

PARKINSON'S DISEASE DETECTION USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements for the award of Bachelor of

Engineering Degree in Computer Science and Engineering

By

UDAYAGIRI SUSHMANTH GIRISH
(Reg. No. 38110607)

YARAGUNDLA RAJESH KUMAR REDDY
(Reg. No. 38110663)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY - 2022



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **UDAYAGIRI SUSHMANTH GIRISH (Reg.No.38110607), YARAGUNDLA RAJESH KUMAR REDDY (Reg.No.38110663)** who carried out the project entitled “**PARKINSON’S DISEASE DETECTION USING MACHINE LEARNING**” under our supervision from November 2021 to May 2022.

Internal Guide

Dr.M.S.Roobini, M.E., P.hD.,

Head of the Department

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I UDAYAGIRI SUSHMANTH GIRISH (Reg.No.38110607), YARAGUNDLA RAJESH KUMAR REDDY (Reg.No.38110663) hereby declare that project report entitled "PARKINSON'S DISEASE DETECTION USING MACHINE LEARNING"

done under the guidance of **Dr.M.S.Roobini, M.E, P.hD.**, at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

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

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

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.M.S.Roobini, M.E., P.hD.,** for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

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

ABSTRACT

Firstly, James Parkinson described Parkinson's disease as a neurological syndrome, it effects the central nervous system as a result the patients face the issue in talking, strolling, tremor during motion. Parkinson's disease patient typically has a low-volume noise with a monotone quality, this system explores the classification of audio signals feature dataset to diagnosis Parkinson's disease (PD), the classifiers we utilized in this system are from Machine Learning. The significant classification models that are effectively utilized are (K-Nearest Neighbor) KNN, Decision Tree, Logistic regression and eXtreme Gradient Boost (XGboost). The performance analysis of the Parkinson's disease datasets has a variety of features. There are 21 features of all in which only 12 features play an important role for predicting the best algorithm in the classification. The system has achieved better result in predicting the PD patient healthy or not, XGBoost provided the peak accuracy of 96% and the Matthews Correlation Coefficient (MCC) of 89%. MDVP:Jitter(%), Jitter(Abs), MDVP:RAP, MDVP:PPQ, Jitter:DPP, MDVP: Shimmer, MDVP:Shimmer(db), Shimmer APQ3,Shimmer APQ5, MDVP:APQ, Shimmer:DDA, NHR are the selected features from the dataset,which are important for the prediction to achieved high accuracy.

TABLE OF CONTENTS

S.NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF TABLES	ix
	LIST OF ABBREVIATIONS	x
1	INTRODUCTION	1
	1.1 GENERAL	1
	1.1.1 HOEHN AND YAHR SCALE	1
	1.1.2 UNIFIED PARKINSON'S DISEASE RATIN SCALE	
	1.1.2.1 INTELLECTUAL FUCTION, MOO BEHAVIOR	
	1.1.2.2 ACTIVITIES OF DAILY LIVING	4
	1.1.2.3 MOTOR EXAMINATION	4
	1.1.2.4 MOTOR COMPLICATIONS	5
2	LITERATURE SURVEY	7
3	AIM AND SCOPE OF PRESENT INVESTIGATION	10
	3.1 AIM OF THE PROJECT	10
	3.2 SCOPE AND OBJECTIVES	10
	3.3 SYSTEM REQUIREMENTS	10
	3.4 PYTHON	10
	3.5 OVERVIEW OF MACHINE LEARNING	11
	3.6 LOGISTIC REGRESSION	11
	3.6.1 TYPES OF LOGISTIC REGRESSION	12
	3.6.1.1 BINARY OR BINOMIAL	12
	3.6.1.2 MULTINOMIAL	12
	3.6.1.3 ORDINAL	12
	3.7 XGBOOST	13
	3.7.1 BAGGING	14
	3.7.2 BOOSTING	14
	3.7.3 UNIQUE FEATURES OF XGBOOST	15
	3.8 DECISION TREE	16

	3.9 K-NEAREST NEIGHBORS ALGORITHM	19
	3.10 MATTHEWS CORRELATION COEFFICIENT	24
4	DESIGN AND IMPLEMENTATION	26
	4.1 DATA PREPROCESSING	26
	4.2 FEATURE SELECTION	26
	4.3 HEAT MAP	28
	4.4 LINE GRAPH	28
	4.5 BAR CHART	28
	4.6 ARCHITECTURE	29
	4.6.1 ARCHITECTURE EXPLANATION	29
5	RESULT AND DISCUSSION	30
6	CONCLUSION AND FUTURE WORK	31
	6.1 CONCLUSION	31
	6.2 FUTURE WORK	31
	REFERENCES	34
	APPENDIX	37
	A. SOURCE CODE	37

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
3.1	Entropy function to a boolean classification, as the proportion p_+ , of positive examples varies between 0 & 1.	18
3.2	Example to understand the concept of K-nearest neighbor	20
3.3	Example to understand the concept of K-nearest neighbor	21
3.4	Boundary becoming smoother with increasing the K value	22
3.5	Training error rate with a varying value of K	23
3.6	Validation error curve with varying value of K	23
4.1	Architecture diagram of the process involved in this model	29
5.1	Relationship between data values on heat map	31
5.2	Bar chart for the classifier choice	32

LIST OF TABLES

TABLE NO	TABLE NAME	PAGE NO
3.1	Example to place KNN in a scale	20
4.1	List of Speech dataset	27
5.1	Tabular representation of Accuracy and MCC values of the various algorithms	33

LIST OF ABBREVIATIONS

PD	:	Parkinson's Disease
MCC	:	Matthews Correlation Coefficient
UPDRS	:	Unified Parkinson's Disease Rating Scale
XGBoost	:	eXtreme Gradient Boost
KNN	:	K – Nearest Neighbor
LR	:	Logistic Regression
MDVP	:	Multi-Dimensional Voice Program

1.1 GENERAL

C H A P T E R 1 I N T R O D U C T I O N

In a person who experiences health, bodily functions are working at peak performance, due not only to a lack of disease, but also to regular exercise, balanced nutrition, and adequate rest. We receive treatment, when necessary, to maintain the balance. Physical wellbeing involves pursuing a healthful lifestyle to decrease the risk of disease. Maintaining physical fitness, for example, can protect and develop the endurance of a person's breathing and heart function, muscular strength, flexibility, and body composition. Physical health and well-being also help reduce the risk of an injury or health issue. Examples include minimizing hazards in the workplace, practicing safe sex, practicing good hygiene, or avoiding the use of tobacco, alcohol, or illegal drugs.

Parkinson's disease (PD) is a progressive nervous system disorder that affects movement. Symptoms start gradually, sometimes starting with a barely noticeable tremor in just one hand. Tremors are common, but the disorder also commonly causes stiffness or slowing of movement. In the early stages of Parkinson's disease, your face may show little or no expression. Your arms may not swing when you walk. Your speech may become soft or slurred. Parkinson's disease symptoms worsen as your condition progresses over time. Although Parkinson's disease can't be cured, medications might significantly improve your symptoms. Occasionally, your doctor may suggest surgery to regulate certain regions of your brain and improve your symptoms. It can be diagnosed by a physician using the two most commonly used are the Hoehn and Yahr scale and the Unified Parkinson's Disease Rating Scale (UDPRS).

1.1.1 Hoehn and Yahr Scale:

The Hoehn and Yahr scale, named for its authors, was published in 1967 and was the first rating scale to describe the progression of PD. The Hoehn and Yahr scale describe five stages to PD progression:

- a. **Stage One of Parkinson's Disease** – In stage one, the earliest stage, the symptoms of PD are mild and only seen on one side of the body (unilateral involvement), and there is usually minimal or no functional impairment. The symptoms of PD at stage one may be so mild that the person doesn't seek medical attention or the physician is unable to make a diagnosis. Symptoms at stage one may include tremor, rigidity, or slowness of movement in the arm or leg on one side of the body, or one side of the face may be affected, impacting the expression.
- b. **Stage Two of Parkinson's Disease** – Stage two is still considered early disease in PD, and it is characterized by symptoms on both sides of the body (bilateral involvement) or at the midline without impairment to balance. Stage two may develop months or years after stage one. Symptoms of PD in stage two may include the loss of facial expression on both sides of the face and decreased blinking. Speech abnormalities may be present, such as soft voice, monotone voice, fading volume after starting to speak loudly, or slurring speech. There also may be stiffness or rigidity of the muscles in the trunk that may result in neck or back pain, stooped posture, and general slowness in all activities of daily living. Diagnosis may be easy at this stage if the patient has a tremor, however, if stage one was missed and the only symptoms of stage two are slowness or lack of spontaneous movement, PD could be misinterpreted as only advancing age.
- c. **Stage Three of Parkinson's Disease** – Stage three is considered mid-stage and is characterized by loss of balance and slowness of movement. Balance is compromised by the inability to make the rapid, automatic, and involuntary adjustments necessary to prevent falling, and falls are common at this stage. All other symptoms of PD are also present at this stage, and generally diagnosis is not in doubt at stage three. An important clarifying factor of stage three is that the patient is still fully independent in their daily living activities, such as dressing, hygiene, and eating.
- d. **Stage Four of Parkinson's Disease** – In stage four, PD has progressed to a severely disabling disease. Patients with stage four PD may be able to walk

and stand unassisted, but they are noticeably incapacitated. Many use a walker to help them. At this stage, the patient is unable to live an independent life and needs assistance with some activities of daily living.

- e. **Stage Five of Parkinson's Disease** – Stage five is the most advanced and is characterized by confinement to a bed or wheelchair. People with stage five PD may be unable to rise from a chair or get out of bed without help, they may have a tendency to fall when standing or turning, and they may freeze or stumble when walking. Around-the-clock assistance is required at this stage to reduce the risk of falling and help the patient with all daily activities. At stage five, the patient may also experience hallucinations or delusions.

While the symptoms worsen over time, it is worth noting that some patients with PD never reach stage five. People with PD may also never experience some of the above symptoms. In addition, there are treatments available that can help at every stage of the disease.

1.1.2 UNIFIED PARKINSON'S DISEASE RATING SCALE

The Unified Parkinson's Disease Rating Scale (UPDRS) has four parts. Each part has multiple points that are individually scored, using zero for normal or no problems, 1 for minimal problems, 2 for mild problems, 3 for moderate problems, and 4 for severe problems. These scores are tallied to indicate the severity of the disease, with 199 points being the worst and total disability and 0 meaning no disability. In 2001, the Movement Disorder Society (MDS) updated the rating scale with involvement from patients and caregivers. The updated scale is referred to as the UPDRS-MDS, and it was published in 2008. It now includes the following sections:

1.1.2.1 INTELLECTUAL FUNCTION, MOOD, BEHAVIOR

- Forgetfulness, disorientation in time and space
- Vivid dreaming
- Hallucinations
- Delusions and paranoia

- Depressed mood
- Anxious mood
- Apathy
- Features of dopamine dysregulation syndrome
- Nighttime sleep problems
- Daytime sleepiness
- Pain and other sensations
- Urinary problems
- Constipation problems
- Lightheadedness on standing
- Fatigue

1.1.2.2 ACTIVITIES OF DAILY LIVING

- Speech: difficulty being understood
- Salivation and drooling
- Chewing and swallowing
- Cutting food
- Small handwriting
- Needing help with getting dressed, buttons, arms in sleeves
- Requires assistance with bathing, brushing teeth
- Trouble doing hobbies and other activities
- Difficulties with turning in bed
- Tremor impact on activities
- Getting in and out of bed
- Walking, balance, falling
- Freezing

1.1.2.3 MOTOR EXAMINATION

- Speech – volume, diction
- Reduced facial expressions
- Rigidity

- Finger tapping
- Slowed hand movements
- Rapid alternating movements of hands (pronation-supination)
- Toe tapping
- Leg agility – when tapping heel on the ground, is it slowed, early fatiguing
- Arising from a chair – degree of difficulty
- Gait – shuffling, walking with difficulty
- Freezing of gait
- Postural stability – difficulty recovering balance
- Posture – stooped
- Global spontaneity of movement (body bradykinesia) – slowness of movement, lack of movement
- Tremor at rest

1.1.2.4 MOTOR COMPLICATIONS

- Dyskinesia, including time spent with dyskinesia, functional impact of dyskinesia, and painful off-state dystonia
- Motor fluctuations, including time spent in the off state, functional impact of fluctuations, and complexity of motor fluctuations

As speech of PD patient have change in the frequency spectre in there voice because they loss the control of the limb, which decrease the frequency of the audio. So, the low frequency region gives important data to differentiate the speech impairments in PD. Unified Parkinson disease rating scale (UPDRS) is used to find the severity of the PD by help of clinical expertise and experience.

We perform a feature selection for the audio features dataset created by Max Little of the University of Oxford, high prediction has been achieved with classification accuracy, algorithm predict various accuracy for various variables that are relevant on the other attributed present in the feature dataset, as feature plays important role the dataset which we taken from UCI repository contains 21 features and applied a pearson's correlation coefficient on feature to determine the coefficient correlation

among features. This section presents the comparative determination endeavors, here both model-based and model-free techniques algorithms are used for predicting parkinson's disease. Model-based approach heavily depend on the priori statistical statements, such as relation between variables. Most commonly used model-based tool is Logistic regression, which it measure the outcome on a binary scale(e.g healthy/not), here classification process carried out based on the estimation probabilitie. Where as model-free methods like XGBoost adapt to the intrinsic data characteristic without any priori model. we used XGBoost algorithm for classification, XGBoost algorithm benefit from caonstant learning or retraining, they don't guarante optimized classification/regression.

However, when trained and maintained, XGBoost learning method have great potential than Logistic regression in solving real world problems. The prior report of using XGBoost technique to diagonose parkinson's disease are determines according to their cognitive statues. XGBoost provided an accuracy of 96% for classification the dataset and logistic regression provided an accuracy of 79%, this system that predict PD has been formulated which compares the accuracy of LR and XGBoost on the train and test dataset. It utilized co-efficient correlation to find the correlation among features, on comparison it provided that XGBoost performed better than LR with accuracy of 96 percent.

CHAPTER 2

LITERATURE

SURVEY

Mohammad S Islam.Et Al Conducted A Comparative Analysis To Detect Parkinson's Disease Using Various Classifiers. Support Vector Machine (SVM), Feedforward Back-Propagation Based Artificial Neural Network (FBANN) And Random Tree (RT) Classifiers Were Used And A Comparison Between Them Is Made To Differentiate Between PD And Healthy Patients. The Study Has Utilized The UCI Machine Learning Repository. The Dataset Consists Of 195 Voice Samples From 31 Individuals Comprising Of Both Males And Females. From The Taken Subjects 23 Were Determined With PD And 8 Were Healthy. To Improve The Classification Accuracy With Minimal Error Rate A 10-Fold Cross Validation Which Was Repeated

100 Times Has Been Implemented For All The Three Classifiers. The FBANN Classifier Has Achieved A 97.37% Recognition Accuracy Thus Outperforming The Other Two Classifiers

R. Arefi Shirvan.Et Al Proposed A System For Detect PD. The Data Classification Was Done Using Knn Method. Simplest Method In Grouping The Similarity Is KNN. Among Classification Method KNN Is Used Whenever The Facts For Data Distribution Are Not Enough. In This Method It Has Two Parts: A) Determine K Close Neighbors, B) Determining Class Type Using These Close Neighbors. It Was Shown That A 93.7% Of Accuracy Per 4 Optimized Features, An Accuracy Of 94.8% Per 7 Optimized Features And 98.2% Accuracy For 9 Optimized Features Is Achieved Which Is A Remarkable Result Compared To Other Studies. In This Research Data From [6] From UCI Repository Is Used. The Data Include 192 Voice Sample Recordings From 32 Male And Female. Each Subject Has Had 6 Voice Signal Recordings. 23 People Suffer From PD And The Rest Are Healthy. People Were About 46-85 Years Old Themain Disadvantage Of The KNN algorithm Is That It Is A Lazy Learner, I.E. Classification Is Done By Using Training Data And From The Training Data It Doesn't Learn Anything

A Comparative Study on Tumor Classification research done by K. Srilatha and V. Ulagamuthalvi.Et Al, classification is an important task within the field of

computer vision. Image classification refers to the labelling of images into one of a number of predefined categories that includes image sensors, image pre-processing, object detection, object segmentation, feature extraction and object classification. Many classification techniques have been developed for image classification. The highest concentration is on using various classifiers combined with several segmentation algorithms for detection of tumor using image processing.

S. R. Khonde & V. Ulagamuthalvi. Et Al proposed Feature selection used is based on average probability score of each feature. The features having less AP score are removed from the set used for training and testing classifiers. Performance parameters used by authors are true positive, true negative and accuracy. Authors make use of various semi-supervised classifiers for intrusion detection. All classifiers used NSL KDD dataset for intrusion detection.

Resul Das .Et Al Used Various Classification Methods For Identifying PD. Four Different Classification Techniques Were Implemented And Analyzed And They Are Dmneural, Neural Networks, Regression, And Decision Tree Respectively. Various Evaluation Methods Were Used For Calculating The Performance Of The Classifiers. After Evaluation From The Results, The Neural Networks Classifier Yielded The Best Results. The Input Dataset Was Randomly Partitioned Into Train And Test Dataset. 65% Of The Input Dataset Was Used For Training And The Rest Of The Data-Set Was Used For Testing. The Adjustable Parameters Of Each Classifier Were Tuned. For The Neural Networks Classifier, The BPNN Algorithm Has Been Used In The Feed-Forward, Single Hidden Layer Neural Network. The Algorithm Used In The Study Is The Levenberg– Marquardt (Lm) Algorithms. The Neural Network Has 92.9% Accuracy.

Mercy Paul selvam. Et Al used a machine learning techniques for predicting student dropout using data mining. In this model decision tree was used to predict the dropout in student and they obtained an accuracy with 97.69% and the prediction was done by using various parapets, which are considered for every student.

Dr. R. Geetha Ramani. Et Al Has Proposed A System To Classify PD And Non- PD Patients By The Following Methods Binary Logistic Regression, Linear

Discriminant Analysis (LDA), Partial Least Square Regression (PLS) , Random Tree (Rnd Tree) And Support Vector Machine (SVM). The Dataset Is The Parkinson's Disease Data Acquired From The UCI Repository. The Training Dataset Comprises Of 197 Samples With 22 Features Extracted From The Patients. Fisher Filtering Feature Selection Algorithm Was Found To Be An Effective Feature Ranking System. The Rnd Tree Algorithm Achieved 100% Classification Accuracy While The Lda, C4.5, Cs-Mc4 And K-Nn Yielded Accuracy Results Greater Than 90%. Among All, The C-Pls Algorithm Achieved The Least Accuracy Of 69.74%.

The Multi-Layer Perceptron (MLP) With Back-Propagation Learning Algorithm Is Used By Anchana khemphila Et Al., Et Al To Effectively Classify And Diagnosis PD. The Study Has Utilized The UCI Machine Learning Repository's Parkinson's Disease Dataset. Experiments Were Done And Information Gain Was Calculated Using Weka

3.6.6 Tool. They Used Information Gain To filter Features And Hence Need Not Be Taken From Patients. To Calculate Information Gain, The Input Must Be Discrete Numbers. Because Continuous Real Numbers Were Used In The Experiment, The Range Of Values Was Partitioned To Obtain Discretization. The Accuracy Obtained In The Training Data Set Is 91.453% And The Validation Data Set Is 80.769% By Utilizing The Main Model.

CHAPTER 3

AIM AND SCOPE OF PRESENT INVESTIGATION

3.1 AIM OF THE PROJECT

The aim of the project is to find the important features from the voice feature dataset and provide best algorithm in machine learning to detect the Parkinson's disease, to display which algorithm provides the high accuracy of prediction for the Parkinson's disease dataset.

3.2 SCOPE AND OBJECTIVES

The scope of the project is to select the important features from the voice feature dataset which are very useful for detection and splitting the selected features dataset train and test for the classification. Here classification perform on four machine learning algorithms to determine which algorithm provides best result in detection, after training and testing the model, test the model with new dataset to predict the high accuracy and after evaluating all the performance of all four algorithms showing best algorithm for the PD dataset for detecting Parkinson's disease.

3.3 SYSTEM REQUIREMENTS

Software requirements

1. Python
2. Jupiter Notebook

3.4 PYTHON

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. When you write programs in Python, you never need to bother about low-level details such as managing the memory used by your program.

3.5 OVERVIEW OF MACHINE LEARNING

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all needed steps to solve the problem in hand. So on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than have human programmers specify every needed step.

The discipline of machine learning employs various approaches to help computers learn to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithm(s) it uses to determine correct answers. For example, to train a system for the task of digital character recognition, the MNIST dataset has often been used.

3.6 LOGISTIC REGRESSION

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts $P(Y=1)$ as a function of X . It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

3.6.1 TYPES OF LOGISTIC REGRESSION

Generally, logistic regression means binary logistic regression having binary target variables, but there can be two more categories of target variables that can be predicted by it. Based on those number of categories, Logistic regression can be divided into following types –

3.6.1.1 BINARY OR BINOMIAL

In such a kind of classification, a dependent variable will have only two possible types either 1 and 0. For example, these variables may represent success or failure, yes or no, win or loss etc.

3.6.1.2 MULTINOMIAL

In such a kind of classification, dependent variable can have 3 or more possible **unordered** types or the types having no quantitative significance. For example, these variables may represent “Type A” or “Type B” or “Type C”.

3.6.1.3 ORDINAL

In such a kind of classification, dependent variable can have 3 or more possible **ordered** types or the types having a quantitative significance. For example, these variables may represent “poor” or “good”, “very good”, “Excellent” and each category can have the scores like 0,1,2,3.

Logistic Regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous. It is used to explain the relationship between one or more independent variables and one dependent binary variable, the dependent variable must be binary in nature. They shouldn't be high correlation among the prediction, this can be assessed by a correlation matrix, here the outcome has two classes, Logistic regression starts with different model setup than linear regression instead of modeling Y as a function of X directly, we model the probability that Y is

equal to class 1, gives X. First, abbreviation $P(X)=P(Y=1/X)$. Then the logistic model is:

$$p(x) = \frac{\exp(\beta^T x)}{1 + \exp(\beta^T x)}$$

$\exp(\beta^T x)$ i.e., plot $(e^a/1 + e^a)$ as a function of a

$$\log \left(\frac{p(x)}{1-p(x)} \right) = \beta^T x$$

Where

$$\beta^T x = \beta_0 + \beta_1 x_{i2} + \beta_2 x_{i3} + \dots + \beta_p x_{in}$$

3.7 XGBoost

XGBoost is a boosting algorithm, it is statistical learning method and derived from gradient boosting decision tree, it has better performance and optimization. The reason why we used XGBoost is it has good efficiency and feasibility, XGBoost allows dense and sparse matrix as the input and a numeric vector uses integer starting from 0 for classification, we can add number of iteration to the model A dataset with of n

samples and d features of every sample then s_k is the prediction from decision tree.

$$D = \{(x_1, y_1)\} (|D| = n \times 1 \in R^d, y_1 \in R)$$

The prediction score of each individual are summed up to get the final score.

Mathematically, our model in the form

$$\hat{y} = \phi(x_i) = \sum_{k=1}^k s_k(x_i), s_k \in s$$

Where k is number of trees, s is function in function space s

The objective function to be optimized is given by

$$L(\phi) = \sum_i l(\hat{y}, y_1) + \sum_k \Omega(F_k)$$

Here, l is loss function and it measure the error in prediction

3.7.1 BAGGING

While decision trees are one of the most easily interpretable models, they exhibit highly variable behavior. Consider a single training dataset that we randomly

split into two parts. Now, let's use each part to train a decision tree in order to obtain two models.

When we fit both these models, they would yield different results. Decision trees are said to be associated with high variance due to this behavior. Bagging or boosting aggregation helps to reduce the variance in any learner. Several decision trees which are generated in parallel, form the base learners of bagging technique. Data sampled with replacement is fed to these learners for training. The final prediction is the averaged output from all the learners.

3.7.2 BOOSTING

In boosting, the trees are built sequentially such that each subsequent tree aims to reduce the errors of the previous tree. Each tree learns from its predecessors and updates the residual errors. Hence, the tree that grows next in the sequence will learn from an updated version of the residuals.

The base learners in boosting are weak learners in which the bias is high, and the predictive power is just a tad better than random guessing. Each of these weak learners contributes some vital information for prediction, enabling the boosting technique to produce a strong learner by effectively combining these weak learners. The final strong learner brings down both the bias and the variance.

In contrast to bagging techniques like Random Forest, in which trees are grown to their maximum extent, boosting makes use of trees with fewer splits. Such small trees, which are not very deep, are highly interpretable. Parameters like the number of trees or iterations, the rate at which the gradient boosting learns, and the depth of the tree, could be optimally selected through validation techniques like k-fold cross validation. Having a large number of trees might lead to overfitting. So, it is necessary to carefully choose the stopping criteria for boosting.

The boosting ensemble technique consists of three simple steps:

- An initial model F_0 is defined to predict the target variable y . This model will be associated with a residual $(y - F_0)$
- A new model h_1 is fit to the residuals from the previous step
- Now, F_0 and h_1 are combined to give F_1 , the boosted version of F_0 . The mean squared error from F_1 will be lower than that from F_0 :

$$F_1(x) <- F_0(x) + h_1(x)$$

To improve the performance of F_1 , we could model after the residuals of F_1 and create a new model F_2 :

$$F_2(x) <- F_1(x) + h_2(x)$$

This can be done for ' m ' iterations, until residuals have been minimized as much as possible:

$$F_m(x) <- F_{m-1}(x) + h_m(x)$$

Here, the additive learners do not disturb the functions created in the previous steps. Instead, they impart information of their own to bring down the errors.

3.7.3 UNIQUE FEATURES OF XGBOOST

XGBoost is a popular implementation of gradient boosting. Let's discuss some features of XGBoost that make it so interesting.

- **Regularization:** XGBoost has an option to penalize complex models through both L1 and L2 regularization. Regularization helps in preventing overfitting
- **Handling sparse data:** Missing values or data processing steps like one-hot encoding make data sparse. XGBoost incorporates a sparsity-aware split finding algorithm to handle different types of sparsity patterns in the data
- **Weighted quantile sketch:** Most existing tree based algorithms can find the split points when the data points are of equal weights (using quantile

sketch

algorithm). However, they are not equipped to handle weighted data. XGBoost has a distributed weighted quantile sketch algorithm to effectively handle weighted data

- **Block structure for parallel learning:** For faster computing, XGBoost can make use of multiple cores on the CPU. This is possible because of a block structure in its system design. Data is sorted and stored in in-memory units called blocks. Unlike other algorithms, this enables the data layout to be reused by subsequent iterations, instead of computing it again. This feature also serves useful for steps like split finding and column sub-sampling
- **Cache awareness:** In XGBoost, non-continuous memory access is required to get the gradient statistics by row index. Hence, XGBoost has been designed to make optimal use of hardware. This is done by allocating internal buffers in each thread, where the gradient statistics can be stored
- **Out-of-core computing:** This feature optimizes the available disk space and maximizes its usage when handling huge datasets that do not fit into memory

3.8 DECISION TREE

Decision Tree Analysis is a general, predictive modelling tool that has applications spanning a number of different areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

The decision rules are generally in form of if-then-else statements. The deeper the tree, the more complex the rules and fitter the model.

Decision trees classify the examples by sorting them down the tree from the root to some leaf node, with the leaf node providing the classification to the example. Each node in the tree acts as a test case for some attribute, and each edge descending from that node corresponds to one of the possible answers to the test

case. This process is recursive in nature and is repeated for every subtree rooted at the new nodes.

The basic algorithm used in decision trees is known as the ID3 (by Quinlan) algorithm. The ID3 algorithm builds decision trees using a top-down, greedy approach. Briefly, the steps to the algorithm are: - Select the best attribute $\rightarrow A$ - Assign A as the decision attribute (test case) for the node. - For each value of A, create a new descendant of the node. - Sort the training examples to the appropriate descendant node leaf. - If examples are perfectly classified, then STOP else iterate over the new leaf nodes.

To define information gain precisely, we need to define a measure commonly used in information theory called *entropy* that measures the level of *impurity* in a group of examples. Mathematically, it is defined as:

$$\text{Entropy: } \sum_{i=1}^n -p_i \log_2(p_i)$$

Since, the basic version of the ID3 algorithm deal with the case where classification are either positive or negative, we can define entropy as:

$$\text{Entropy}(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

where,

S is a sample of training examples

p_+ is the proportion of positive examples in S

p_- is the proportion of negative examples in S

To illustrate, suppose S is a sample containing 14 boolean examples, with 9 positive and 5 negative examples. Then, the entropy of S relative to this boolean classification is:

$$\text{Entropy}([9+, 5-]) = -(9/14) \log_2(9/14) - (5/14) \log_2(5/14) = 0.940$$

Note that entropy is 0 if all the members of S belong to the same class. For example, if all members are positive ($p_+ = 1$), then p_- is 0, and $Entropy(S) = -1 \cdot \log_2(1) - 0 \cdot \log_2(0) = 0$. Entropy is 1 when the sample contains an equal number of positive and negative examples. If the sample contains unequal number of positive and negative examples, entropy is between 0 and 1. The following figure shows the form of the entropy function relative to a boolean classification as p_+ varies between 0 and 1.

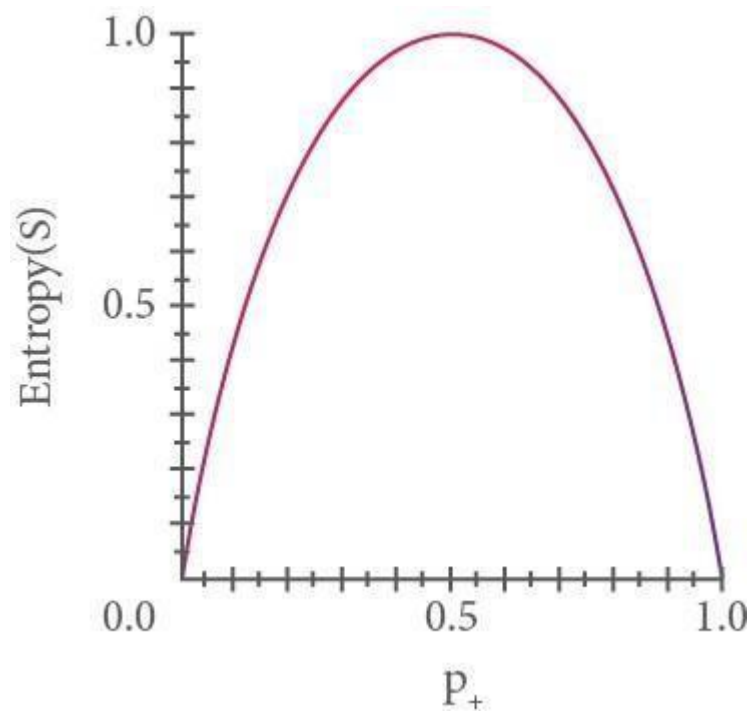


Fig 3.1: Entropy function to a boolean classification, as the proportion p_+ , of positive examples varies between 0 & 1.

Now, given entropy as a measure of the impurity in a sample of training examples, we can now define *information gain* as a measure of the effectiveness of an attribute in classifying the training data. Information gain, $Gain(S, A)$ of an attribute A , relative to a sample of examples S , is defined as:

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \cdot Entropy(S_v)$$

Where $\text{Values}(A)$ is the set of all possible values for attribute the A , and S_v is the subset of S for which attribute A has value v . Note the first term in the equation is just *entropy* of the original sample S , and the second term is the expected value of entropy after S is partitioned using attribute A , i.e. entropy of its children. Expected entropy described by this second term is simply the sum of entropies of each subset

S_v , weighted by the fraction of examples $\frac{|S_v|}{|S|}$ that belong to S_v . $\text{Gain}(S, A)$ is therefore the expected reduction in *entropy* caused by knowing the value of attribute A .

In short: $\text{InformationGain} = \text{Entropy}(\text{parentnode}) - [\text{AverageEntropy}(\text{children})]$

3.9 K-NEAREST NEIGHBORS ALGORITHM (KNN)

A k-nearest-neighbor algorithm, often abbreviated k-nn, is an approach to data classification that estimates how likely a data point is to be a member of one group or the other depending on what group the data points nearest to it are in.

The k-nearest-neighbor is an example of a "lazy learner" algorithm, meaning that it does not build a model using the training set until a query of the data set is performed.

A k-nearest-neighbor is a data classification algorithm that attempts to determine what group a data point is in by looking at the data points around it.

An algorithm, looking at one point on a grid, trying to determine if a point is in group A or B, looks at the states of the points that are near it. The range is arbitrarily determined, but the point is to take a sample of the data. If the majority of the points are in group A, then it is likely that the data point in question will be A rather than B, and vice versa.

The k-nearest-neighbor is an example of a "lazy learner" algorithm because it does not generate a model of the data set beforehand. The only calculations it makes are when it is asked to poll the data point's neighbors. This makes k-nn very easy to implement for data mining.

It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as

opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data).

KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. To evaluate any technique, we generally look at 3 important aspects:

1. Ease to interpret output
2. Calculation time
3. Predictive Power

Table 3.1: Example to place KNN in a scale

	Logistic Regression	CART	Random Forest	KNN
1. Ease to interpret output	2	3	1	3
2. Calculation time	3	2	1	3
3. Predictive Power	2	2	3	2

KNN algorithm fares across all parameters of considerations. It is commonly used for its easy of interpretation and low calculation time.

Let's take a simple case to understand this algorithm. Following is a spread of red circles (RC) and green squares (GS):

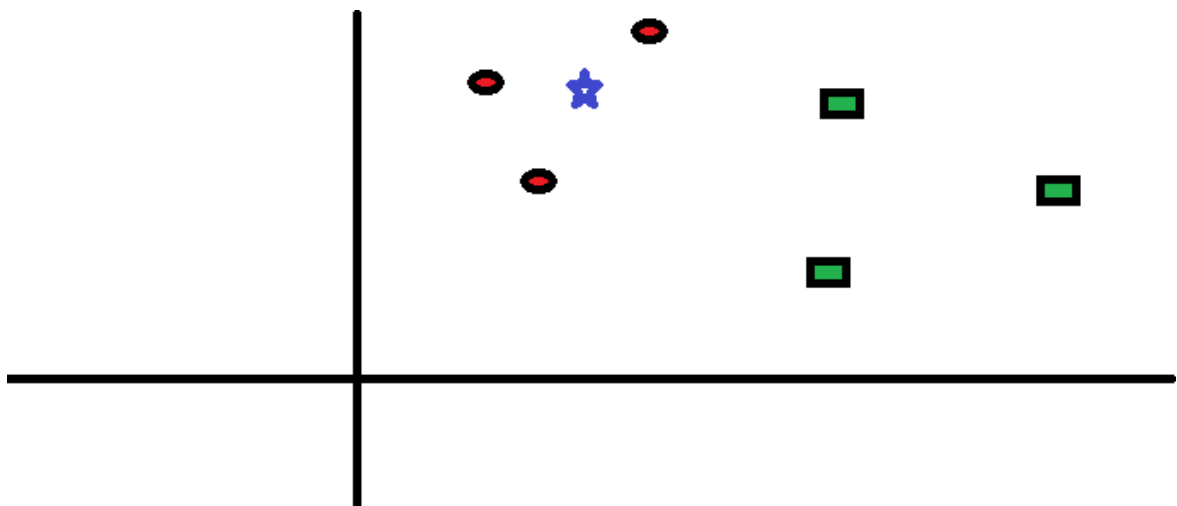


Fig 3.2: Example to understand the concept of K-nearest neighbor

You intend to find out the class of the blue star (BS). BS can either be RC or GS and nothing else. The “K” is KNN algorithm is the nearest neighbor we wish to take the vote from. Let’s say $K = 3$. Hence, we will now make a circle with BS as the center just as big as to enclose only three datapoints on the plane. Refer to the following diagram for more details:

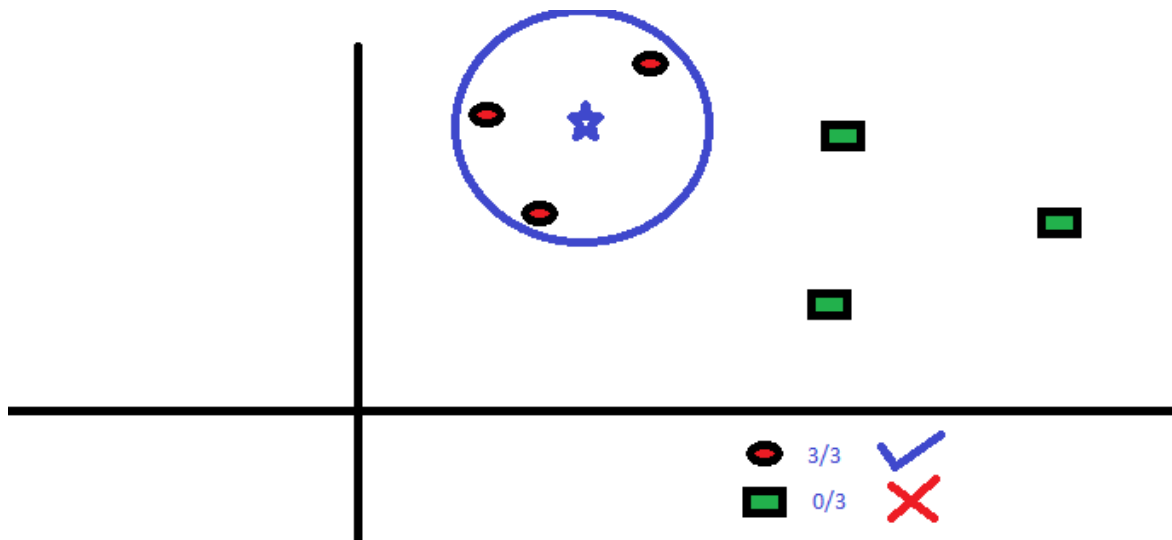


Fig 3.3: Example to understand the concept of K-nearest neighbor

The three closest points to BS is all RC. Hence, with a good confidence level, we can say that the BS should belong to the class RC. Here, the choice became very obvious as all three votes from the closest neighbor went to RC. The choice of the parameter K is very crucial in this algorithm. Next, we will understand what are the factors to be considered to conclude the best K.

First let us try to understand what exactly does K influence in the algorithm. If we see the last example, given that all the 6 training observation remain constant, with a given K value we can make boundaries of each class. These boundaries will segregate RC from GS. In the same way, let’s try to see the effect of value “K” on the class boundaries. The following are the different boundaries separating the two classes with different values of K.

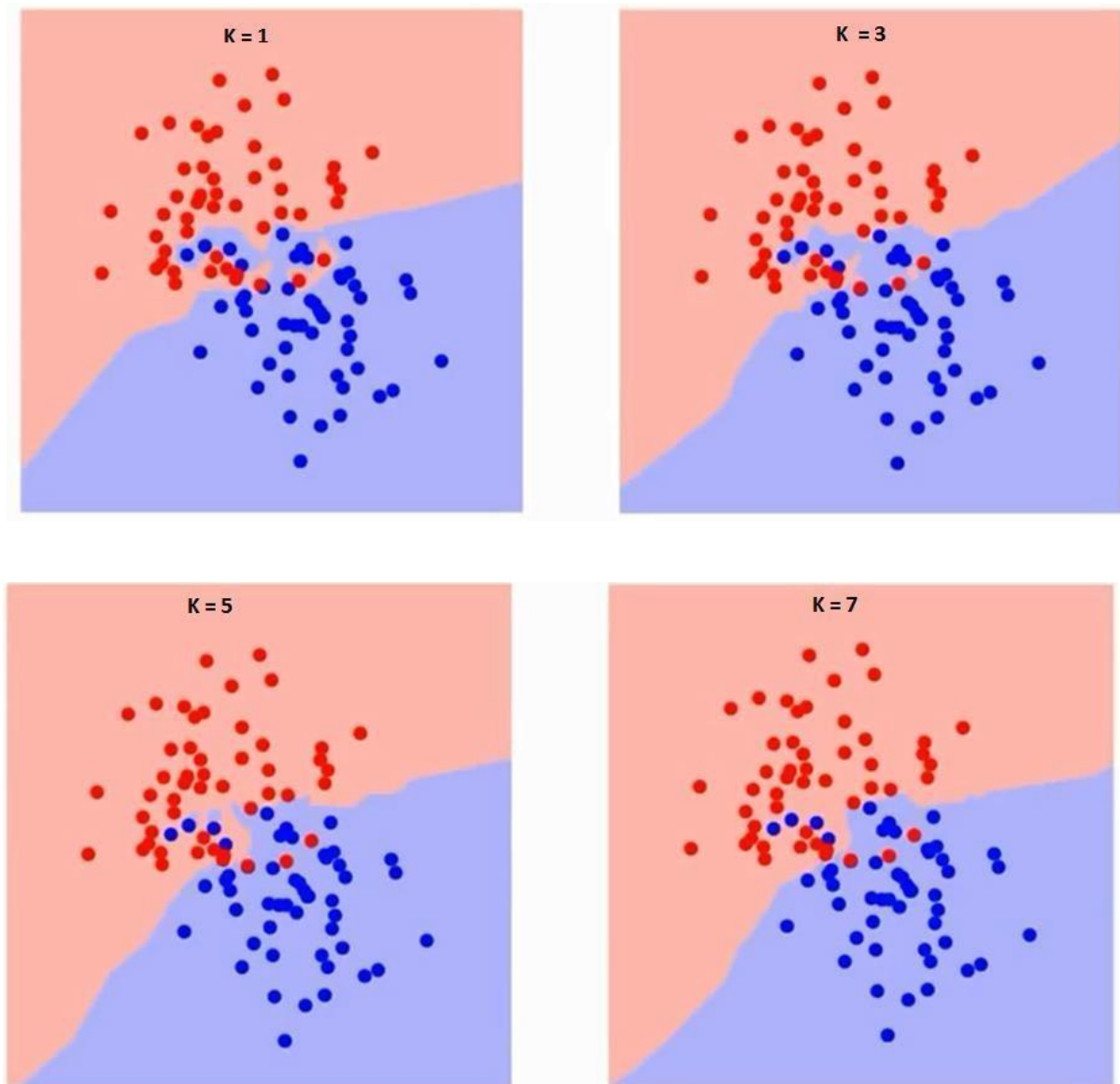


Fig 3.4: Boundary becoming smoother with increasing the K value

If you watch carefully, you can see that the boundary becomes smoother with increasing value of K. With K increasing to infinity it finally becomes all blue or all red depending on the total majority. The training error rate and the validation error rate are two parameters we need to access different K-value. Following is the curve for the training error rate with a varying value of K :

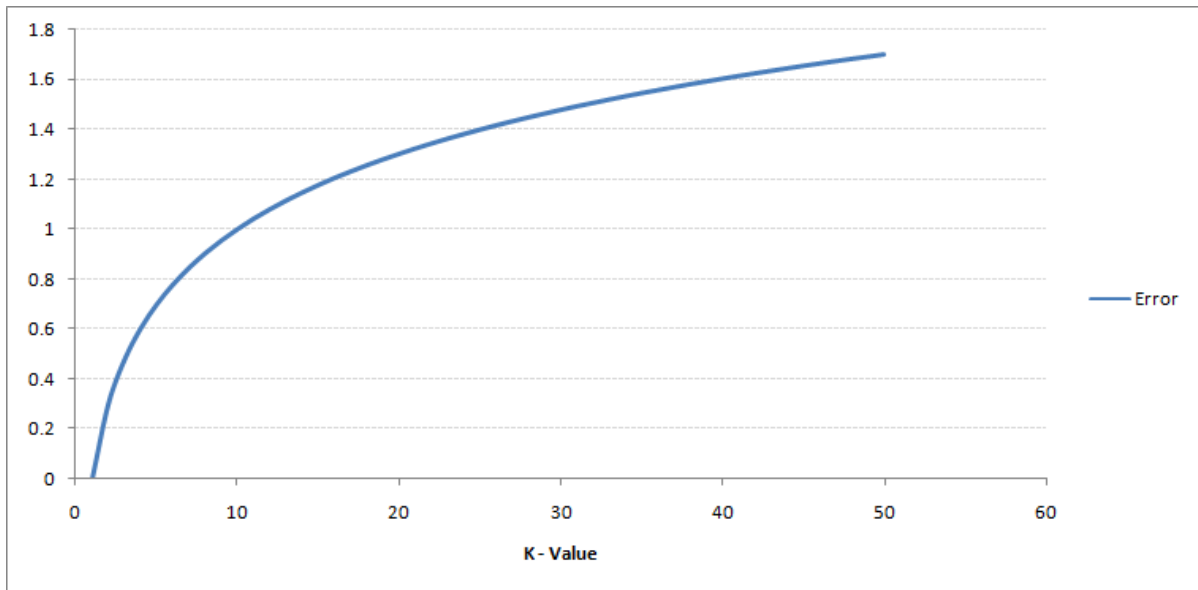


Fig 3.5: Training error rate with a varying value of K

As you can see, the error rate at $K=1$ is always zero for the training sample. This is because the closest point to any training data point is itself. Hence the prediction is always accurate with $K=1$. If validation error curve would have been similar, our choice of K would have been 1. Following is the validation error curve with varying value of K:

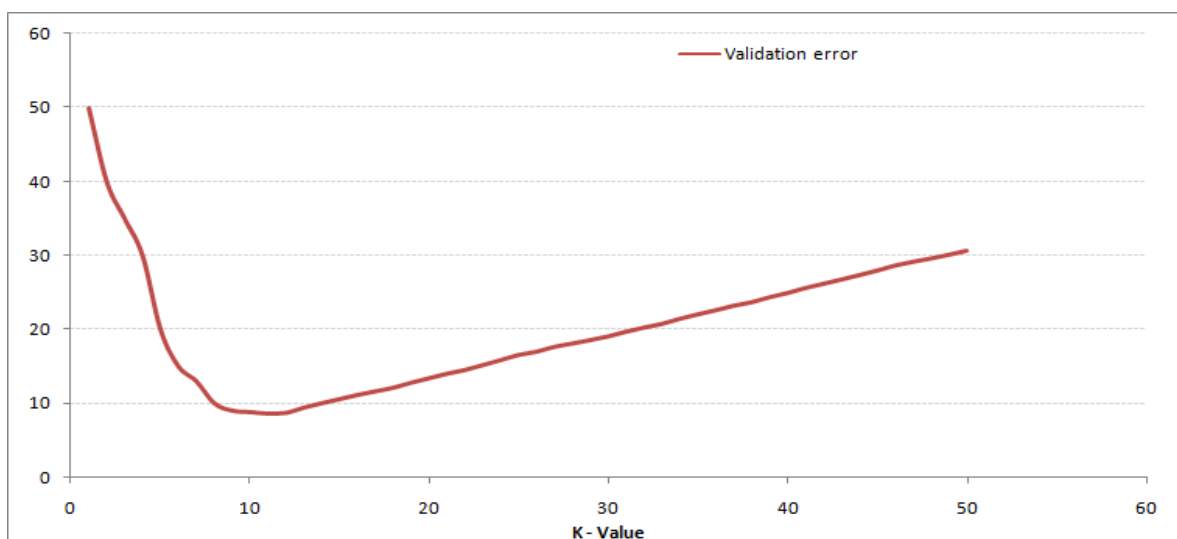


Fig 3.6: Validation error curve with varying value of K

This makes the story clearer. At $K=1$, we were overfitting the boundaries. Hence, error rate initially decreases and reaches a minimal. After the minima point, it then increases with increasing K . To get the optimal value of K , you can segregate the training and validation from the initial dataset. Now plot the validation error curve to get the optimal value of K . This value of K should be used for all predictions.

The process of KNN algorithm is measuring the similar values exist in close proximity as if we decrease the value of K from k to 1, the prediction may become less stable, the higher k value it becomes more stable and high prediction

In KNN, classification done based on the value of k , as it calculate the distance between datapoints

$$d(qp) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

3.10 MATTHEWS CORRELATION COEFFICIENT (MCC)

The Matthews Correlation Coefficient is used in machine learning as a measure of the quality of binary (two-class) classifications. It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC is in essence a correlation coefficient between the observed and predicted binary classifications; it returns a value between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 no better than random prediction and -1 indicates disagreement between prediction and observation.

The formula for the Matthews Correlation Coefficient is:

$$\text{"MCC"} = \frac{(TP \cdot TN - FP \cdot FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

where:

- MCC is the Matthews Correlation Coefficient
- TP = true positives
- FP = false positives
- TN = true negatives
- FN = false negatives

CHAPTER 4

DESIGN AND IMPLEMENTATION

In this system, we applied four machine learning algorithms which are Logistic Regression, K-Nearest Neighbors, Decision Tree and XGBoost. We implemented this model to find the best model among them for the datasets

4.1 DATA PREPROCESSING

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing. Data preprocessing is used database-driven applications such as customer relationship management and rule-based applications (like neural networks).

MinMaxScaler, Normalizer are method in scikit-learn are preprocessing methods, based on our features values we select the method, as we know machine learning algorithm will perform better and faster when features are relative or similar scale, we suggest MinMaxScale() for preprocessing, as it subtracts the minimum value in feature and divide with its range, difference of maximum and minimum is range MinMaxScale return the default range 0 to1.

4.2 FEATURE SELECTION

As we know, features play important role in classification, there are different approach in feature selection and based on the threshold value and benchmark algorithm we determine the optimality of feature in the dataset. Correlation coefficient features selection is the most widely used parameter, because feature selection is based on their correlation factor among the features, Let suppose f1 and f2 are two correlated features then to find Pearson's correlation coefficient (ρ)

$$\rho_{xy} = \frac{\text{cov}(x,y)}{\sigma_x \sigma_y}$$

Where, $\text{cov}(x,y)$ are covariance of variable x and y

$$R = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

Table 4.1: List of Speech dataset

Features	Description
Jitter(%) Jitter(Abs) MDVP:RAP MDVP:PPQ Jitter:DDP	Measure of variation in fundamental frequency
Shimmer Shimmer(db) Shimmer:APQ3 Shimmer:APQ5 MDVP:APQ	Measure of variation in fundamental frequency
NHR HNR	Two measures of ratio of noise to tonal components in the voice
MDVP:F0(Hz)	Average vocal fundamental frequency
MDVP:Fhi(Hz)	Maximum vocal fundamental frequency
MDVP:Flo(Hz)	Minimum vocal fundamental frequency
DFA	Signal fractal scaling exponent-

The correlation between features, it differentiates the strong positive and negative correlation among features, with that we selected few important features, MDVP:Jitter(%), Jitter(Abs), MDVP:RAP, MDVP:PPQ, Jitter:DPP, MDVP:Shimmer, MDVP:Shimmer(db), Shimmer APQ3, Shimmer APQ5, MDVP:APQ, Shimmer:DDA, NHR are the selected features from the dataset which is shown in table1 for the classification and the classification is made on this selected features for better accuracy

4.3 HEAT MAP

A heat map is a two-dimensional representation of data in which values are represented by colors. A simple heat map provides an immediate visual summary of

information. More elaborate heat maps allow the viewer to understand complex data sets.

There can be many ways to display heat maps, but they all share one thing in common -- they use color to communicate relationships between data values that would be much harder to understand if presented numerically in a spreadsheet.

4.4 LINE GRAPH

A line graph (also known as a line plot or line chart) is a graph which uses lines to connect individual data points that display quantitative values over a specified time interval. Line graphs use data point "markers" that are connected by straight lines to aid in visualization. Used across many fields, this type of graph can be quite helpful in depicting the changes in values over time.

4.5 BAR CHART

A bar graph is a chart that plots data using rectangular bars or columns (called bins) that represent the total amount of observations in the data for that category. Bar charts can be displayed with vertical columns, horizontal bars, comparative bars (multiple bars to show a comparison between values), or stacked bars (bars contain multiple types of information).

4.6 ARCHITECTURE

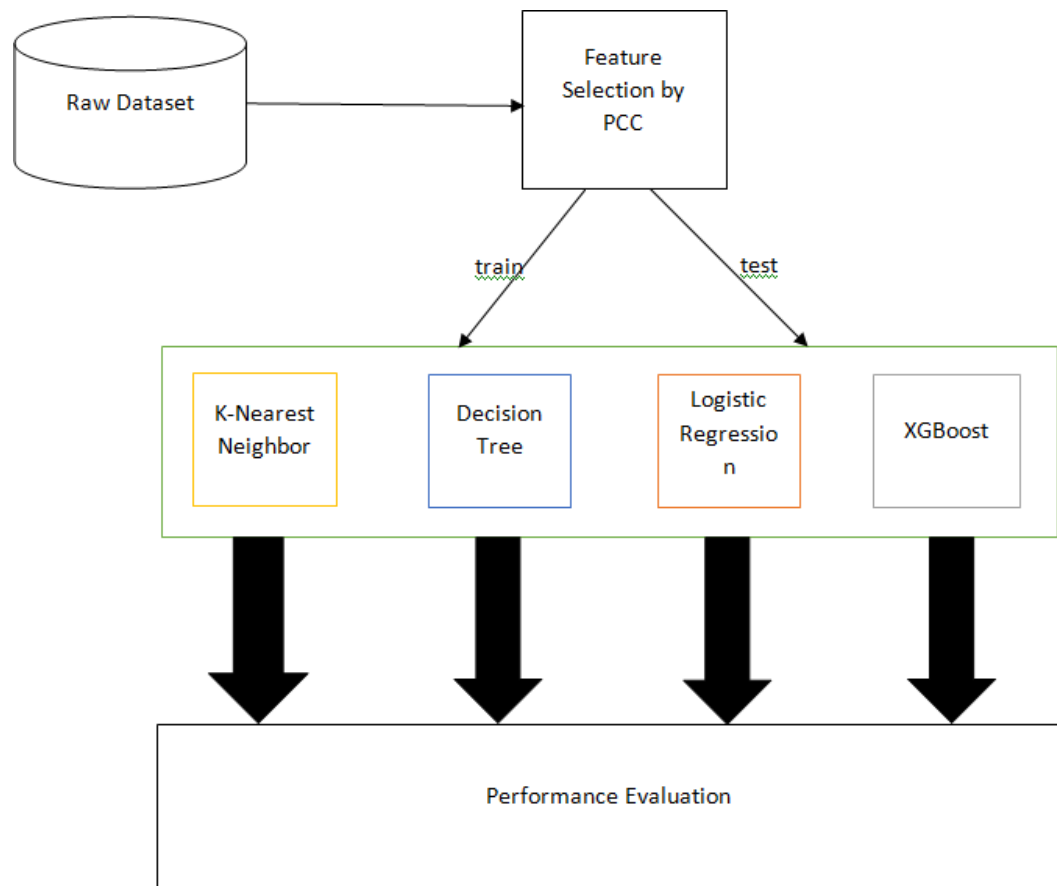


Fig 4.1: Architecture diagram of the process involved in this model

4.6.1 ARCHITECTURE EXPLANATION

The architecture describes the complete process involved in this model. At first, select the important features from the voice feature dataset which are very useful for detection and splitting the selected features dataset train and test for the classification. Here classification perform on four machine learning algorithms to determine which algorithm provides best result in detection, after training and testing the model, test the model with new dataset to predict the high accuracy and after evaluating all the performance of all four algorithms showing best algorithm for the PD dataset for detecting Parkinson's disease.

4.7 EVALUATION METRICS

Evaluating a machine learning algorithm is an essential part of any project. A model may give satisfying results when evaluated using a metric say accuracy score but may give poor results when evaluated against other metrics such as logarithmic loss or any other such metric. Most of the times we use classification accuracy to measure the performance of our model, however it is not enough to truly judge our model.

The success of classifier measured based on Accuracy, Sensitivity, Specificity and MCC. Formula for accuracy is ratio of correctly classified instance to whole instance

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

Where TP, TN, FP, FN are true positive, true negative, false positive, false negative respectively and Sensitivity and Specificity are:

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

$$\text{Specificity} = \frac{TN}{TN+FP}$$

MCC (Matthews Correlation Coefficient) shows the quality of binary classification. Correlation Coefficient between the predicted and binary classification is MCC

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

CHAPTER 5

RESULTS AND DISCUSSION

The performance analysis of the Parkinson's disease datasets has a variety of features. There are 21 features of all in which only 12 features play an important role for predicting the best algorithm in the classification.

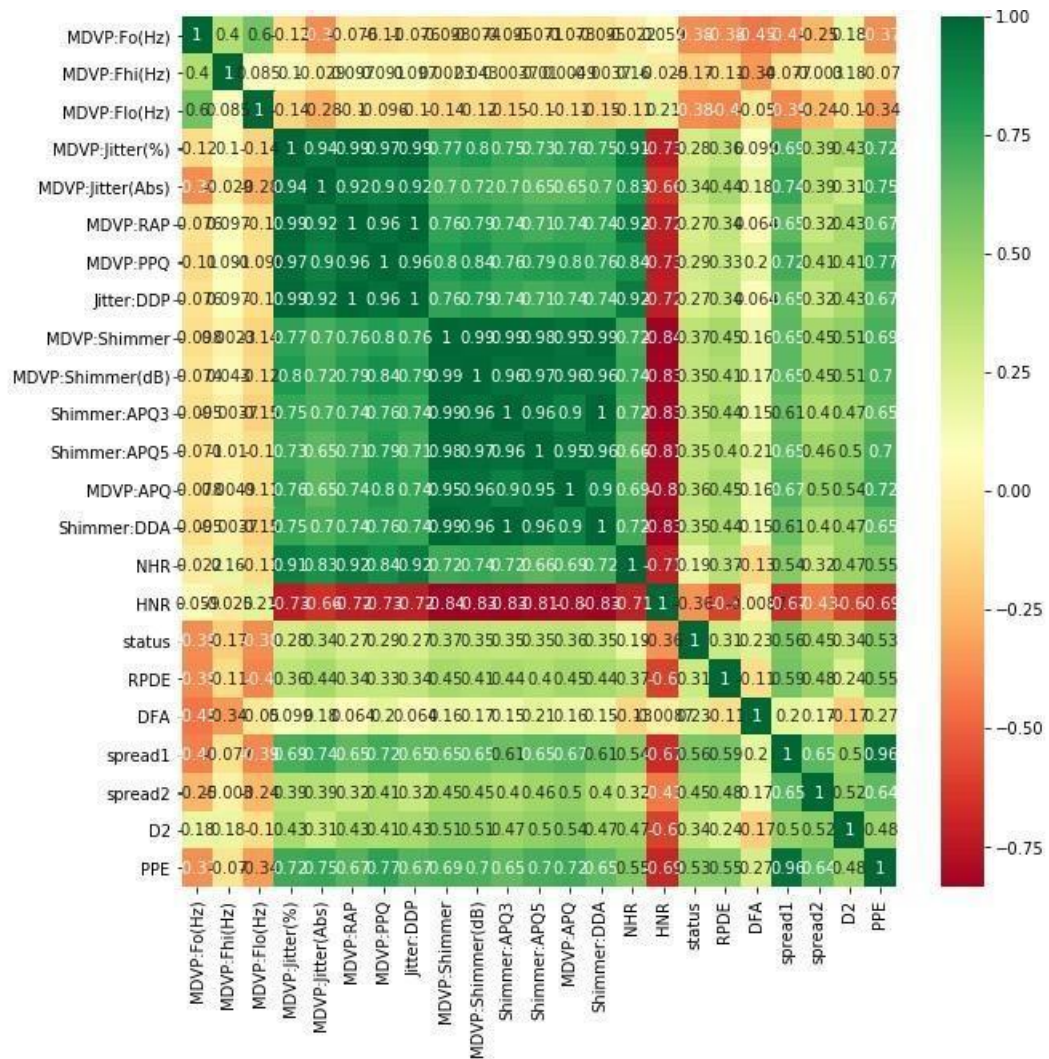


Fig 5.1: Relationship between data values on heat map

The colors in the graph is to communicate relationships between data values that would be much harder to understand if presented numerically in a spreadsheet. The green squares are the important features which we use them for the classification. The red squares are the negative values that are considered less in the algorithms.

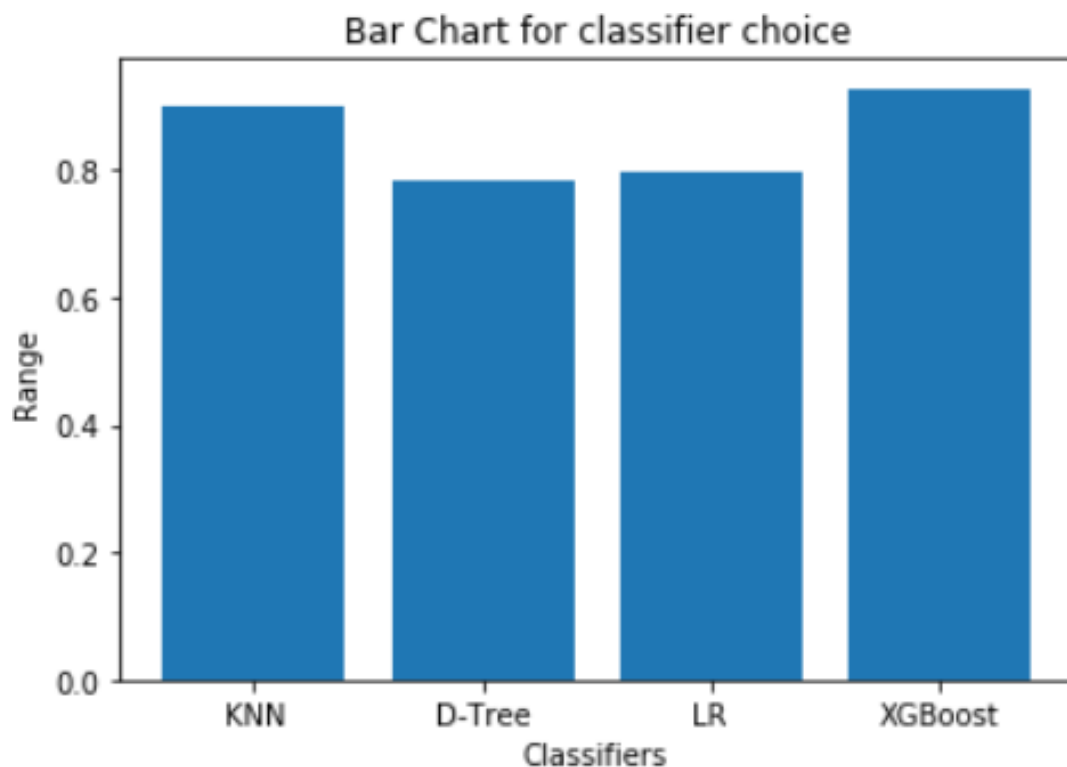


Fig 5.2: Bar chart for the classifier choice

Above bar graph represents the classification of each classifier which tells that decision tree has less accuracy which equals to 0.78 preceding that the next lowest accuracy is for logistic regression equals to 0.79. The highest accuracy has XGBoost of 0.93 and of the preceding K-nearest neighbor has 0.89 of accuracy. Matthews correlation coefficient also varies from to each algorithm in which logistic regression is the less MCC with 0.42, Decision tree has 0.60 MCC, K-nearest neighbor has 0.73

MCC where XGBoost has the highest MCC of 0.78. The tabular representation of the MCC and accuracy values are presented below:

Table 5.1: Tabular representation of Accuracy and MCC values of the various algorithms

S.No	Algorithm	Accuracy	MCC
0	K Nearest Neighbor	0.898305	0.733639
1	Decision Tree	0.785714	0.603023
2	Logistic Regression	0.796610	0.425447
3	XGBoost	0.928571	0.781736

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

A genetic algorithm is the most common method used to extract voice signal characteristics. For analysis, the common features considered are jittering, shimmering and voice signal frequency. Various algorithms for machine learning have been used in the prediction of Parkinson's disease. The range of prediction accuracy from applied algorithms lies between 75% and 95%, which demonstrates that machine learning algorithms can be used to predict Parkinson's disease by analyzing voice signals. Here the classification accuracy was studied and compared, with good performance and fast implementation, XGBoost achieved a high accuracy with 96%. This system provides the comparison between machine learning classifiers of Logistic regression, Decision tree, K-Nearest neighbor and XGBoost in PD disease diagnosis with high dimensional data MDVP:Jitter(%), Jitter(Abs), MDVP:RAP, MDVP:PPQ, Jitter:DPP, MDVP: Shimmer, MDVP:Shimmer(db), Shimmer APQ3, Shimmer APQ5, MDVP:APQ, Shimmer:DDA, NHR are the selected features from the dataset to achieve high accuracy in prediction.

6.2 FUTURE WORK

In future, we review the models of parkinson's disease detection and system of the proposed healthcare framework and design a model or system to diagnosis the parkinson's disease by patient audio files and increase the accuracy rate of prediction and at the same time to assist doctors and patient in diagnosis with an usefull information studies.

REFERENCES

- [1] A. Tsanas, M. A. Little, P. E. McSharry, J. Spielman and L. O. Ramig, "Novel Speech Signal Processing Algorithms for HighAccuracy Classification of Parkinson's Disease," in IEEE Transactions on Biomedical Engineering, vol. 59, no. 5, pp. 1264- 1271, May 2012.
- [2] Abós, A. et al. Discriminating cognitive status in Parkinson's disease through functional connectomics and machine learning. Scientific reports 7, 45347 (2017).
- [3] B. Zhang, J. Ren, Y. Cheng, B. Wang and Z. Wei, "Health Data Driven on Continuous Blood Pressure Prediction Based on Gradient Boosting Decision Tree Algorithm," in IEEE Access, vol. 7, pp. 32423-32433, 2019.
- [4] C. B. Levine, K. R. Fahrbach, A. D. Siderowf, R. P. Estok, V. M. Ludensky, and S. D. Ross, "Diagnosis and treatment of Parkinson's disease: a systematic review of the literature," Evid. Rep. Technol. Assess. (Summ.), no. 57, pp. 1–4, May 2003.
- [5] Center for Machine Learning and Intelligent System, Website:<http://Archive.lcs.UCI.Edu/ML/Datasets/Parkinsons>,2009.
- [6] Dobson, A. J. & Barnett, A. An introduction to generalized linear models. (CRC press, 2008).
- [7] Dr. Geetha R Ramani, G Sivagami,"Parkinson Disease Classification Using Data Mining Algorithms", International Journal Of Computer Applications 32(9):17-22, October 2011.
- [8] Gladence, L. Mary, M. Karthi, and V. Maria Anu. "A statistical comparison of logistic regression and different Bayes classification methods for machine learning." ARPN Journal of Engineering and Applied Sciences 10, no. 14 (2015): 5947-5953.
- [9] H. Ramezani and O. B. Akan, "Rate region analysis of multi-terminal neuronal nanoscale communication channel," in 17th IEEE NANO Conf. IEEE, 2017.
- [10] J. Jankovic, "Parkinson's disease: clinical features and diagnosis" Journal of Neurology, Neurosurgery & Psychiatry, pp.368-376, 2008.

- [11] K. Srilatha and V. Ulagamuthalvi "A Comparative Study on Tumor Classification "Research J. Pharm. and Tech 2019; 12(1): 407-411.
- [12] Mercy Paul Selvan, Nagubadi. Navadurga, Nimmagadda. Lakshmi Prasanna, "An Efficient Model for Predicting Student Dropout using Data Mining and Machine Learning Techniques", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue- 9S2, July 2019.
- [13] Mohammad S Islam ; Imtiaz Parvez ; Hai Deng ; Parijat Goswami," Performance Comparison Of Heterogeneous Classifiers For Detection Of Parkinson's Disease Using Voice Disorder (Dysphonia)", International Conference On Informatics, Electronics & Vision (Iciev), Page(S):1 – 7.
- [14] Mohammad shahbakhti, Danial Taherifar, Linear and Non-Linear speech features for detection of Parkinson's Disease, BMEiCON-2013.
- [15] R. Arefi Shirvan, E. Tahami, Voice analysis for detecting Parkinson's disease using genetic algorithm and KNN classification method, Proc 18th Int Con on Biomedical Engineering, Tehran, pp. 550-555, 2011.
- [16] Rätsch, G., Onoda, T. & Müller, K.-R. Soft margins for AdaBoost. Machine learning 42, 287–320 (2001).
- [17] Resuldas, "A Comparison of Multiple Classification Methods For Diagnosis Of Parkinson Disease", Expert Systems With Applications, Volume 37, Issue 2, March 2010, Pages 1568-1572.
- [18] Shraddha Khonde and V. Ulagamuthalvi(2019), Fusion of feature selection and Random Forest for an Anomaly based intrusion detection system , Journal of Computational and Theoretical Nanoscience , Volume 16, PP 3603-3607.

APPENDIX

A. SOURCE CODE

Classification-K

```
NN      import
pandas  as  pd
import numpy as
np import os, sys
import matplotlib.pyplot as
plt import seaborn as sns
from sklearn import preprocessing
from sklearn.metrics import matthews_corrcoef
%matplotlib inline

data = pd.read_csv('P.data')
#data =
pd.read_csv('PU.data')

data.head()

features = data.loc[:, data.columns != 'status'].values[:,
1:] labels = data.loc[:, 'status'].values

corrmat = data.corr()
top_corr_features =
corrmat.index
plt.figure(figsize=(10,10))
#plot heat map
g=sns.heatmap(data[top_corr_features].corr(),annot=True,cmap="RdYlGn"
)

from sklearn.preprocessing import
MinMaxScaler scaler = MinMaxScaler((-1, 1))
```

```
X = scaler.fit_transform(features)
```

```
Y = labels
```

```
#X =
```

```
scaler.fit_transform(features)
```

```
#Y =
```

```
scaler.fit_transform(labels)
```

```
from sklearn.model_selection import
```

```
train_test_split from sklearn import metrics
```

```
from sklearn.metrics import matthews_corrcoef
```

```
X_train, X_test, y_train, y_test = train_test_split( X, Y, test_size=0.3,  
random_state=4)
```

```
from sklearn.neighbors import
```

```
KNeighborsClassifier k = 7
```

```
#Train Model and Predict
```

```
neigh = KNeighborsClassifier(n_neighbors =
```

```
k).fit(X_train,y_train) yhat = neigh.predict(X_test)
```

```
A1=metrics.accuracy_score(y_test,
```

```
yhat) B1=matthews_corrcoef(y_test,
```

```
yhat) print("KNN Accuracy: ", A1)
```

```
print("MCC: ",B1)
```

```
Ks = 10
```

```
mean_acc =
```

```
np.zeros((Ks-1)) std_acc
```

```
= np.zeros((Ks-1))
```

```
ConfusionMx = [];
```

```
for n in range(1,Ks):
```

#Train Model and Predict

```

neigh = KNeighborsClassifier(n_neighbors =
n).fit(X_train,y_train) yhat=neigh.predict(X_test)
mean_acc[n-1] = metrics.accuracy_score(y_test, yhat)

std_acc[n-1]=np.std(yhat==y_test)/np.sqrt(yhat.shape[0])
plt.plot(range(1,Ks),mean_acc,'g')
plt.fill_between(range(1,Ks),mean_acc - 1 * std_acc,mean_acc + 1 *
std_acc, alpha=0.10)
plt.legend(('Accuracy ', '+/-
3xstd')) plt.ylabel('Accuracy ')
plt.xlabel('Number of Nabors
(K)') plt.tight_layout()
plt.show()

```

```

from sklearn.metrics import
confusion_matrix cm =
confusion_matrix(y_test, yhat)

```

```
cm
```

Decision Tree

```

import pandas as
pd import numpy
as np import os,
sys
import matplotlib.pyplot as
plt import seaborn as sns

```

```

from sklearn import preprocessing
from sklearn.metrics import matthews_corrcoef
%matplotlib inline

```

```

data = pd.read_csv('P.data')
#data =
pd.read_csv('PU.data')

data.head()
features = data.loc[:, data.columns != 'status'].values[:,
1:] labels = data.loc[:, 'status'].values

corrmat = data.corr()
top_corr_features =
corrmat.index

plt.figure(figsize=(10,
10)) #plot heat map

g=sns.heatmap(data[top_corr_features].corr(),annot=True,cmap="RdYlGn")

from sklearn.preprocessing import

MinMaxScaler scaler = MinMaxScaler((-1, 1))

X = scaler.fit_transform(features)
Y = labels

#X =
scaler.fit_transform(features)
#Y =
scaler.fit_transform(labels)

from sklearn.model_selection import
train_test_split from sklearn import metrics

```

```
from sklearn.metrics import matthews_corrcoef
```

```

from sklearn.model_selection import train_test_split

X_trainset, X_testset, y_trainset, y_testset =
train_test_split(X, Y, test_size=0.14, random_state=7)

from sklearn.tree import DecisionTreeClassifier

DT = DecisionTreeClassifier(criterion="entropy", max_depth
= 7) DT.fit(X_trainset,y_trainset)
y_tree = DT.predict(X_testset)

A2=metrics.accuracy_score(y_testset,
y_tree) B2=matthews_corrcoef(y_testset,
y_tree)

print("Decision Tree Accuracy: ",
A2) print("MCC: ",B2)

from sklearn.metrics import
confusion_matrix cm =
confusion_matrix(y_testset, y_tree)
cm

```

Logistic

Regression

```

import pandas as
pd import numpy
as np import os,
sys
import matplotlib.pyplot as
plt import seaborn as sns

from sklearn import preprocessing

```

```
from sklearn.metrics import matthews_corrcoef  
%matplotlib inline
```

```
data = pd.read_csv('P.data')
```

```
#data =
```

```
pd.read_csv('PU.data')
```

```
data.head()
```

```
features = data.loc[:, data.columns != 'status'].values[:,
```

```
1:] labels = data.loc[:, 'status'].values
```

```
corrmat = data.corr()
```

```
top_corr_features =
```

```
corrmat.index
```

```
plt.figure(figsize=(10,
```

```
10)) #plot heat map
```

```
g=sns.heatmap(data[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```

```
from sklearn.preprocessing import
```

```
MinMaxScaler scaler = MinMaxScaler((-1, 1))
```

```
X = scaler.fit_transform(features)
```

```
Y = labels
```

```
#X =
```

```
scaler.fit_transform(features)
```

```
#Y =
```

```
scaler.fit_transform(labels)
```

```
from sklearn.model_selection import
```

```
train_test_split from sklearn import metrics
```

```
from sklearn.metrics import matthews_corrcoef
from sklearn.model_selection import
train_test_split
```

```
from sklearn import metrics
from sklearn.metrics import matthews_corrcoef

from sklearn.model_selection import

train_test_split

X_trainset, X_testset, y_trainset, y_testset =
train_test_split(X, Y, test_size=0.14, random_state=7)
from sklearn.linear_model import
LogisticRegression from sklearn.metrics import
confusion_matrix

LR = LogisticRegression(C=0.098,
solver='liblinear').fit(X_train,y_train) yhat = LR.predict(X_test)

A3=metrics.accuracy_score(y_test,
yhat) B3=matthews_corrcoef(y_test,
yhat)

print("Logistic's Accuracy: ", A3)
```

XGBoost

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

```
from sklearn import preprocessing
from sklearn.metrics import matthews_corrcoef
```

```
%matplotlib inline
```

```
data = pd.read_csv('P.data')
```

```
#data =
```

```
pd.read_csv('PU.data')
```

```
data.head()
```

```
features = data.loc[:, data.columns != 'status'].values[:,
```

```
1:] labels = data.loc[:, 'status'].values
```

```
corrmat = data.corr()
```

```
top_corr_features =
```

```
corrmat.index
```

```
plt.figure(figsize=(10,
```

```
10)) #plot heat map
```

```
g=sns.heatmap(data[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```

```
from sklearn.preprocessing import
```

```
MinMaxScaler scaler = MinMaxScaler((-1, 1))
```

```
X = scaler.fit_transform(features)
```

```
Y = labels
```

```
#X =
```

```
scaler.fit_transform(features)
```

```
#Y =
```

```
scaler.fit_transform(labels)
```

```
from sklearn.model_selection import
```

```
train_test_split from sklearn import metrics
```

```
from sklearn.metrics import matthews_corrcoef
```

```

from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.14,
random_state=7)

model =
XGBClassifier()
model.fit(X_train,
Y_train)

XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
               colsample_bynode=1, colsample_bytree=1, gamma=0,
               learning_rate=0.1, max_delta_step=0, max_depth=3,
               min_child_weight=1, missing=None, n_estimators=100,
               n_jobs=1, nthread=None, objective='binary:logistic',
               random_state=0, reg_alpha=0, reg_lambda=1,
               scale_pos_weight=1, seed=None, silent=None, subsample=1,
               verbosity=1)

Y_hat = [round(yhat) for yhat in model.predict(X_test)]

A4=accuracy_score(Y_test, Y_hat)
B4=matthews_corrcoef(Y_test,
Y_hat)

print("accuracy ",accuracy_score(Y_test,

Y_hat)) print("MCC ",B4)

df2=pd.DataFrame({'Algorithm':['KNN','Decision Tree','Logistic
Regression','XGBoost'],'Accuracy':[A1,A2,A3,A4],'MCC':[B1,B2,B3,B4]})

```

df2

```
x = ['KNN', 'Decision_Tree',  
'Logistic_Reg','XGBoost'] y = [A1,A2,A3,A4]
```

```
# create bar chart
```

```
# show line plot
```

```
plt.show()
```

```
plt.xlabel('Classifie  
rs')
```

```
plt.ylabel('Ran
```

```
ge') plt.bar(x,  
y)
```

```
plt.title('Bar Chart for classifier choice')
```