NAME: SHAIK NYAMATHULLA

USN: 1NH18IS101

SEM: 6

SEC: B

TOPIC: PARKINSONS DISEASE PREDICTION

1NH18IS101

ORIGINALITY REPORT

24%	9%	11%	19%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Thapar University, Patiala Student Paper	4%
2	Submitted to University of Wales Institute, Cardiff Student Paper	3%
3	Submitted to GGS IP University Delhi Student Paper	2%
4	waset.org Internet Source	1%
5	"Big Data Analysis and Deep Learning Applications", Springer Science and Business Media LLC, 2019 Publication	1%
6	Aarushi Agarwal, Spriha Chandrayan, Sitanshu S Sahu. "Prediction of Parkinson's disease using speech signal with Extreme Learning Machine", 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016 Publication	1%
7	www.hindawi.com Internet Source	1%
8	Khalid Shaikh, Sabitha Krishnan, Rohit Thanki. "Chapter 3 Artificial Intelligence and Learning Algorithms", Springer Science and Business Media LLC, 2021 Publication	1%
9	Submitted to University of Huddersfield Student Paper	1%
10	www.researchgate.net Internet Source	1%
11	Submitted to University of Bradford Student Paper	1%
12	Submitted to Coventry University Student Paper	1%
13	www.epda.eu.com Internet Source	1%
14	Amit Gawade, Rohit Pandharkar, Subodh Deolekar, Uday Salunkhe. "Chapter 44 Early Diagnosis of Parkinson's Disease Using LSTM: A Deep Learning Approach", Springer Science and Business Media LLC, 2021 Publication	1%
15	Submitted to National College of Ireland Student Paper	1%

CHAPTER 1

INTRODUCTION

1.1 Introduction

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

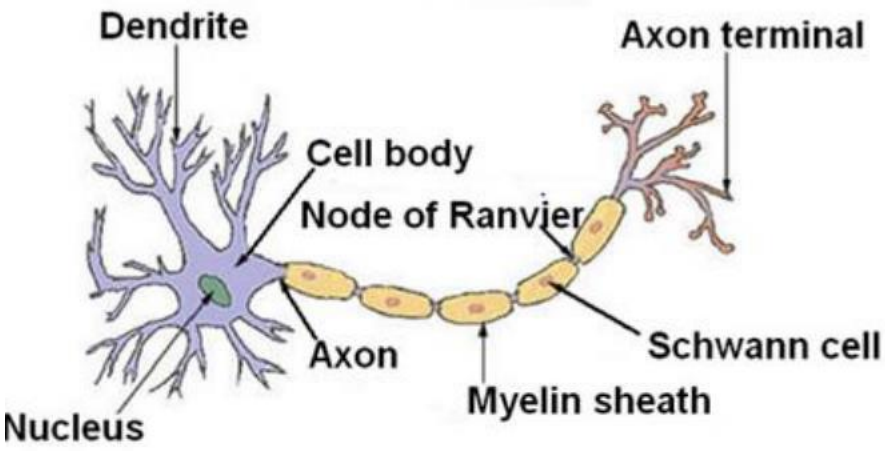

Fig 1: Structure of neuron present in human brain

This work deals with the prediction of Parkinson's disorder which is now a days is a tremendously increasing incurable disease. Parkinson's disease is most spreading disease [19] which get its xii name from James Parkinson who earlier described it as a paralysis agitans and later gave his surname was known as a PD. It generally affects the neurons which is responsible for overall body movements. Main chemicals are dopamine and acetylcholine which affects human brain.

1.2 Objectives

The main objective is to predict the prediction efficiency that would be beneficial for the patients who are suffering from Parkinson and the percentage ratio will be reduced. Generally in the first stage Parkinson can be cured by the proper treatment. So it's important to identify the PD at the early stage for the betterment of the patients. The main purpose of this research work is to find the best prediction model i.e. the best machine learning technique which will distinguishes the Parkinson's patient from the healthy person. The techniques investigated are Neural

Network, SVM, Adaboost, Bagging, Linear Regression, Random Forest, Decision trees. We have found that Neural network ,SVM, Linear Regression have been reported in various researches, whereas it has been found that only few researchers have explored Adaboost and bagging. The experimental study is performed on the biomedical voice measurement from 31 people, 23 with Parkinson's xv disease. The prediction is evaluated using error rates.. Further the Feature selection technique has been implemented with the aim to get the important features that can detect the Parkinson's disease.

1.3 Parkinson's Disease Symptoms

The symptoms of the Parkinson disease broadly divided into two categories.

• Motor symptoms •

Non-motor symptoms

1.3.1 Motor symptoms

This is a symptoms where any voluntary action involved. It's indicates the movement related disorder like tremor, rigidity, freezing, Bradykinesia or any voluntary muscle movement.

1.3.2 Non-motor symptoms

Non motor symptoms include disorders of mood and affect with apathy, cognitive dysfunction as well as complex behavioral disorders. There are two other categories of PD which are divided by doctors: Primary symptom and Secondary symptom.

1.3.3 Primary symptoms

It is the most important symptom. Primary symptoms are rigidity, tremor and slowness of movement.

1.3.4 Secondary symptoms

It is a symptom which directly impact life of an individual. These can be either motor or nonmotor. Its effect depends on person to person.. A very wide range of symptoms is associated with Parkinson's,.

Besides these symptoms, there are some other symptoms found that leads to Parkinson's disease. These symptoms are micrographia, decreased olfaction & postural instability, slowing of the digestive system, constipation, fatigue, weakness and Hypotension . Speech difficulties i.e. dysphonia (impaired speech production) and dysarthria (speech articulation difficulties) are found in patients of parkinsons..

1.4 Motivation

Ten percent of people aged 65 or more do have a neurodegenerative disease, and there are no cures for them. Almost 30% of the people are facing this incurable disease[23]. Current treatment, if available at all,only reduces symptoms and that too for limited period of time. The main cause for the parkinson's disease is the accumulation of protein molecules in the neuron which gets misfolded and hence causing Parkinson's disease. So till now researchers got the symptoms and the root cause i.e. from where this disease had evolved. But very few have come to its cure.So in this era where parkinson's disease is progressing with double pace ,it is very important to find the solution which can detect it in its early phases

CHAPTER 2

SYSTEM REQUIRMENTS

2.1Hardware Requirements

- Processor : Intel® Core™
- RAM : 4.00 GB
- Hard Disk : 488 GB
- Monitor : Any

2.2Software Requirements

- Operating System: Windows 7
- Programming Language: Python
- Code Editor: Jupyter
- Browser :Google Chrome, Internet Explorer, etc.

CHAPTER3

PROBLEM DEFINATION AND OBJECTIVES

3.1 Research Gap and Problem Definition

Most of the studies reported in the literature survey focused on the usage of machine learning techniques like Logistic regression, Decision Tree, Support vector machine ,Random Forest .Very few studies performed Adaptive boosting ,Bagging and neural network . The study evaluated and compared various machine learning techniques for the early prediction of Parkinson's disease[17]. Our study is proposed with the aim to perform feature selection and to provide the comparative study of machine learning technique algorithms i.e. adaptive boosting, Bagging, Neural Network, Support vector machine, Random Forest, Decision Tree. So our study will focus on finding the best model to provide an automated method to extract the necessary biomarkers which will help in the prediction of Parkinson's disease.

3.2 Objectives

Various objectives that are needed to be fulfilled to solve the problem in hand are listed as below:

- To study and review various machine learning that could enhance the process of prediction of Parkinson's disease
- To find out the error rate using the predicted and actual values using different error techniques.
- To find various performance evaluation metrics and providing the comparative analysis to find the best method among them.
- To compute the performance of different ML techniques with various features selected by Boruta feature selection method

CHAPTER 4

MACHINE LEARNING METHODS FOR DISEASE PREDICTION

This chapter deals with the description of the dataset used and the approaches taken to achieve the early prediction of Parkinson's disease in a PD patient .The approaches taken were selected with the aim to distinguish a Parkinson's disease patient from those who are healthy patient. The idea is to do a comparative analysis of different machine learning technique by implementing different models on the selected dataset and finding the best machine learning technique among them by evaluating some performance metrics like accuracy, ROC, AAE, and ARE etc. Further the work is extended by implementing Boruta feature selection technique.

4.1 Dataset 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. This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). The parameters are classified into 6 categories i.e. Amplitude parameters, Pulse parameters , Frequency Parameters, Voicing Parameters, Pitch parameters, Harmonicity parameters as shown in the table 2. The datasets has 195 instances .Each column in the table is a particular voice measure, and each row corresponds one of the 195 voice recordings from these individuals. The 'Status' parameter is the most importance among all other parameter as it is the only parameter which will differentiate healthy people from those with Parkinson's disease. 0 states that the person is healthy while 1 states that the person has Parkinson's disease. The fig 2 illustrates the sample of data set used.

Table 2: Extracted Features From Speech Recordings

Feature	Group
Shimmer (dda) Shimmer (local) Shimmer (apq3) Shimmer (apq11) Shimmer (apq5) Shimmer (local,dB)	**Amplitude Parameters**
Number of pulses Mean period Number of periods Standard deviation of period	**Pulse Parameters**
Jitter (ddp) Jitter (local) Jitter (rap) Jitter (local, absolute) Jitter (ppq5)	**Frequency Parameters**
Number of voice breaks Fraction of locally unvoiced frames Degree of voice breaks	**Voicing Parameters**
Mean pitch Median pitch Standard Deviation Maximum pitch Minimum pitch	**Pitch Parameters**
Harmonic-to-Noise Noise-to-Harmonic Autocorrelation	**Harmonicity Parameters**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	MDVP:Fo	MDVP:Fhi	MDVP:Flo	MDVP:Jitt	MDVP:Jitt	MDVP:RA	MDVP:PP	Jitter:DDP	MDVP:Shi	MDVP:Shi	Shimmer:	Shimmer:	MDVP:APi	Shimmer:	NHR	HNR	RPDE	DFA	spread1	spread2
2	119.992	157.302	74.997	0.00784	0.00007	0.0037	0.00554	0.01109	0.04374	0.426	0.02182	0.0313	0.02971	0.06545	0.02211	21.033	0.414783	0.815285	-4.81303	0.266482
3	122.4	148.65	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	0.626	0.03134	0.04518	0.04368	0.09403	0.01929	19.085	0.458359	0.819521	-4.07519	0.33559
4	116.682	131.111	111.555	0.0105	0.00009	0.00544	0.00781	0.01633	0.05233	0.482	0.02757	0.03858	0.0359	0.0827	0.01309	20.651	0.429895	0.825288	-4.44318	0.311173
5	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	0.517	0.02924	0.04005	0.03772	0.08771	0.01353	20.644	0.434969	0.819235	-4.1175	0.334147
6	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	0.584	0.0349	0.04825	0.04465	0.1047	0.01767	19.649	0.417356	0.823484	-3.74779	0.234513
7	120.552	131.162	113.787	0.00968	0.00008	0.00463	0.0075	0.01388	0.04701	0.456	0.02328	0.03526	0.03243	0.06985	0.01222	21.378	0.415564	0.825069	-4.24287	0.299111
8	120.267	137.244	114.82	0.00333	0.00003	0.00155	0.00202	0.00466	0.01608	0.14	0.00779	0.00937	0.01351	0.02337	0.00607	24.886	0.59604	0.764112	-5.63432	0.257682
9	107.332	113.84	104.315	0.0029	0.00003	0.00144	0.00182	0.00431	0.01567	0.134	0.00829	0.00946	0.01256	0.02487	0.00344	26.892	0.63742	0.763262	-6.1676	0.183721
10	95.73	132.068	91.754	0.00551	0.00006	0.00293	0.00332	0.0088	0.02093	0.191	0.01073	0.01277	0.01717	0.03218	0.0107	21.812	0.615551	0.773587	-5.49868	0.327769
11	95.056	120.103	91.226	0.00532	0.00006	0.00268	0.00332	0.00803	0.02838	0.255	0.01441	0.01725	0.02444	0.04324	0.01022	21.862	0.547037	0.798463	-5.01188	0.325996

Fig 2: Sample dataset of biomedical voice measurements of 31 people

4.2 Prediction Techniques

4.2.1 Neural Network

Neural Network had its base as that of biological neuron which is used for prediction.. Let's understand the single neuron. In the fig 3 you can see a diagram of single neuron with single input. The given equation will explain the single input neuron where O is the output ,σ is the sigmoid function or transformed function ,ξ is the input to the neuron and ω is the weight that connects that input to the neuron O = σ(ξ ω) ……. 1

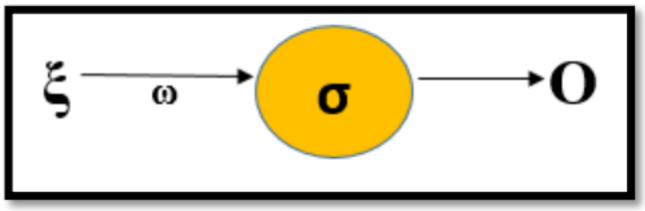

Fig 3: A Single input neuron0

So when multiple inputs are given to a neuron as mentioned in fig 4, it will form a MLP.hich consists of inputs connected through the weights in the form of layers. So the neuron takes multiple inputs and generates output which is known as Multilayer perceptron. The diagram below demonstrates a multilayer perceptron

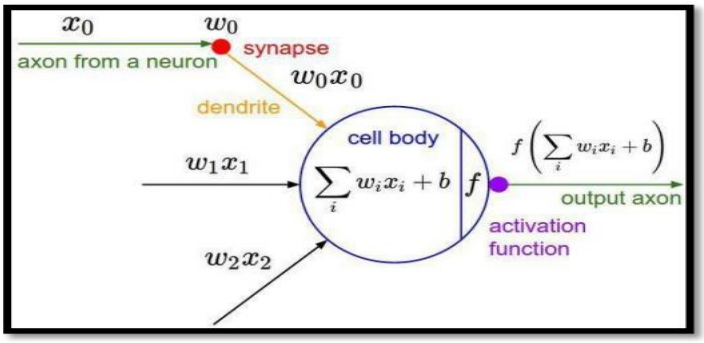

Fig 4 : Multilayer perceptron

O = $\boldsymbol{\sigma(\xi 1\ \omega 1 + \xi 2\ \omega 2 + \cdots \ldots \ldots + \xi k \omega k)}$ +Θ ……..2

where O is the output

σ is the sigmoid function or transformed function

ξ is the input to the neuron

ω is the weight of input (1 to k)

Θ is the bias

4.2.2 Linear Regression

This model is used to find relationship between two continuous variable. One variable is called the dependent or response and the other one is called the independent or predictor using a best fit straight line known as regression line. The purpose of linear regression model is that it looks for a statistic relationship between the two variable and not the deterministic variable .By deterministic relationship we mean that if one variable can be accurately expressed by the other one.

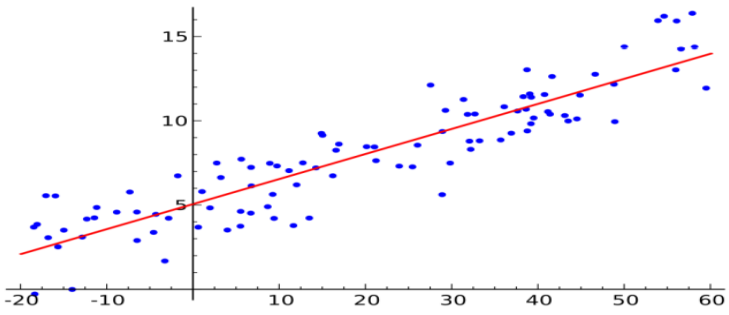

Fig 5: Straight line plot in Linear regression

The mathematical representation of Linear Regression:

Y=[X][W]+[B]……3

Y=b0 +b1X1 +b2X2 ……..4

In eq 3 and eq 4, Y is the the dependent variable and X1,X2 represents the independent variables b1, b2 are the coefficients of the independent variables and b0 is the intercept .

4.2.3 Random Forest

Random Forest is one of the machine learning method which is used for both classification as well as regression tasks. It is a type of ensemble method with which a group of weak model when combines turns into a powerful model. In random forest, multiple tress are created .To classify every tree gives a classification, are supposed to vote for that class. The forest selects the classification having the highest votes. The selection process by random forest is shown in fig 6.

Random Forest Prediction Pseudo code:

1. Takes the test sets features and make decision trees to predict the outcomes and stores the predicted outcomes.

2. Calculate the votes for each predicted outcome.

3. Consider the high voted predicted outcome as the final prediction.

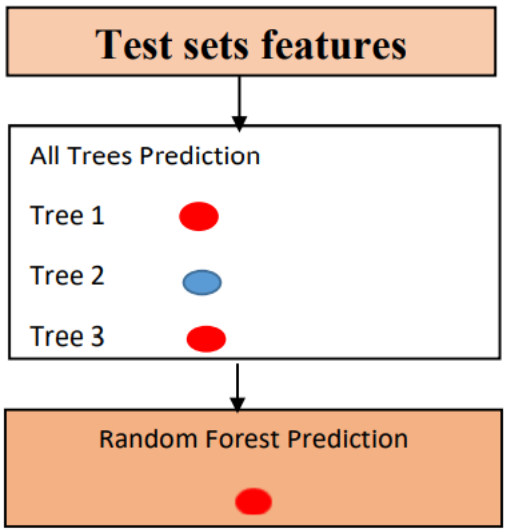

Fig 6: Prediction process taken by random forest

4.2.4. Decision Tree

Decision tree algorithm is a supervised learning algorithm which is used for the classification as well as regression problems. The main objective of using Decision tree is to create a training model which can be used for prediction of Parkinson's by learning decision rules inferred from

training datasets. It tries to resolve the problem by using tree representation or tree hierarchy.It has three nodes:

1. Root

2. Internal Nodes

3. Leaf nodes

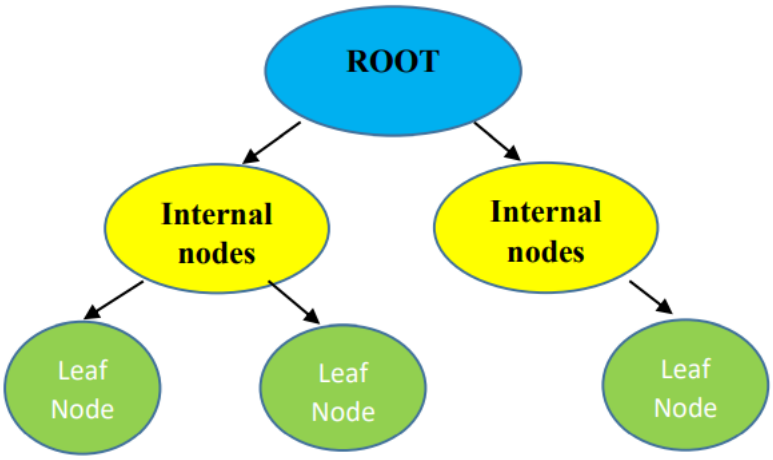

Fig 7 : Representation of decision tree

Root node represents the entire sample which is further splits into nodes known as leaf nodes which represents the attribute which is further divided into leaf nodes which represents the class labels.

4.2.5 Adaboost

Adaboost like random forest classifier is another ensemble classifier. AdaBoost which is known as adaptive boosting which is used for classification rather than regression .It is a best algorithm for predicting. It is used to boost the performance of decision tree or binary classification problems. . For the new input we are providing to adaboost, each weak learner calculates a predicted value .the vaue can be either 1.0 or -1.0. Each weak learner weights the predicted values. The prediction for the ensemble model is calculated by taking the sum of the weighted predictions.If the Sum is positive it will be assigned First predicted class,if Sum is negative it comes under Second predicted class.

Mathematics involved in Adaboost :

T

H(x) = sign(∑ ah(x)

t=1

h(x) is the output of weak classifier t for input x

a is the weight assigned to classifier.

a = 0.5 * ln ((1-E)/E)

4.2.6 Support Vector Machine

Support vector machine is defined by separating hyperplane. The output of the approach is an optimal hyper-plane which categorizes new examples. In 2 dimensional space, this new hyper plane in a line dividing a plane in two parts where each class lies in one side.It gives better result for complex classification problems. Each data item is plotted as a point in n-dimensional space with value of every feature reflecting the coordinates of the plane. The SVM is performed classification that differentiating the two classes very efficiently.

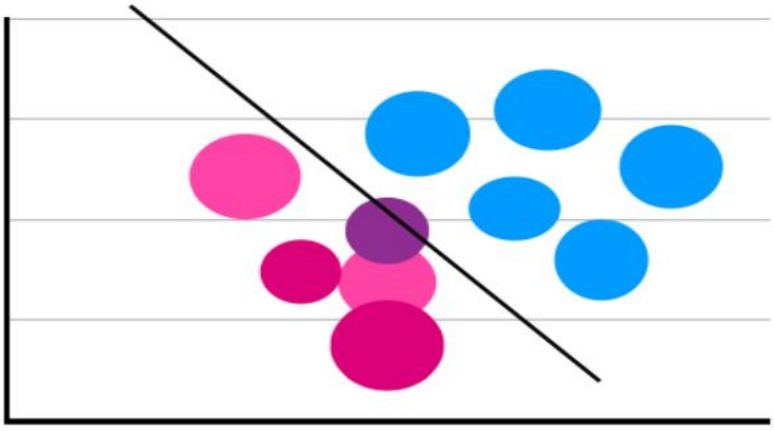

Fig 8: Hyper plane classifying two classes.

4.2.7 Bagging

Bagging is an ensemble algorithm, bagging methods forms an efficient class of algorithms which bring together several instances of black box estimators on random subsets of the original data set and then efficiently aggregate their individual predictions to process and formulate the final prediction. The bagging methods make immense efforts to reduce the variance of the base estimators by efficiently introducing the randomization into its construction and then makes an ensemble from it. Let's take an instance where you have a learner for example The Decision Tree. Many times you have made efforts to improve its accuracy and variance by applying Bootstrap technique.

1. You end up generating multiple number of samples from your data set that has been classified as training set using an approach of next scheme: you can take randomly any element from your training set and then can pull it back. This results in a scenario where some of the elements of training set will be present multiple times in the generated new sample and some will be accidentally be absent. These samples should have the same size as the train set.

2. You can train your learner on each generated sample to gain the efficient results and improve the model better.

3. When you apply the algorithm you are just doing an average predictions of learners in case of regression or make the voting in case of classification.

CHAPTER 5

METHODOLOGY

5.1 Methodology

This section explains the steps taken to achieve the prediction of Parkinson's disease using various machine learning. The various steps taken are Data gathering , Data Preprocessing, Model Selection, Training, Evaluation, prediction .

5.1.1 Data Gathering

The first step is Data gathering .This step is very important because the quality and quantity of the data you gather will directly affects the level of your prediction model. So we have taken data of different voice recordings of the patient.

5.1.2 Data preparation

In this step the data is visualized well to spot the relationship between the parameters present in the data so as to take the advantage of as well as to get the data imbalances. With this ,we need to split the data into two parts .The first part for training the model like in our model we have used 70 percent of data for training and 30 percentage for testing. Which is the second part of the data

5.1.3 Model Selection

The next step in our workflow is model selection. There are various models that have been used till date by researchers and scientist. Some are meant for image processing ,some for sequences like text, numbers or patterns. In our case we have 26 features which defines the voice recording of various patients so we have chosen such models which will classify or differentiates the unhealthy patient with the healthy one.

5.1.4 Training

Training the dataset is one of the main task of machine learning .we will apply the data to progressively improve the selected model's ability to predict better ie the actual result should

be approx. to predict one. 5.2.4 Evaluation The metrics we have calculated are ROC, Accuracy, Specificity , Precision etc. which will highlights the best algorithm among all.

5.1.4 Prediction

In this phase we finally get the model ready to detect the prediction of Parkinson's disease based on the given dataset.

5.2 Using R Tool on Standalone machine Environment

The R computer programs are an essential tool for progression in the numeric examination and machine learning spaces. R is a perfect way to deal with make reproducible, extraordinary examination. R is extensible and offers rich value for architects to manufacture their own specific gadgets and procedures for examining data. With machines winding up recognizably more basic as data generators, the noticeable quality of the dialects must be depended upon to create. In this module, the accuracy of different machine learning algorithms has been explored using R Tool on the Standalone machine. Here initial analysis has been done using Microsoft excel. A csv file has been provided as an input for R-Studio. Analysis has been done using programming language R as illustrated in fig 9.

Fig 9: Workflow of training the models of ML in R

In any case, R has both upsides and downsides that designers ought to know. With enthusiasm for the programming developing, as appeared on language notoriety files, for example, TIobe, Redmond and PyPL, R initially showed up in the 1990s and has filled in as an execution of the S measurable programming languages.

"R is the most mainstream dialect utilized as a part of the field of statistics."It has all the adaptability and power. R is in reality only accumulations of scripts that are sorted out into projects."

Data purifying/cleaning is a term identified with getting the significant data from the crude information and noisy data removal (information not profitable to us). This should be possible effectively in Microsoft Excel and is a generally utilized strategy for each information researcher.

5.3 Evaluation Criteria Used for Classification

Performance evaluations measures are the parameters which helps in comparative analysis of different machine learning techniques i.e. it tells the best algorithm among all other algorithms or method which can be used by medical science in the early prediction of neurodegenerative diseases. We have used several measures to evaluate the predictive results. These measures are average absolute error (AAE), average related error (ARE), accuracy (ACC), Precision, Receiver Operating Characteristics (ROC) , Area under ROC curve (AUC) ,sensitivity and specificity. Let's understand the performance evaluation measures.

5.3.1 Correlation Matrix

The confusion matrix is also called as Error matrix. It is a table that is often used to describe the performance of a classification method on a set of test data for which actual value are known. Each column of the matrix represents the instances in a predicted class. the correlation matrix is represented as given

Actual	Predicted		
		No	Yes
	No	TN	FP
	Yes	FN	TP

True Positive: is the count of healthy patients predicted accurately as healthy

True Negative: is the count of diseased subjects accurately predicted diseased.

False Positive: is the count of diseased patients predicted as healthy

False Negative: is the count of healthy patients predicted to be diseased

5.3.2 Accuracy and Precision

In classification, accuracy and precision are two important evaluation parameters. Accuracy is the proportion of the total number of predictions that were correct. It can be obtained by the sum of true positive and true negative instances divided by 100.And Precision is fraction of true positive and predicted yes instances. The formula of Accuracy and Precision are given below:

Accuracy = TP+TN/100 Precision = TP/TP+FP

5.3.3 Recall and F-Square

Recall is defined as the fraction between True Positive instances and Actual yes instances whereas F-Square is the fraction between product of the recall and precision to the summation of recall and precision parameter of classification. The formula of recall and precision given below:

Recall = TP/Actual Yes F-Square = 2∗Recall∗Precision/Recall +Precision

5.4.4 Sensitivity, Specificity and ROC

Sensitivity is defined as the fraction of true positive and actual yes instances whereas specificity is the difference between one and false positive rate value. .ROC is defined as the fraction between true positive rate and false positive rate.

Sensitivity = TP/TP+FN Specificity = TN/FP+TN ROC = TPR/FPR

CHAPTER 6

IMPLEMENTATION AND RESULTS

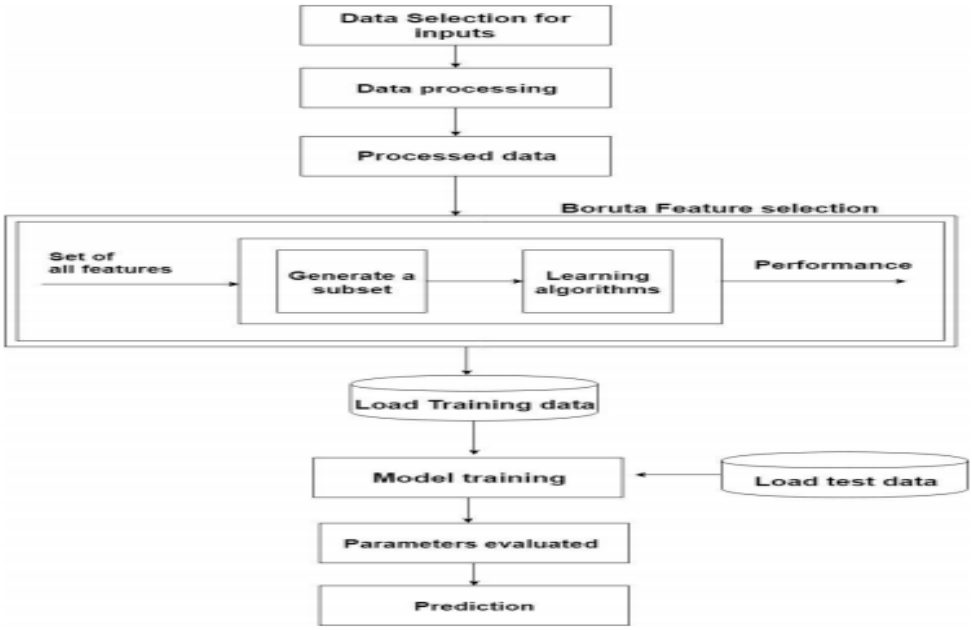

Fig 10: Feature selection by boruta method

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn import svm
from sklearn.metrics import accuracy_score
```

```
parkinsons_data = pd.read_csv('parkinsons.csv')
```

```
parkinsons_data.head()
```

	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	...	Shim
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109	0.04374	...	
1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	...	
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633	0.05233	...	
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	...	
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	...	

5 rows × 24 columns

```
parkinsons_data.shape
```

```
(195, 24)
```

```
parkinsons_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   MDVP:Fo(Hz)       195 non-null    float64
 2   MDVP:Fhi(Hz)      195 non-null    float64
 3   MDVP:Flo(Hz)      195 non-null    float64
 4   MDVP:Jitter(%)    195 non-null    float64
 5   MDVP:Jitter(Abs)  195 non-null    float64
 6   MDVP:RAP          195 non-null    float64
 7   MDVP:PPQ          195 non-null    float64
 8   Jitter:DDP        195 non-null    float64
 9   MDVP:Shimmer      195 non-null    float64
 10  MDVP:Shimmer(dB)  195 non-null    float64
 11  Shimmer:APQ3      195 non-null    float64
 12  Shimmer:APQ5      195 non-null    float64
 13  MDVP:APQ          195 non-null    float64
 14  Shimmer:DDA       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
```

```
parkinsons_data.isnull().sum()
```

```
name                0
MDVP:Fo(Hz)         0
MDVP:Fhi(Hz)        0
MDVP:Flo(Hz)        0
MDVP:Jitter(%)      0
MDVP:Jitter(Abs)    0
MDVP:RAP            0
MDVP:PPQ            0
Jitter:DDP          0
MDVP:Shimmer        0
MDVP:Shimmer(dB)    0
Shimmer:APQ3        0
Shimmer:APQ5        0
MDVP:APQ            0
Shimmer:DDA         0
NHR                 0
HNR                 0
status              0
RPDE                0
DFA                 0
spread1             0
spread2             0
D2                  0
PPE                 0
dtype: int64
```

```
parkinsons_data.describe()
```

	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	MDVP:Shimmer(dB)	.
count	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	.
mean	154.228641	197.104918	116.324631	0.006220	0.000044	0.003306	0.003446	0.009920	0.029709	0.282251	.
std	41.390065	91.491548	43.521413	0.004848	0.000035	0.002968	0.002759	0.008903	0.018857	0.194877	.
min	88.333000	102.145000	65.476000	0.001680	0.000007	0.000680	0.000920	0.002040	0.009540	0.085000	.
25%	117.572000	134.862500	84.291000	0.003460	0.000020	0.001660	0.001860	0.004985	0.016505	0.148500	.
50%	148.790000	175.829000	104.315000	0.004940	0.000030	0.002500	0.002690	0.007490	0.022970	0.221000	.
75%	182.769000	224.205500	140.018500	0.007365	0.000060	0.003835	0.003955	0.011505	0.037885	0.350000	.
max	260.105000	592.030000	239.170000	0.033160	0.000260	0.021440	0.019580	0.064330	0.119080	1.302000	.

8 rows × 23 columns

```
parkinsons_data['status'].value_counts()
```

```
1    147
0     48
Name: status, dtype: int64
```

```
parkinsons_data.groupby('status').mean()
```

status	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	MDVP:Shimmer(dB)	..
0	181.937771	223.636750	145.207292	0.003866	0.000023	0.001925	0.002056	0.005776	0.017615	0.162958	..
1	145.180762	188.441463	106.893558	0.006989	0.000051	0.003757	0.003900	0.011273	0.033658	0.321204	..

2 rows × 22 columns

```
X = parkinsons_data.drop(columns=['name','status'], axis=1)
Y = parkinsons_data['status']
```

```
print(X)
```

```
     MDVP:Fo(Hz)  MDVP:Fhi(Hz)  MDVP:Flo(Hz)  MDVP:Jitter(%)  \
0        119.992       157.302        74.997         0.00784
1        122.400       148.650       113.819         0.00968
2        116.682       131.111       111.555         0.01050
3        116.676       137.871       111.366         0.00997
4        116.014       141.781       110.655         0.01284
..           ...           ...           ...             ...
190      174.188       230.978        94.261         0.00459
191      209.516       253.017        89.488         0.00564
192      174.688       240.005        74.287         0.01360
193      198.764       396.961        74.904         0.00740
194      214.289       260.277        77.973         0.00567

     MDVP:Jitter(Abs)  MDVP:RAP  MDVP:PPQ  Jitter:DDP  MDVP:Shimmer  \
0             0.00007   0.00370   0.00554     0.01109       0.04374
1             0.00008   0.00465   0.00696     0.01394       0.06134
2             0.00009   0.00544   0.00781     0.01633       0.05233
3             0.00009   0.00502   0.00698     0.01505       0.05492
4             0.00011   0.00655   0.00908     0.01966       0.06425
..                ...       ...       ...         ...           ...
190           0.00003   0.00263   0.00259     0.00790       0.04087
191           0.00003   0.00331   0.00292     0.00994       0.02751
192           0.00008   0.00624   0.00564     0.01873       0.02308
193           0.00004   0.00370   0.00390     0.01109       0.02296
194           0.00003   0.00295   0.00317     0.00885       0.01884

     MDVP:Shimmer(dB)  ...  MDVP:APQ  Shimmer:DDA      NHR     HNR      RPDE  \
0               0.426  ...   0.02971      0.06545  0.02211  21.033  0.414783
1               0.626  ...   0.04368      0.09403  0.01929  19.085  0.458359
2               0.482  ...   0.03590      0.08270  0.01309  20.651  0.429895
3               0.517  ...   0.03772      0.08771  0.01353  20.644  0.434969
4               0.584  ...   0.04465      0.10470  0.01767  19.649  0.417356
..                ...  ...       ...          ...      ...     ...       ...
190             0.405  ...   0.02745      0.07008  0.02764  19.517  0.448439
191             0.263  ...   0.01879      0.04812  0.01810  19.147  0.431674
192             0.256  ...   0.01667      0.03804  0.10715  17.883  0.407567
193             0.241  ...   0.01588      0.03794  0.07223  19.020  0.451221
194             0.190  ...   0.01373      0.03078  0.04398  21.209  0.462803

          DFA   spread1   spread2        D2       PPE
0    0.815285 -4.813031  0.266482  2.301442  0.284654
1    0.819521 -4.075192  0.335590  2.486855  0.368674
2    0.825288 -4.443179  0.311173  2.342259  0.332634
3    0.819235 -4.117501  0.334147  2.405554  0.368975
4    0.823484 -3.747787  0.234513  2.332180  0.410335
..        ...       ...       ...       ...       ...
190  0.657899 -6.538586  0.121952  2.657476  0.133050
191  0.683244 -6.195325  0.129303  2.784312  0.168895
192  0.655683 -6.787197  0.158453  2.679772  0.131728
193  0.643956 -6.744577  0.207454  2.138608  0.123306
194  0.664357 -5.724056  0.190667  2.555477  0.148569

[195 rows x 22 columns]
```

```
print(Y)
```

```
0      1
1      1
2      1
3      1
4      1
      ..
190    0
191    0
192    0
193    0
194    0
Name: status, Length: 195, dtype: int64
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
```

```
print(X.shape, X_train.shape, X_test.shape)
```

```
(195, 22) (156, 22) (39, 22)
```

```
scaler = StandardScaler()
```

```
scaler.fit(X_train)
```

```
StandardScaler()
```

```
X_train = scaler.transform(X_train)

X_test = scaler.transform(X_test)
```

```
print(X_train)
```

```
[[ 0.63239631 -0.02731081 -0.87985049 ... -0.97586547 -0.55160318
   0.07769494]
 [-1.05512719 -0.83337041 -0.9284778  ...  0.3981808  -0.61014073
   0.39291782]
 [ 0.02996187 -0.29531068 -1.12211107 ... -0.43937044 -0.62849605
  -0.50948408]
 ...
 [-0.9096785  -0.6637302  -0.160638   ...  1.22001022 -0.47404629
  -0.2159482 ]
 [-0.35977689  0.19731822 -0.79063679 ... -0.17896029 -0.47272835
   0.28181221]
 [ 1.01957066  0.19922317 -0.61914972 ... -0.716232    1.23632066
  -0.05829386]]
```

```
model = svm.SVC(kernel='linear')
```

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

```
SVC(kernel='linear')
```

```
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(Y_train, X_train_prediction)
```

```
print('Accuracy score of training data : ', training_data_accuracy)
```

```
Accuracy score of training data :  0.8846153846153846
```

```
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(Y_test, X_test_prediction)
```

```
print('Accuracy score of test data : ', test_data_accuracy)
```

```
Accuracy score of test data :  0.8717948717948718
```

```
input_data = (186.16300,197.72400,177.58400,0.00298,0.00002,0.00165,0.00175,0.00496,0.01495,0.13500,0.00774,0.00941,0.01233,0.023
)


input_data_as_numpy_array = np.asarray(input_data)


input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)


std_data = scaler.transform(input_data_reshaped)

prediction = model.predict(std_data)
print(prediction)


if (prediction[0] == 0):
  print("The Person does not have Parkinsons Disease")

else:
  print("The Person has Parkinsons")
```

```
[1]
The Person has Parkinsons
```

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

In this work, various prediction models for Parkinson's disease detection. For this purpose seven machine learning techniques i.e. are used such as adaptive boosting, bagging, neural networks, random forest, decision tree, SVM and linear regression. To obtain the desired results, error rates are calculated i.e. AAE and ARE as well as four performance metrics are evaluated. These four metrics are accuracy, sensitivity, ROC, specificity. From the results, Random forest outstands from all the other ML techniques with the accuracy of 87%, Precision 85.0%, ROC 96.4%. After that , we tried to selected the most important and minimum number of features from the speech articulation data of 31 people where we have 23 features as explained in chapter 4 in dataset description .For that we have used Boruta feature selection whose working is shown in fig 12 by changing the number of features selected in multiples of 5 ie firstly we check over 20 features than 15 features, 10 features and lastly 5 features. From all the experiments random forest with 20 features selection outstands from all the other ML techniques as it is giving the overall accuracy 96.6%, ROC value 93.6 and precision of 88.7 which is better in comparison to all other machine learning techniques when compared with 5,10 and 15 feature's performance metrics.

7.2 Future scope

In this study we have used machine learning techniques, however very few researches have been done on deep learning methods. In future, the work can be extended by using autoencoders to reduce the number of feature and to extract the most important from them. Also the dataset used in this work is not so complex , so autoencoder did not learn well from that but with complex dataset it would definitely give better results.

BIBLIOGRAPHY

[1] www.greeksforgreek.com

[2]www.projectworld.com

[3] www.kashipara.com

[4] www.github.com