the tree that grows next in succession will learn from the revised version of the fossils. The boosting ensemble strategy consists of three simple steps:

- The first model F0 is defined by the predictable variable prediction y. This model will be associated with the remainder (y - F0)
- The new h1 model is equal to the remnants of the previous step • Now, F0 and h1 are combined to offer F1, an improved version of F0.

The average square error from F1 will be lower than that from F0.





We have constructed our 6th model with the help of XG booster for that first we have installed XG booster then imported it then we have fitted X train and Y train values in our model after that we have predicted values and calculated need accuracy with XG booster we have got an accuracy of 100% after that we have constructed a confusion matrix for the same model mentioned above. The implementation and the confusion matrix for the model mentioned above is been presented below :

In [64]:

```
pip install xgboost
```

```
Requirement already satisfied: xgboost in c:\users\yash\anaconda3\envs\tensorflow\lib\sit
e-packages (1.4.2)
Requirement already satisfied: numpy in c:\users\yash\anaconda3\envs\tensorflow\lib\site-
packages (from xgboost) (1.16.6)
Requirement already satisfied: scipy in c:\users\yash\anaconda3\envs\tensorflow\lib\site-
packages (from xgboost) (1.7.1)
Note: you may need to restart the kernel to use updated packages.
```

In [65]:

```
import xgboost as xgb
```

In [66]:

```
model6 = xgb.XGBClassifier()
```

In [67]:

```
model6.fit(X_train,Y_train)
```

```
[15:16:53] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/lear
ner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objec
tive 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric
if you'd like to restore the old behavior.
```

```
C:\Users\yash\anaconda3\envs\tensorflow\lib\site-packages\xgboost\sklearn.py:1146: UserWa
rning: The use of label encoder in XGBClassifier is deprecated and will be removed in a f
uture release. To remove this warning, do the following: 1) Pass option use_label_encoder
=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers
starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
```

Out[67]:

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning_rate=0.300000012, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=nan, monotone_constraints='()',
              n_estimators=100, n_jobs=8, num_parallel_tree=1, random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
              tree_method='exact', validate_parameters=1, verbosity=None)
```

```
Y_predmod6=model6.predict(X_test)
```

In [69]:

```
x6=model6.score(X_test,Y_test)
print('Accuracy of XGB Classifier',x6)
```

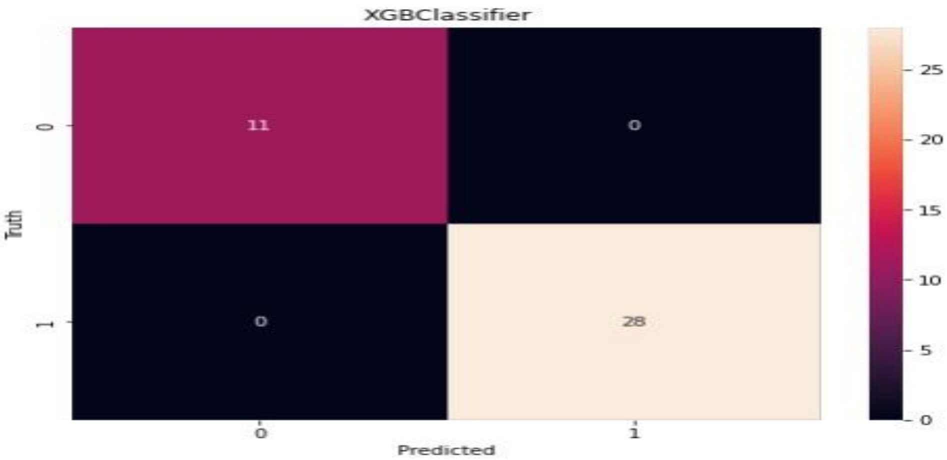
```
Accuracy of XGB Classifier 1.0
```

In [70]:

```
cm=confusion_matrix(Y_test,Y_predmod6)
plt.figure(figsize=(8,6))
plt.title('XGBClassifier')
fg=sn.heatmap(cm,annot=True)
figure=fg.get_figure()
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

Out[70]:

```
Text(51.0, 0.5, 'Truth')
```



CHAPTER 4
RESULT AND EXPERIMENTAL ANALYSIS

4.1) Discussion on the Results Achieved

4.1.1) Accuracy of Models:

The various models that we have used during the implementation of Parkinson's Disease Predictors were:

- Linear_Regression Model
- Logistic_Regression Model
- Decision_Tree Model
- Support_Vector_Machine Model
- Random_Forest Model
- XGBoost Model

During the implementation process we used two different parameters for predicting the goodness of our models which were:

- Accuracy Score
- Confusion Matrix

The Accuracy Score of Our Models Were:

- **Linear Regression Model: Accuracy Of 66%**

```
print("Linear Regression Accuracy :\n")
lr.score(X_test, Y_test)
```

```
Linear Regression Accuracy :


Out[31]:

0.6634994862742847
```

- **Logistic Regression: Accuracy - 97%**

```
print('Logistic Regression Accuracy:')
model2.score(X_test,Y_test)
```

```
Logistic Regression Accuracy:

Out[43]:

0.9743589743589743
```

- **Decision Tree Model: Accuracy - 92%**

```
print('Decision tree Accuracy:')
model3.score(X_test,Y_test)
```

```
Decision tree Accuracy:

Out[50]:

0.9230769230769231
```

- **Support Vector Machine: Accuracy Of 84%**

```
print('Support Vector MachinesAccuracy:')
model4.score(X_test,Y_test)
```

```
Support Vector MachinesAccuracy:

Out[54]:

0.8461538461538461
```

- **Random Forest Classifier: Accuracy Of 97%**

```
print('Random forest Accuracy :')
model5.score(X_test,Y_test)
```

```
Random forest Accuracy :

Out[60]:

0.9743589743589743
```

- **XGBooster: Accuracy of 100%**

```
x6=model6.score(X_test,Y_test)
print('Accuracy of XGB Classifier',x6)
```

```
Accuracy of XGB Classifier 1.0
```

In Our Research, the most effective model with a 100% accuracy score was of XGBoost

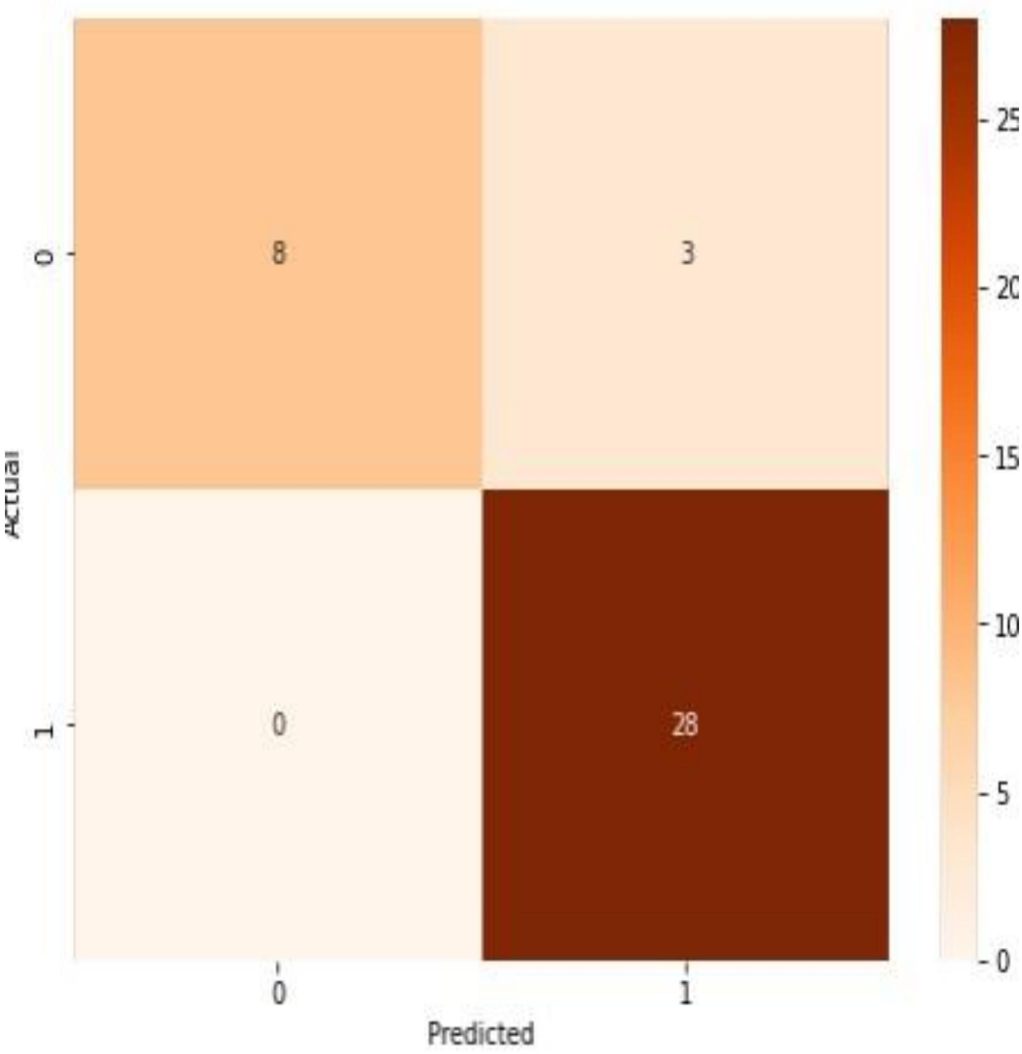
Summary Of Accuracy Evaluation:

```
Linear regression Accuracy :
0.6634994862742847

Logistic Regression Accuracy :
0.9743589743589743

Decision tree Accuracy :
0.9230769230769231
```

```
Support vector machines Accuracy :
0.8461538461538461

Random forest Accuracy :
0.9743589743589743

XGBClassifier Accuracy :
1.0
```

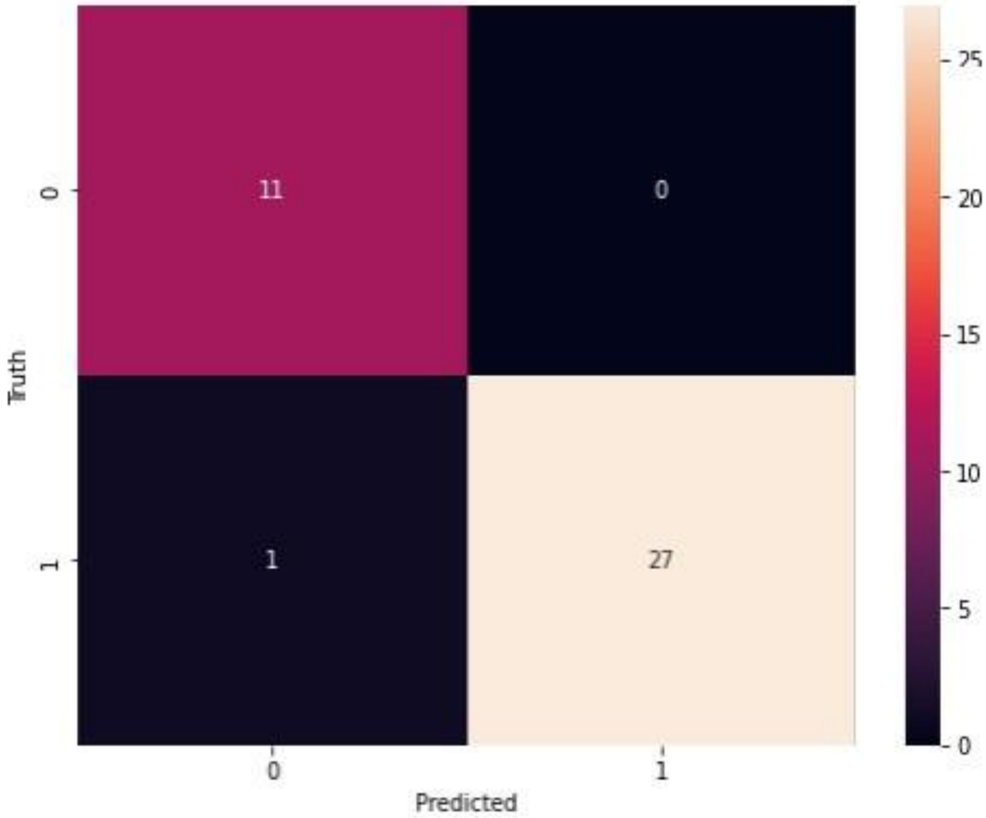
4.1.2) Confusion Matrix of our implemented Models:

- **Linear Regression Model:**



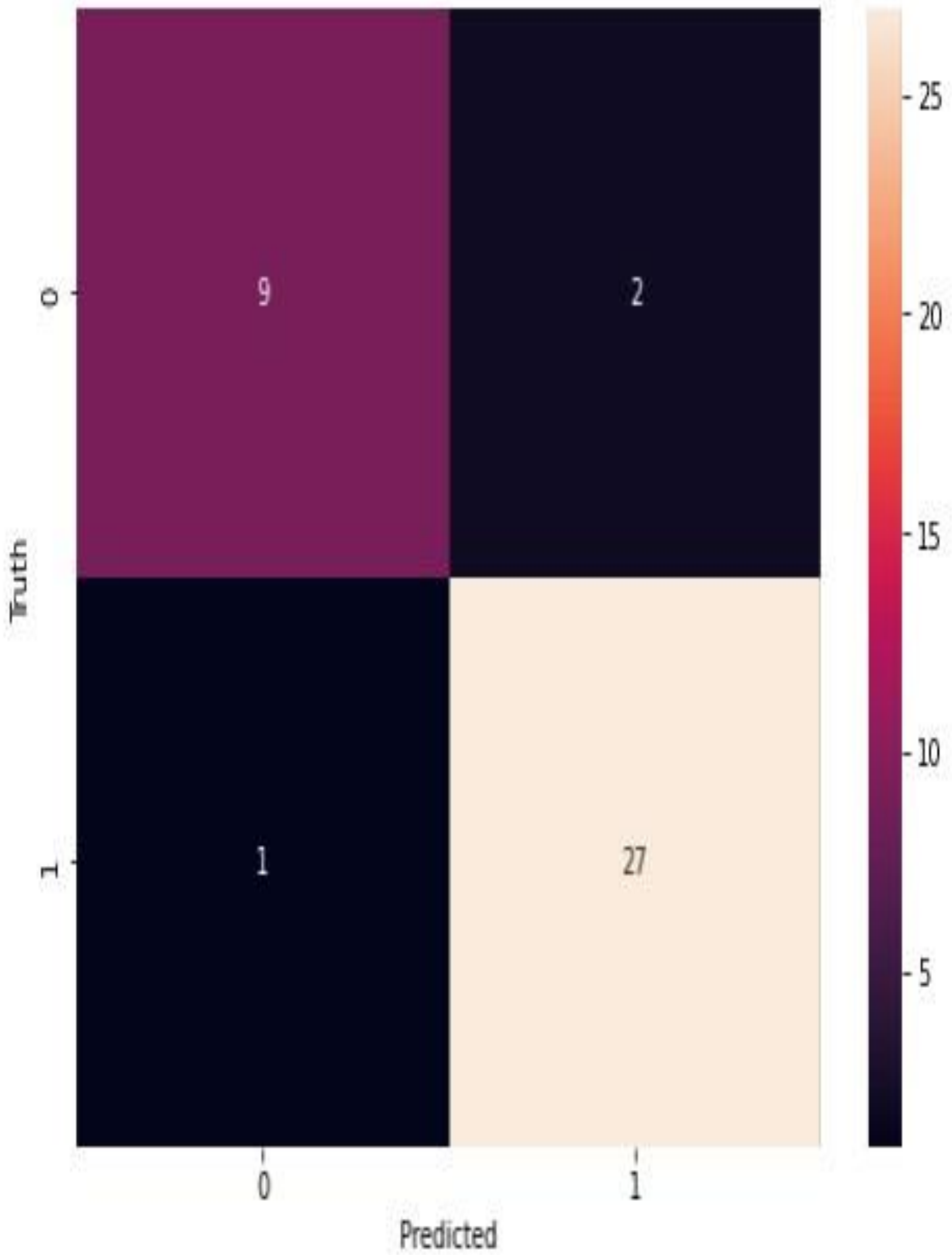
True Positives:28, True Negitave:8, False Positive:0, False Negitive:3.

- **Logistic Regression:**



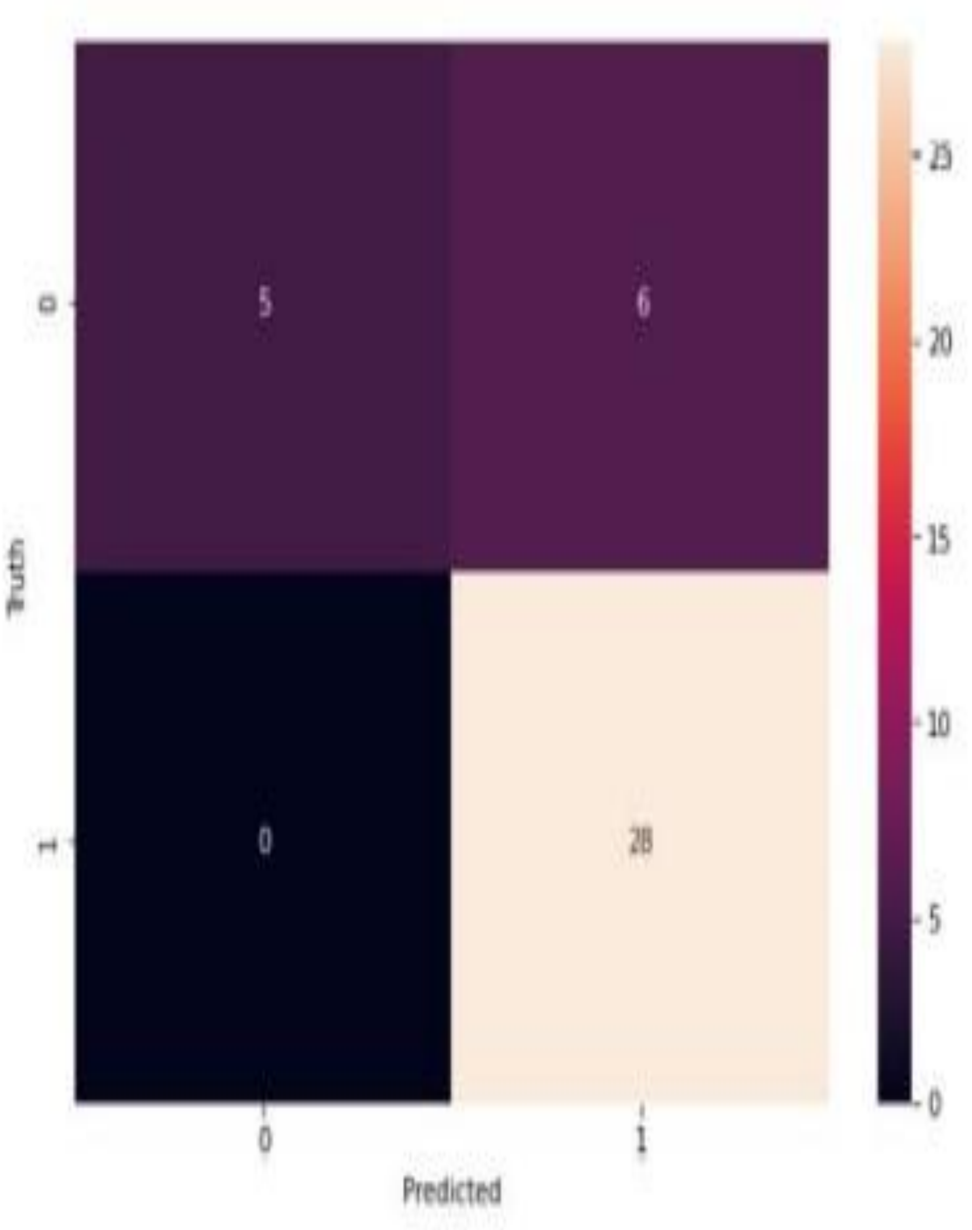
True Positives:27, True Negitave:11, False Positive:1, False Negitive:0.

- **Decision Tree:**



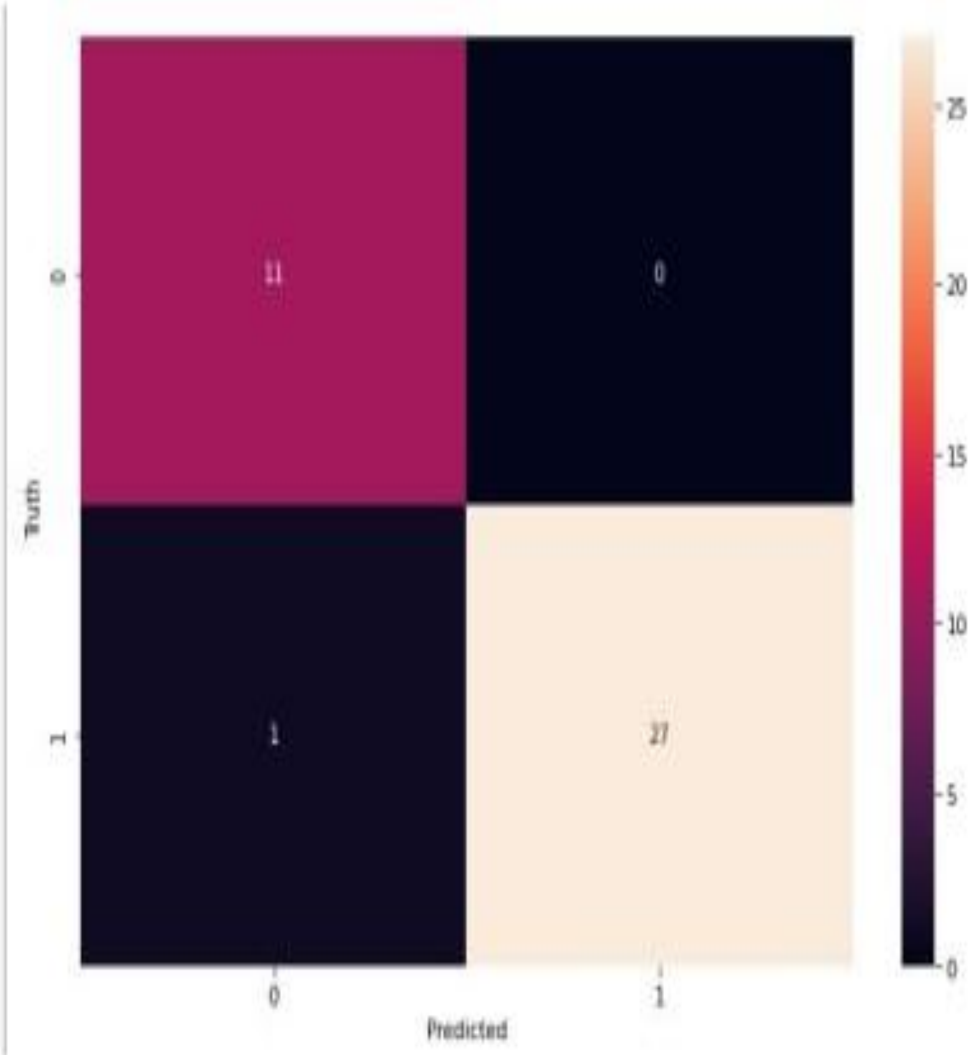
True Positives:27, True Negitave:9, False Positive:1, False Negitive:2.

▪ **Support Vector Machine**



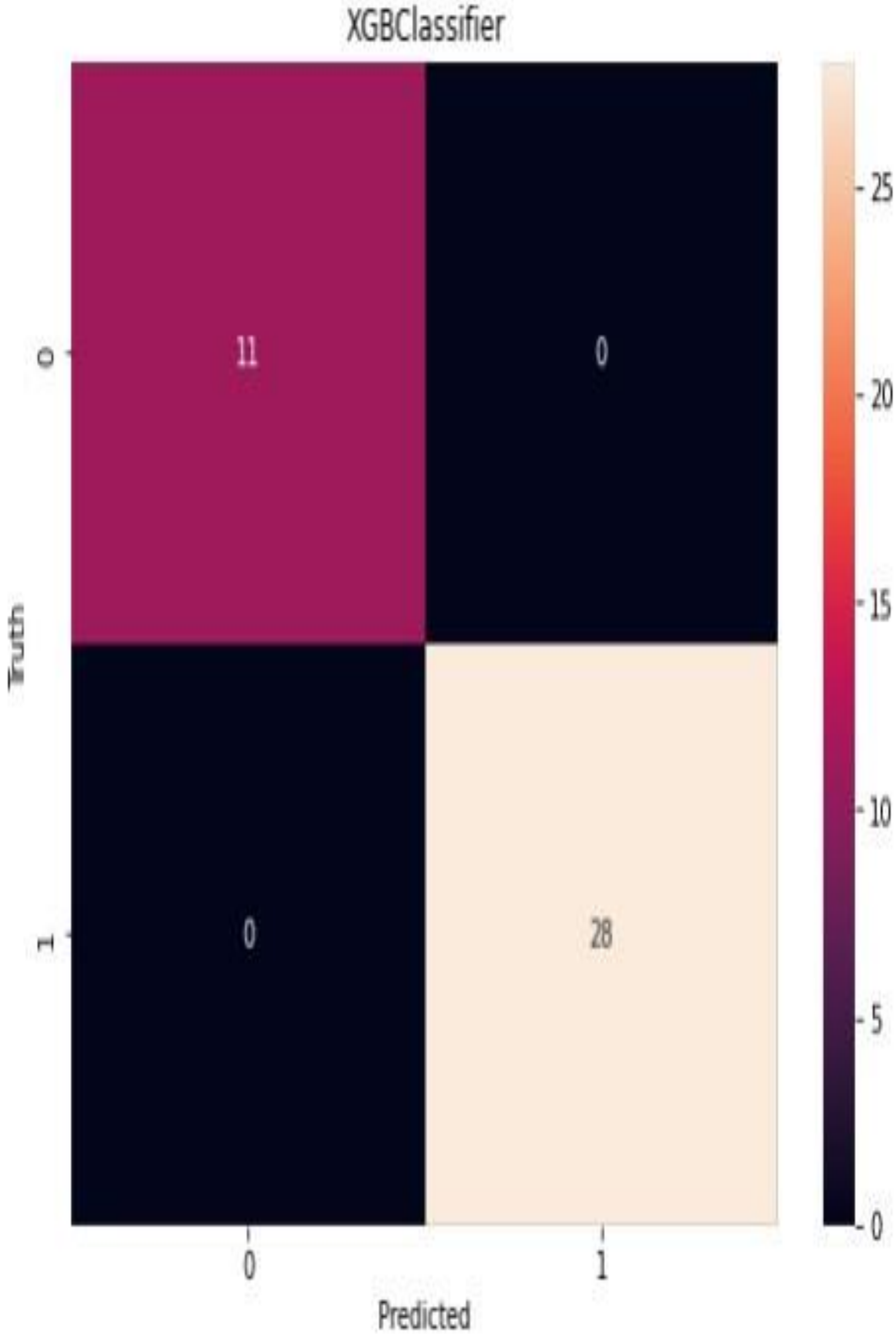
True Positives:28, True Negitave:5, False Positive:0, False Negitive:6.

- **Random Forest Classifier:**



True Positives:28, True Negitave:10, False Positive:0, False Negitive:1.

- **XGBOOSTER**:



True Positives:28, True Negitave:11, False Positive:0, False Negitive:0.

4.1.3 Classification Reports

- **Logistic Rgression**

	precision	recall	f1-score	support
0	0.92	1.00	0.96	11
1	1.00	0.96	0.98	28
accuracy			0.97	39
macro avg	0.96	0.98	0.97	39
weighted avg	0.98	0.97	0.97	39

- **Decision Tree**

	precision	recall	f1-score	support
0	0.90	0.82	0.86	11
1	0.93	0.96	0.95	28
accuracy			0.92	39
macro avg	0.92	0.89	0.90	39
weighted avg	0.92	0.92	0.92	39

- Support Vector Machine

	precision	recall	f1-score	support
0	1.00	0.45	0.62	11
1	0.82	1.00	0.90	28
accuracy			0.85	39
macro avg	0.91	0.73	0.76	39
weighted avg	0.87	0.85	0.82	39

- **Random Forest**

	precision	recall	f1-score	support
0	0.92	1.00	0.96	11
1	1.00	0.96	0.98	28
accuracy			0.97	39
macro avg	0.96	0.98	0.97	39
weighted avg	0.98	0.97	0.97	39

- **XGBooster**

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11
1	1.00	1.00	1.00	28
accuracy			1.00	39
macro avg	1.00	1.00	1.00	39
weighted avg	1.00	1.00	1.00	39

4.2) Project Outcome:

- Project Accuracy: 100%.
- Research Paper cum Project Report: Completed.
- Website: N/A
- Patent: N/A

CHAPTER 5
CONCLUSIONS

5.1) Conclusion:

We, in this Major Project, Developed a Parkinson's Predictor. We Initially started with a database of human voices parameters of 31 different people, 23 of which were actually suffering from the disease. we Initially wanted to develop a model with the highest accuracy and with greater true Positives and true negatives. We used several Supervised algorithms including (Linear Regression, Logistical Regression, Decision Trees, Support Vector Machine, Random Forest and XGBooster. From our calculation and models, we were able to:

- Achieved 100% Accuracy using XGBooster.
- Achieved 97% Accuracy using Random Forest.
- Achieved 84% Accuracy using SVM.
- Achieved 92% Accuracy using the Decision Tree Model.
- Achieved 97% Accuracy using Logistic Regression Model.
- Achieved 66% Accuracy using Linear Regression Model.

XGBooster Provides Us the best Accuracy Score (100%) with (True Positives:28, True Negitave:11, False Positive:0, False Negitive:0), Therefore XGBoost Model was the most accurate for our Parkinson's predictor

5.2) Applications of the Major Project

- Biomarkers derived from the human voice can provide insight into neurological problems, such as Parkinson's disease (PD), due to their subtle cognitive function and neuromuscular function.
- PD is an ongoing neurological disease that affects about 7 million people worldwide (most of them adults), with approximately 150 thousand new clinical diagnoses each year. Historically, PD has been difficult to diagnose and physicians tend to focus on some symptoms while becoming others, relying heavily on the rating scale. Due to the declining motor control that is a symptom of the disease, the term can be used as a means of detecting and diagnosing PD.
- With the advancement of technology and the proliferation of audio collection devices in everyday life, reliable models can translate this audio data into a diagnostic tool for health care professionals who can provide a cheaper and more accurate diagnosis. We provide evidence to substantiate this concept here using a set of voice data collected from people with and without PD.

5.3) Limitations of the Major Project:

Note that although in our Project we have got a 100 % prediction accuracy, we are still researching on deep learning and neural network methods corresponding to this project as they can work better and more efficiently if the data set in hand was more complex and bigger.

Contributions:

- **LALIT YADAV**

Research and Development for the Dataset, Data Preprocessing, Exploratory Data Analysis, XGBoost Implementation, Random Forest Implementation, Visualization of results.

Linear Regression Implementation, Logistic Regression Implementation, Decision Tree Implementation, Support Vector Machine implementation.

5.4) Future Work

- Note that although in our Project we have got a 100 % prediction accuracy, we are still researching on deep learning and neural network methods corresponding to this project as they can work better and more efficiently if the data set in hand was more complex and bigger.
- First, we are thinking of increasing the size of the current data set by a huge amount by gathering data off the internet or by putting some random manual data into it, then we will try to implement a Neural Network for Parkinson's Disease Detection using TensorFlow and keras, as of right now we are in the process of learning and researching about neural network and its implementation on this project

REFERENCES

[1] Max A. Little, Patrick E. McSharry, Eric J. Hunter, Lorraine O. Ramig (2008),'Suitability of dysphonia measurements for telemonitoring of Parkinson's disease',IEEE Transactions on Biomedical Engineering https://dx.doi.org/10.1109%2FTBME.2008.2005954

[2] Little, M.A., McSharry, P.E., Roberts, S.J. et al. Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection. BioMed Eng OnLine 6, 23 (2007). https://doi.org/10.1186/1475-925X-6-23

[3] Diagnosis of Parkinson's Disease Using SVM Classifier G. wiselin Jiji, A. Rajesh and P. Johnson Durai Raj https://doi.org/10.1142/S021946782150011X

[4] LR, LOR for the Diagnosis of Parkinson's Disease Jie Mei, Christian Desrosiers and Johannes Frasnelli. https://doi.org/10.3389/fnagi.2021.63375

[5] Abiyev R. H., Abizade S. (2016). Diagnosing Parkinson's diseases using fuzzy neural system. Comput. Mathe. Methods Med. 2016:1267919. 10.1155/2016/1267919 [PMC free article] [PubMed] [CrossRef] [Google Scholar]

[6] Abos A., Baggio H. C., Segura B., Campabadal A., Uribe C., Giraldo D. M., et al. . (2019). Differentiation of multiple system atrophy from Parkinson's disease by structural connectivity derived from probabilistic tractography. Sci. Rep. 9:16488. 10.1038/s41598-019-52829-8 [PMC free article] [PubMed] [CrossRef] [Google Scholar]

[7] Abujrida H., Agu E., Pahlavan K. (2017). Smartphone-based gait assessment to infer Parkinson's disease severity using crowdsourced data, in

2017 IEEE Healthcare Innovations and Point of Care Technologies (HIPOCT) (Bethesda, MD:), 208–211. 10.1109/HIC.2017.8227621
[CrossRef] [Google Scholar]

[8] Adams W. R. (2017). High-accuracy detection of early Parkinson's Disease using multiple characteristics of finger movement while typing. PLoS ONE 12:e0188226. 10.1371/journal.pone.0188226 [PMC free article] [PubMed] [CrossRef] [Google Scholar]

[9] Adeli E., Shi F., An L., Wee C.-Y., Wu G., Wang T., et al. . (2016). Joint feature-sample selection and robust diagnosis of Parkinson's disease from MRI data. NeuroImage 141, 206–219. 10.1016/j.neuroimage.2016.05.054 [PMC free article] [PubMed] [CrossRef] [Google Scholar]

[10] Adeli E., Thung K.-H., An L., Wu G., Shi F., Wang T., et al. . (2019). Semi-supervised discriminative classification robust to sample-outliers and feature-noises. IEEE Trans. Pattern Anal. Mach. Intell. 41, 515–522. 10.1109/TPAMI.2018.2794470 [PMC free article] [PubMed] [CrossRef] [Google Scholar]