

A Data Mining Tool In Neurlogical Disorder Prediction Using Feature Selection Techniques

P. Sravanthi ^[1], B.Sai Sudha ^[2], P.Sai Ganesh ^[3], M.Dinesh ^[4],
T.Ravi Kumar ^[5], G.Rajasekharam ^[6]

Department of Computer Science and Engineering, Nadimpalli Satyanarayana Raju Institute of Technology, Andhra Pradesh,India

ABSTRACT

Data mining techniques are used for a variety of applications. In healthcare Industry, Data mining plays an important role in predicting diseases. For detecting a disease number of tests should be required from the patient. By using Data mining techniques the number of tests can be reduced. This reduced test plays an important role in time and performance. Neurological disorders are diseases of the brain, spine and the nerves that connect them. The increasing capabilities of technologies are generating massive volumes of complex data at a rapid pace. Evaluating and diagnosing disorders of the nervous system is a complicated and complex task. Many of the same or similar symptoms happen in different combinations among the different disorders. Here, we provide a developed selected data mining methods in the area of neurological diseases diagnosis. Hence, it will help experts to gain an understanding of how data mining techniques can assist them in neurological diseases diagnosis and patients treatment. Gradient boosting tree algorithm is used to predict the diseases as it produces the accurate results. Here, our aim is to find the performance of different classification methods of large database.

Keywords:- Data Mining, Classification Rules, Voice dataset, Bar Graph.

I. INTRODUCTION

PROBLEM STATEMENT

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.

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 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.

There are various environmental factor which have been implicated in PD [20].below are the listed factor which caused Parkinson's disease in an individual.

II. 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 SVM, Adaboost, naive bayes, gradient descent, gradient boosting tree 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 gradient boosting tree. The experimental study is performed on the biomedical voice measurement from 31 people, 23 with Parkinson's 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.

III. SCOPE

SCOPE By using this feature selection technique we can predict the disease in initial stage and we can save the life of effected person Prediction of Parkinson disorder is one of the most important problem that has to be detected in

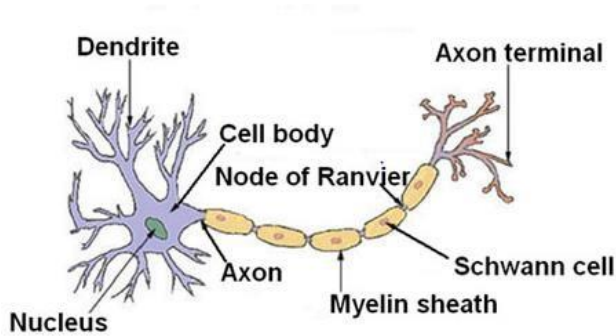


Fig 1: Structure of neuron present in human brain

the early phases of the commencement of the disease so as to reduce the disease progression rate among the individuals. Various researches have been made to find the basic cause and some have reached to the heights by proposing a system which differentiates the healthy people from those with any parkinson's data set using various machine learning techniques. Lots of pre-processing, feature selection and classification techniques have been implemented and developed in the past decades. Following is the given work done in the prediction of Parkinson's disorders.

Machine learning algorithms

1. Navie baye's
2. Svm (support vector machine)
3. Sgd (stochastic gradient descent)

4. Gbt (gradient boosting tree)

1. Naive Baye's Classifiers

This article discusses the theory behind the Naive Bayes classifiers and their implementation.

Naive Bayes classifiers are a collection of classification algorithms based on **Baye's Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other

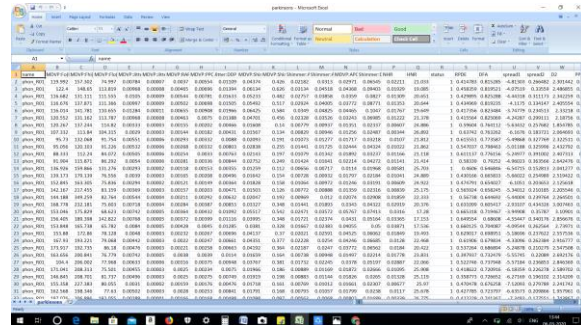


Fig 2: represent data set

To start with, let us consider a dataset.

Consider a fictional dataset that describes the features for prediction of Parkinson's disease.

The dataset is divided into two parts, namely, feature matrix and the response vector.

- Feature matrix contains all the vectors(rows) of dataset in which each vector consists of the value of dependent features.
- Response vector contains the value of class variable(prediction or output) for each row of feature matrix.

2. Support Vector Machine (svm)

I guess by now you would've accustomed yourself with linear regression and logistic regression algorithms. If not, I suggest you have a look at them before moving on to support vector machine. Support vector machine is another simple algorithm that every machine learning expert should have in his/her arsenal. Support vector

machine is highly preferred by many as it produces significant accuracy with less computation power. Support Vector Machine, abbreviated as SVM can be used for both regression and classification tasks. But, it is widely used in classification objectives.

The objective of the support vector machine algorithm is to find a hyper plane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

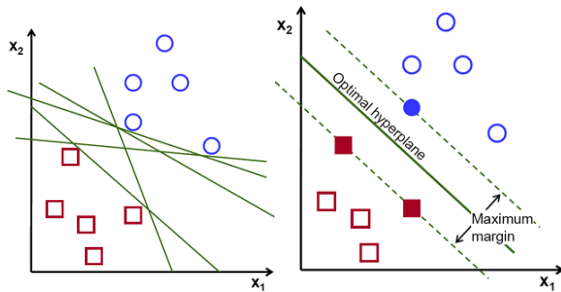


Fig 3 :Possible hyper planes

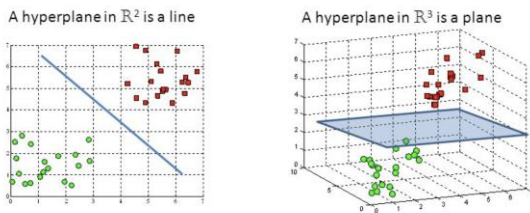


Fig 4 : Hyper planes in 2D and 3D feature space

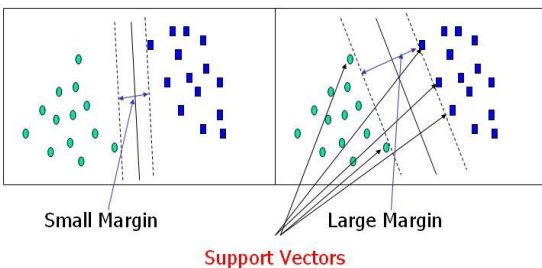


Fig 5 : represents support vector

3. Gradient Descent technique

Gradient descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient. In machine learning, we use gradient descent to update the parameters of our model. Parameters refer to coefficients in Linear Regression and weights in neural networks.

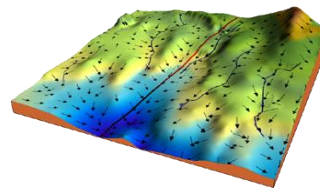
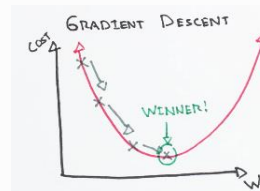


Fig 6 :shows the SGD technique



Step-by-step

Now let's run gradient descent using our new cost function. There are two parameters in our cost function we can control: w (weight) and b (bias). Since we need to consider the impact each one has on the final prediction, we need to use partial derivatives. We calculate the partial derivatives of the cost function with respect to each parameter and store the results in a gradient.

4. Gradient boosting tree

Although most of the Kaggle competition winners use stack/ensemble of various models, one particular model that is part of most of the ensembles is some variant of Gradient Boosting (GBM) algorithm. Take for an example the winner of latest Kaggle competition: Michael Jahrer's solution with representation learning in Safe Driver Prediction. His solution was a blend of 6 models. 1 LightGBM (a variant of GBM) and 5 Neural Nets. Although his

success is attributed to the semi-supervised learning that he used for the structured data, but gradient boosting model has done the useful part too. Even though GBM is being used widely, many practitioners still treat it as complex black-box algorithm and just run the models using pre-built libraries. The purpose of this post is to simplify a supposedly complex algorithm and to help the reader to understand the algorithm intuitively. I am going to explain the pure vanilla version of the gradient boosting algorithm and will share links for its different variants at the end. I have taken base DecisionTree code from **fast.ai** library (fastai/courses/ml1/lesson3-rf_foundations.ipynb) and on top of that, I have built my own simple version of basic gradient boosting model.

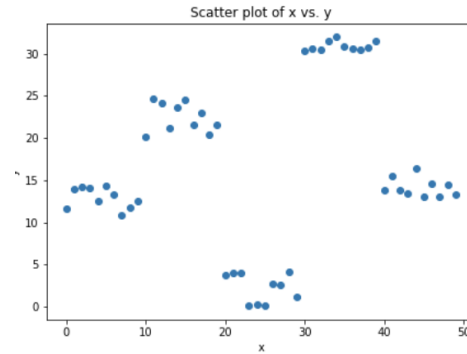


Fig 9 : Simulated data (x: input, y:output)

IV. PROPOSED SYSTEM

We have proposed new method to collect symptoms of diseases we are using data set of Parkinson disease by using some machine learning algorithms for more accuracy In prediction we used algorithms like

- a. NAVIE BAYES
- b. SVM
- c. SGD
- d. GBT

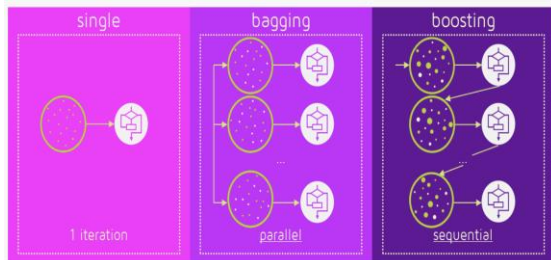


Fig 7 : represents the iteration ,parallel, sequential

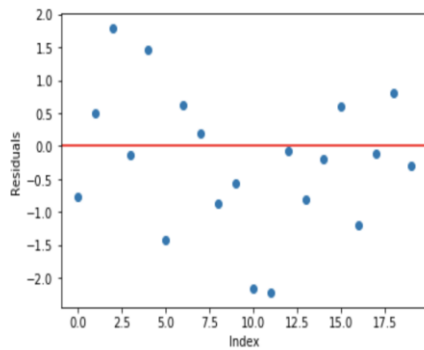


Fig 8 : Sample random normally distributed residuals with mean around 0

While compared with previous research's we have got better results those are shown below

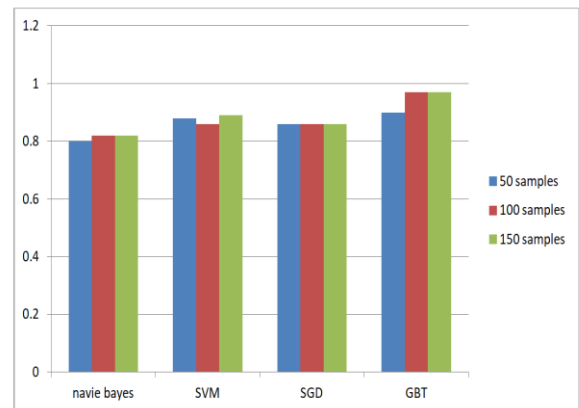


Fig 10 : represents total results of the algorithms

V. ADVANTAGES

- We have taken voice data set of Parkinson disease
- We have taken 195 patients samples 23 feature from voice data set
- ['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)', 'Shimmer:APQ3', 'Shimmer:APQ5', 'MDVP:APQ', 'Shimmer:DDA', 'NHR', 'RPDE', 'DFA', 'spread1', 'spread2', 'D2', 'PPE']
- From these features we are comparing the patients data and predicting the disease
- We are divided the prediction into 3 sub parts like 50,100,150
- In the above bar graph we can see the results
- By dividing the samples into 3 parts we can know the which algorithm is giving better output and accurate results
- In our observation (GBT) is giving best accurate results
- Time complexity is very less
- Reliability
- Portability
- By using this algorithm's we can increase life span of the patient if it is detected in initial stages

SRS (SOFTWARE REQUIREMENT SPECIFICATION)

software requirements specification (SRS) is a detailed description of a software system to be developed with its functional and non-functional requirements. The SRS is developed based the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system. The software requirement specification document consistent of all necessary requirements required for project development. To develop the software system we should have clear understanding of Software system. To achieve this we need to continuous communication with customers to gather all requirements.

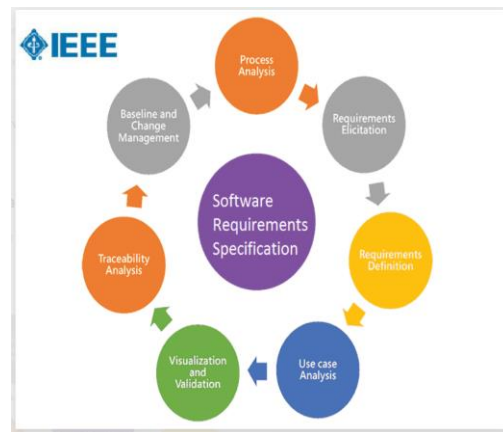


Fig 11: shows the software requirement specification

SPIRAL MODEL

Spiral model is one of the most important Software Development Life Cycle models, which provides support for Risk Handling.

In its diagrammatic representation, it looks like a spiral with many loops.

The exact number of loops of the spiral is unknown and can vary from project to project.

Each loop of the spiral is called a Phase of the software development process. The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks.

As the project manager dynamically determines the number of phases, so the project manager has an important role to develop a product using spiral model.

The Radius of the spiral at any point represents the expenses(cost) of the project so far, and the angular dimension represents the progress made so far in the current phase.

Below diagram shows the different phases of the Spiral Model:

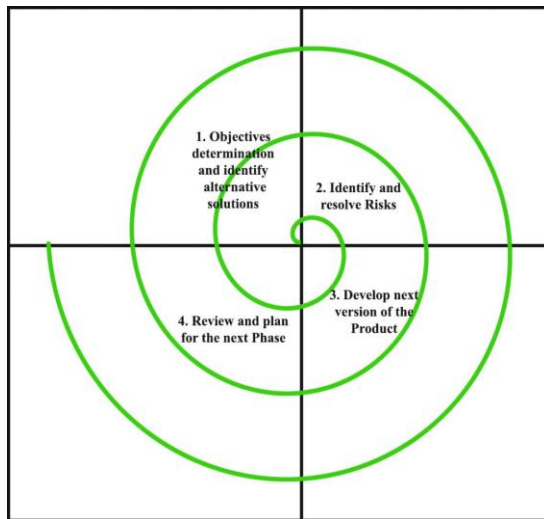


Fig 12: Shows the working of spiral model

Advantages of Spiral Model

- Below are some of the advantages of the Spiral Model.
- **Risk Handling:** The projects with many unknown risks that occur as the development proceeds, in that case, Spiral Model is the best development model to follow due to the risk analysis and risk handling at every phase.
- **Good for large projects:** It is recommended to use the Spiral Model in large and complex projects.
- **Flexibility in Requirements:** Change requests in the Requirements at later phase can be incorporated accurately by using this model.

- **Customer Satisfaction:** Customer can see the development of the product at the early phase of the software development and thus, they habituated with the system by using it before completion of the total product.

VI. FUNCTIONAL REQUIREMENTS

- A Functional Requirement (FR) is a description of the service that the software must offer.
- Functional Requirements are also called Functional Specification.
- Capturing the patient medical data as a Input .
- Handling the large-scale datasets.
- observational and multi-center study that includes early untreated Parkinson's Disease patients along with age and gender-matched healthy normal subjects, to identify progression biomarkers in Parkinson's Disease.
- Output will be the different machine learning comparative analysis graphs

NON-FUNCTIONAL REQUIREMENTS

- A non-functional requirement defines the quality attribute of a software system.
- Non-functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs.
- login for authorized users.
- data backup using SQL queries.
- It should run in offline also


```

Python 3.4 Shell
File Edit Shell Debug Options Window Help
FI score for test set: 0.8600.
Gradient Tree Boosting:
Training a GradientBoostingClassifier using a training set size of 50. . .
Trained model in 0.0025 seconds.
Make predictions in 0.0000 seconds.
FI score for training set: 1.0000.
Make predictions in 0.0000 seconds.
FI score for test set: 0.9067.
Naive Bayes:
Training a GaussianNB using a training set size of 100. . .
Trained model in 0.0000 seconds.
Make predictions in 0.0000 seconds.
FI score for training set: 0.7933.
Make predictions in 0.0000 seconds.
FI score for test set: 0.8276.
Support Vector Machines:
Training a SVC using a training set size of 100. . .
Trained model in 0.0157 seconds.
Make predictions in 0.0000 seconds.
FI score for training set: 0.8920.
Make predictions in 0.0000 seconds.
FI score for test set: 0.8848.
Stochastic Gradient Descent:
Training a SGDClassifier using a training set size of 100. . .
Trained model in 0.0000 seconds.
Make predictions in 0.0156 seconds.
FI score for training set: 0.3769.
Make predictions in 0.0000 seconds.
FI score for test set: 0.4703.
Gradient Tree Boosting:
Training a GradientBoostingClassifier using a training set size of 100. . .
Trained model in 0.1235 seconds.
Make predictions in 0.0000 seconds.
FI score for training set: 1.0000.
Make predictions in 0.0000 seconds.
FI score for test set: 0.8714.
Naive Bayes:
Training a GaussianNB using a training set size of 150. . .
Trained model in 0.0157 seconds.
Make predictions in 0.0046 seconds.
FI score for training set: 0.7568.
Make predictions in 0.0000 seconds.
FI score for test set: 0.8276.
Support Vector Machines:
Training a SVC using a training set size of 150. . .
Trained model in 0.0000 seconds.
Make predictions in 0.0156 seconds.
FI score for training set: 0.8930.
Make predictions in 0.0000 seconds.
FI score for test set: 0.8947.
Stochastic Gradient Descent:
Training a SGDClassifier using a training set size of 150. . .
Trained model in 0.0000 seconds.
Make predictions in 0.0000 seconds.
FI score for training set: 0.3591.
Make predictions in 0.0000 seconds.
FI score for test set: 0.4800.
Gradient Tree Boosting:
Training a GradientBoostingClassifier using a training set size of 150. . .
Trained model in 0.1235 seconds.
Make predictions in 0.0000 seconds.
FI score for training set: 1.0000.
Make predictions in 0.0000 seconds.
FI score for test set: 0.8714.
Tuning the model. This may take a while.....
Successfully fit a model!
The best parameters were:
{'C': 10, 'gamma': 1, 'kernel': 'poly'}
Make predictions in 0.0000 seconds.
Tuned model has a training FI score of 0.8924.
Make predictions in 0.0000 seconds.
Tuned model has a testing FI score of 0.8747.
Make predictions in 0.0025 seconds.
>>>
    
```

```

FI score for training set: 0.8930.
Make predictions in 0.0000 seconds.
FI score for test set: 0.8947.
Stochastic Gradient Descent:
Training a SGDClassifier using a training set size of 150. . .
Trained model in 0.0000 seconds.
Make predictions in 0.0000 seconds.
FI score for training set: 0.3591.
Make predictions in 0.0000 seconds.
FI score for test set: 0.4800.
Gradient Tree Boosting:
Training a GradientBoostingClassifier using a training set size of 150. . .
Trained model in 0.1235 seconds.
Make predictions in 0.0000 seconds.
FI score for training set: 1.0000.
Make predictions in 0.0000 seconds.
FI score for test set: 0.8714.
Tuning the model. This may take a while.....
Successfully fit a model!
The best parameters were:
{'C': 10, 'gamma': 1, 'kernel': 'poly'}
Make predictions in 0.0000 seconds.
Tuned model has a training FI score of 0.8924.
Make predictions in 0.0000 seconds.
Tuned model has a testing FI score of 0.8747.
Make predictions in 0.0025 seconds.
>>>
    
```

Fig 14 :HOME PAGE OF HTML

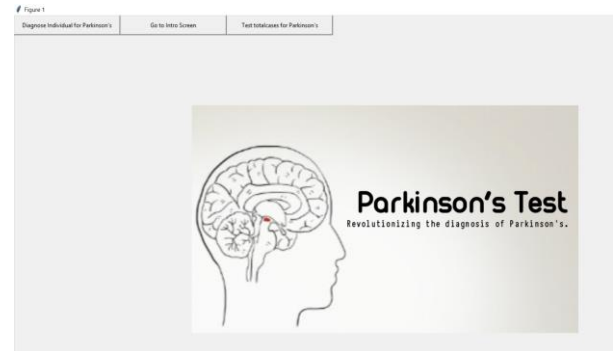
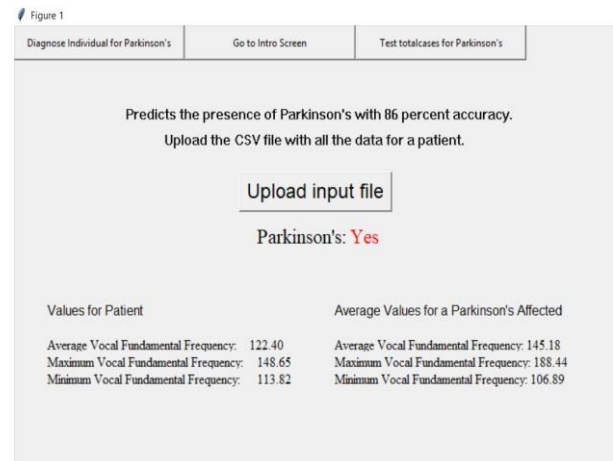


Fig 15 :TO DIAGNOSE INDIVIDUAL PERSION

If it is matched with the symptoms well get “YES” /if it is not matched we will get” NO”



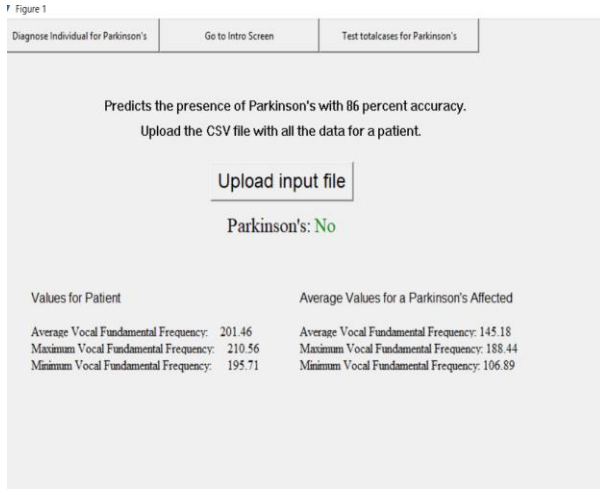


Fig 16 : UPLOAD TRAINING SET

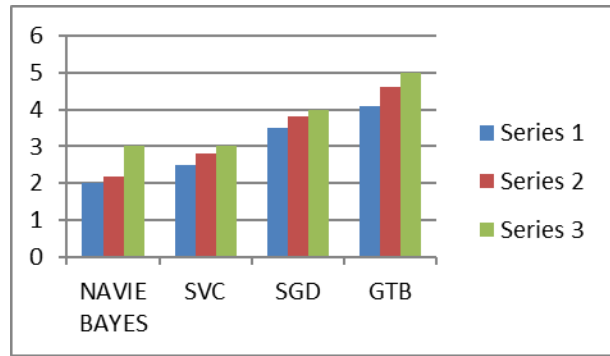


Fig 18: Generated graph

Fig 19 : CLICK ON THE UPLOAD TEST FILE

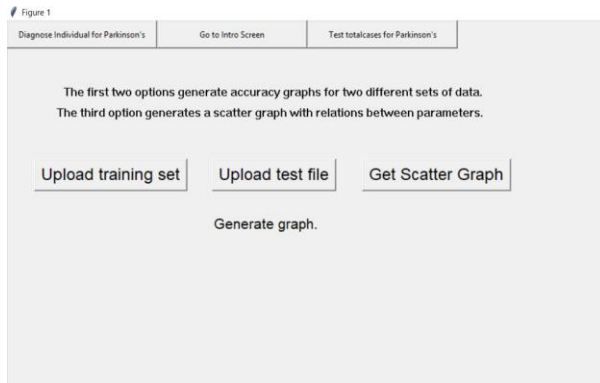


Fig 17 : TRAINING SET OUT PUT

```
Shuffling of data into test and training sets complete!
Training set: 8 samples
Test set: 45 samples
Naive Bayes:
Training a GaussianNB using a training set size of 8. . .
Support Vector Machines:
Training a SVC using a training set size of 8. . .
Stochastic Gradient Descent:
Training a SGDClassifier using a training set size of 8. . .
Gradient Tree Boosting:
Training a GradientBoostingClassifier using a training set size of 8. . .
Set score details for 100 set
Naive Bayes:
Training a GaussianNB using a training set size of 8. . .
Support Vector Machines:
Training a SVC using a training set size of 8. . .
Stochastic Gradient Descent:
Training a SGDClassifier using a training set size of 8. . .
Gradient Tree Boosting:
Training a GradientBoostingClassifier using a training set size of 8. . .
Score details for 150 set:
Naive Bayes 150 :
Training a GaussianNB using a training set size of 8. . .
Support Vector Machines:
Training a SVC using a training set size of 8. . .
Stochastic Gradient Descent:
Training a SGDClassifier using a training set size of 8. . .
Gradient Tree Boosting:
Training a GradientBoostingClassifier using a training set size of 8. . .
```

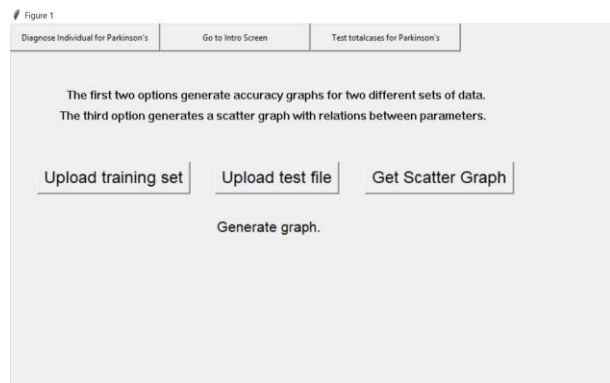


Fig 20 : TEST FILE OUTPUT

```
Gradient Tree Boosting:
Training a GradientBoostingClassifier using a training set size of 8. . .
Shuffling of data into test and training sets complete!
Training set: 150 samples
Test set: 45 samples
Naive Bayes:
Training a GaussianNB using a training set size of 50. . .
Support Vector Machines:
Training a SVC using a training set size of 50. . .
Stochastic Gradient Descent:
Training a SGDClassifier using a training set size of 50. . .
Gradient Tree Boosting:
Training a GradientBoostingClassifier using a training set size of 50. . .
Set score details for 100 set
Naive Bayes:
Training a GaussianNB using a training set size of 100. . .
Support Vector Machines:
Training a SVC using a training set size of 100. . .
Stochastic Gradient Descent:
Training a SGDClassifier using a training set size of 100. . .
Gradient Tree Boosting:
Training a GradientBoostingClassifier using a training set size of 100. . .
Score details for 150 set:
Naive Bayes 150 :
Training a GaussianNB using a training set size of 150. . .
Support Vector Machines:
Training a SVC using a training set size of 150. . .
Stochastic Gradient Descent:
Training a SGDClassifier using a training set size of 150. . .
Gradient Tree Boosting:
Training a GradientBoostingClassifier using a training set size of 150. . .
```

VII. FUTURE SCOPE

In this study we have used machine learning techniques, however very few researches have been done on machine learning algorithms. In future, the work can be extended by using auto encoders 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 auto encoder did not learn well from that but with complex dataset it would definitely give better results.

VIII. CONCLUSION

In this work, various prediction models for Parkinson’s disease detection. For this purpose four machine learning techniques i.e. are naviebaye’s , support vector machine(SVM),StochasticGradient Descent(SGD),Gradient tree boosting(GBT). To obtain the desired results, are as well as four performance metrics are seen. These four metrics are accuracy, sensitivity, ROC, specificity.

From the results, GBT outstands from all the other ML techniques with the accuracy. After that , we tried to selected the most important and minimum number of features from the speech articulation data of 195 people where we have 23 features as explained in dataset description .For that we have used feature selection techniques whose working is shown below by changing the number of features selected as it is giving the overall accuracy 96.6%, which is better in comparison to all other machine learning techniques when compared with 50,100 and 150 sub sets feature’s performance metrics.

REFERENCES

- [1] Kamal Nayan Reddy, Challa, Venkata Sasank Pagolu and Ganapati Panda, “An Improved Approach for Prediction of Parkinson’s Disease using Machine Learning Techniques”, in Proceedings of the International conference on Signal Processing, Communication, Power and Embedded System (SCOPES)-2016, pp. 1446-145, 2016.
- [2] Geeta Yadav, Yugal Kumar and G. Sahoo,

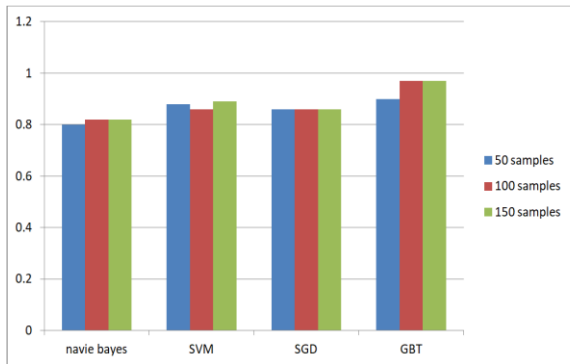


Fig 21: Generated graph

Fig 22 :CLICK ON GET SCATTER GRAPH

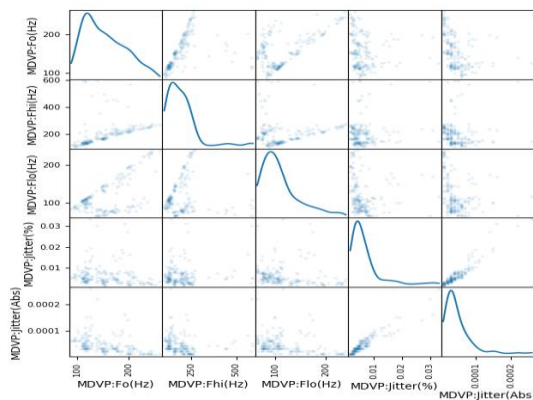
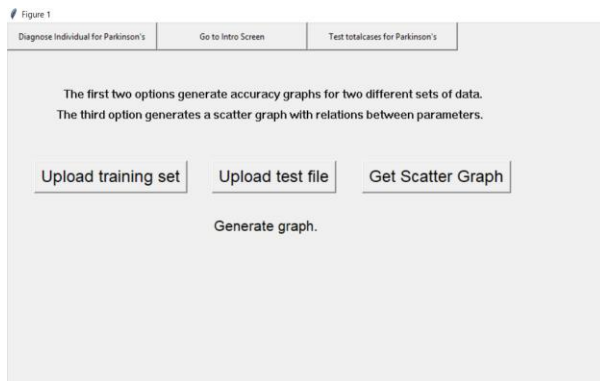


Fig 23 : Generated scatter graph

“Predication of Parkinson’s disease using Data Mining Methods: a comparative analysis of tree, statistical and support vector machine classifiers”, in Proceedings of the National Conference on Computing and Communication Systems (NCCCS), pp. 1-4, 2012.

[3] Paolo Bonato, Delsey M. Sherrill, David G. Standaert, Sara S. Salles and Metin Akay, “Data Mining Techniques to Detect Motor Fluctuations in Parkinson’s Disease”, in Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 4766-4769, 2004.

[4] Sonu S. R., Vivek Prakash and Ravi Ranjan, “Prediction of Parkinson’s Disease using Data Mining”, in Proceedings of the International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), pp. 1082-1085, 2017.

[5] Aarushi Agarwal, Spriha Chandrayan and Sitanshu S Sahu, “Prediction of Parkinson’s Disease using Speech Signal with Extreme Learning Machine”, in Proceedings of the International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), pp. 1-4, 2016.






[6] Akshaya Dinesh and Jennifer He, “Using Machine Learning to Diagnose Parkinson’s Disease from Voice Recording”, in Proceedings of the IEEE MIT Undergraduate Research Technology Conference (URTC), pp. 1-4, 2017.

[7] Giulia Fison, Emanuel Weitschek, Giovanni Felici and Paola Bertolazzi, “Alzheimer’s disease patients classification through EEG signals processing”, in Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM). pp 1-4, 2014.

[8] Pedro Miguel Rodrigues, Diamantino Freitas and Joao Paulo Teixeirab, “Alzheimer electroencephalogram temporal events detection by K-means”, in Proceedings of the International Conference on Health and Social Care Information Systems and Technologies HCIST. pp. 859 – 864, 2012.

[9] Daniel Johnstone¹, Elizabeth A. Milward¹, Regina Berrettal and Pablo Moscatol, “Multivariate Protein Signatures of Pre-Clinical Alzheimer’s Disease in the Alzheimer’s Disease Neuroimaging Initiative (ADNI) Plasma Proteome Dataset”, in Proceedings of the Disease Neuroimaging Initiative, vol-7, pp. 1-17, 2017.

AUTHOR DETAILS

| | |
|---|---|
|  | P. SRAVANTHI is presently pursuing B.Tech (CSE) Department of Computer Science Engineering from Nadimpalli satyanarayana raju institute of technology. |
|  | B.SAI SUDHA is presently pursuing B.Tech(CSE) Department of Computer Science Engineering from Nadimpalli satyanarayana raju institute of technology |
|  | M.DINESH is presently pursuing B.Tech (CSE) Department of Computer Science Engineering from Nadimpalli satyanarayana raju institute of technology |
|  | P.SAI GANESH is presently pursuing B.Tech (CSE) Department of Computer Science Engineering from Nadimpalli satyanarayana raju institute of technology |
|  | MR.T.RAVI KUMAR (M.TECH),is working as an Assistant Professor in Department of computer science and engineering in Nadimpalli satyanarayana raju institute of technology, sontyam, Visakhapatnam. |



MR.G.RAJASEKHARAM
(M.TECH,PH.D), is working as an Associate Professor,(HOD) in the Department of Computer Science and Engineering in Nadimpalli satyanarayana raju institute of technology , sontyam ,Visakhapatnam.