

PARKINSON'S DISEASE CLASSIFICATION USING MACHINE LEARNING TECHNIQUES

A PROJECT REPORT

Submitted By:

GIRESH VENKATA TRINADH KANDRU (20BCS6536)

AYUSH SHARMA (20BCS6550)

MUDIT SHARMA (20BCS6542)

RISHABH RAI (20BCS6757)

POOJA JAIN (20BCS6551)

in the partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE ENGINEERING

Under the supervision of:

Ms. Merry K.P



DEC 2022

DECLARATION

We, *Rishabh Rai, Pooja Jain, Mudit Sharma, Giresk Venkata Trinadh , Ayush Sharma* of final semester B.Tech., in the department of Computer Science and Engineering from Chandigarh University, hereby declare that the project work entitled “***PARKINSON’S DISEASE PREDICTION USING CLASSIFICATION***” is carried out by us and submitted in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science Engineering, under Chandigarh University (Apex institute of Technology) during the academic year 2020-2024.

ABSTRACT

Parkinson's disease (PD) among Alzheimer's and epilepsy are one of the most common neurological disorders which appreciably affect not only live of patients but also their households. However, traditional diagnostic approaches may suffer from subjectivity as they rely on the evaluation of movements that are sometimes subtle to human eyes and therefore difficult to classify, leading to possible misclassification. In the meantime, early non-motor symptoms of PD may be mild and can be caused by many other conditions. The symptoms of PD are often overlooked, making diagnosis of PD at an early stage challenging. To address these difficulties and to refine the diagnosis and assessment procedures of PD, machine learning methods have been implemented for the classification of PD and healthy controls or patients with similar clinical presentations (e.g., movement disorders or other Parkinsonian syndromes). Even though there is no cure for PD, a proper medication at the early stage can help significantly in alleviating the symptoms. Since, the traditional method for identifying Parkinson disease is rather invasive, expansive and complicated for self-use, there is a high demand for using ML algorithms like classification method on PD detection. To solve these issues which we conducted a literature analysis of research papers. And most of the research papers were included in this study, with an examination of their targets, data sources and different types of datasets, ML algorithms, and associated outcomes. The results showed that ML approaches have a lot of promise for being used in clinical decision-making, resulting in a more systematic and informed way to predict Parkinson.

Key Words: Classification algorithm, Support Vector Machine, Parkinson's disease symptoms Traits.

CONTENTS

ABSTRACT

LIST OF SYMBOLS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CHAPTER 1: INTRODUCTION

CHAPTER 2: PROJECT OVERVIEW

2.1 GANTT CHART

2.2 MACHINE LEARNING

2.3 DATA PREPROCESSING

2.4 PYTHON

2.5 MOTIVATION OF THE WORK

2.6 PROBLEM STATEMENT

CHAPTER 3: LITERATURE SURVEY

CHAPTER 4: PROPOSED METHODOLOGY

CHAPTER 5: EXPERIMENTAL ANALYSIS AND RESULTS

5.1 MODULE DIVISION

5.1.1 DATA COLLECTION

5.1.2 ATTRIBUTE SELECTION

5.1.3 PRE-PROCESSING OF DATA

5.1.4 PREDICTION CLASSIFICATION

5.2 ALGORITHM

5.2.1 LOGISTIC REGRESSION

5.2.2 DECISION TREE

5.2.3 SUPPORT VECTOR MACHINE

5.3 SAMPLE CODE

5.3.1 CODE SCREENSHOT

5.4 RESULT

5.4.1 OUTPUT IMAGE

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

6.2 FUTURE SCOPE

APPENDIX

REFERENCES

LIST OF SYMBOLS

SYMBOL	MEANING
Σ	Summation (Uppercase Sigma)
Φ	Phi
tanh	Hyperbolic tangent function
σ	Sigmoid Function (Lowercase Sigma)
Θ	Angle

LIST OF FIGURES

FIGURE NO.	TITLE
1	Project overview
2	Machine Learning
3	Splitting of data set
4	Data mining
5	System Architecture
6	Logistic Regression
7	Decision Tree
8	Support Vector Machine

LIST OF TABLES

TABLE	CONTENTS
Table 1	Accuracy on Test data
Table 2	Accuracy on Training data

LIST OF ABBREBVIATIONS

SHORT FORM	FULL FORM
PD	PARKINSON'S DISEASE
SVM	Support vector machine

CHAPTER 1: INTRODUCTION

Parkinson's disease is a progressive nervous system disorder that affects movement leading to shaking, stiffness, and difficulty with walking, balance, and coordination. Parkinson's symptoms usually begin gradually and get worse over time.

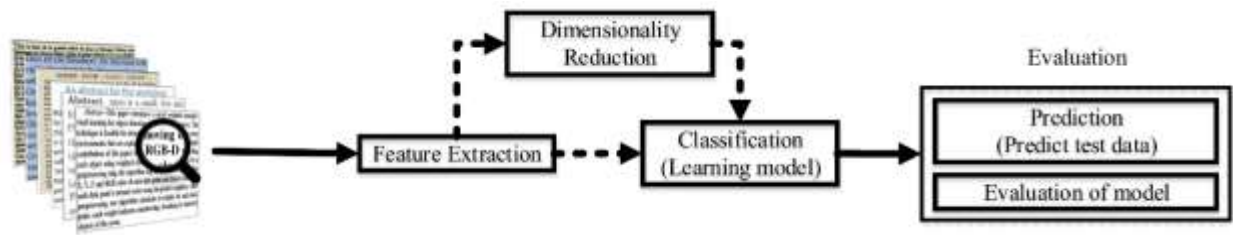
Parkinson's disease symptoms can be different for everyone. Early signs are mild that goes unnoticed. Symptoms usually begin on one side of your body and gets worsen on that side, afterwards it affects both the sides.

Parkinson's symptoms may include:

- Tremor
- Slowed movement
- Rigid muscles.
- Impaired posture and balance.
- Loss of automatic movements
- Speech changes
- Writing changes

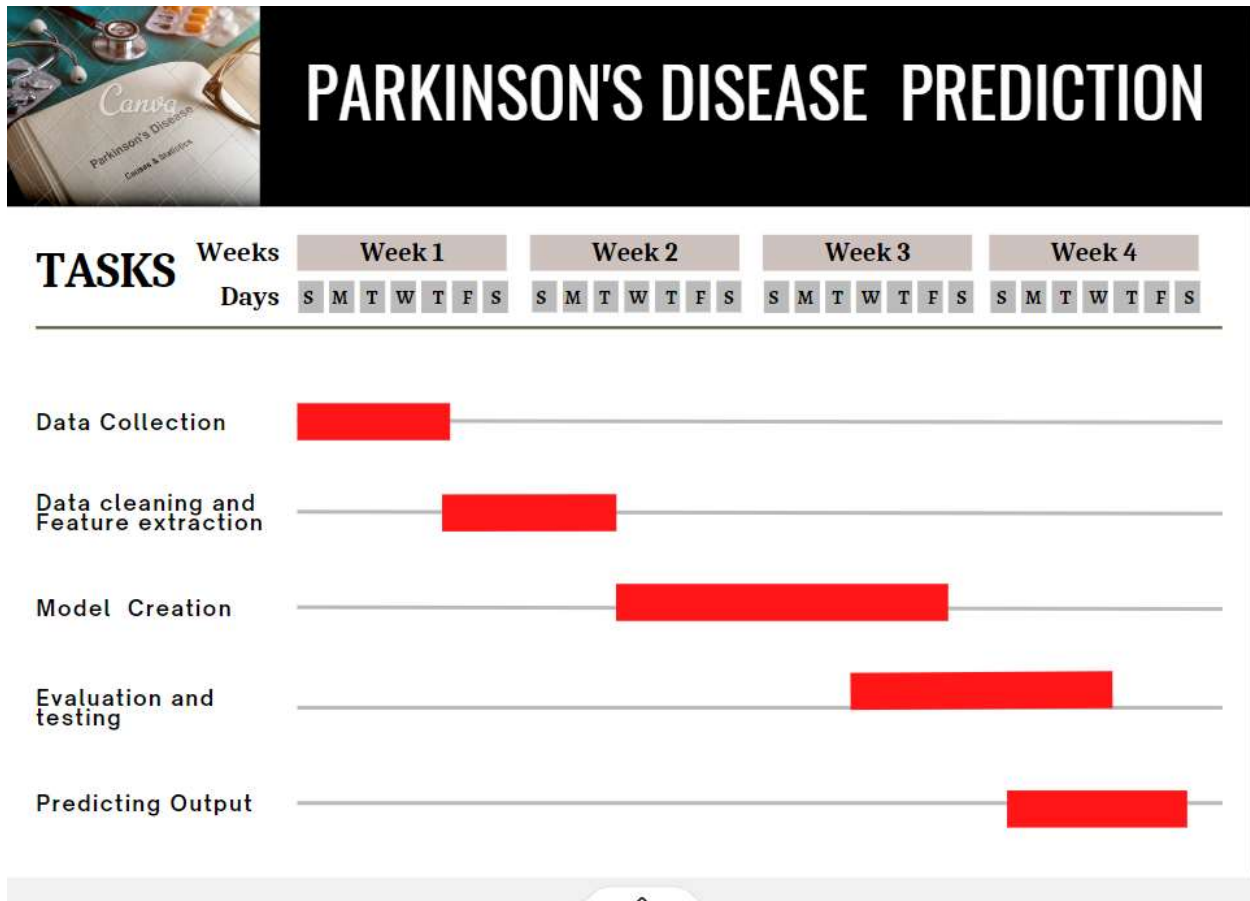
CHAPTER 2: PROJECT OVERVIEW

Our project is based on ML classification algorithm and through this project we will detect whether the person is suffering from Parkinson Disease or not for that, first we have to collect the dataset of the patients. So from Kaggle we will download Parkinson detection Datasets of PD affected and unaffected patients collected by neurologists are obtained from Machine Learning repository. These are stored into the python environment as Testing and Training datasets and imported using necessary packages. Python is an open source dynamic, high level, free and interpreted programming language. This supports object-oriented programming and procedural programming. Python is currently the most popular programming language for Machine Learning research and development. Jupyter notebook is an integrated development environment (IDE) primarily for the Python language, used in computer programming.



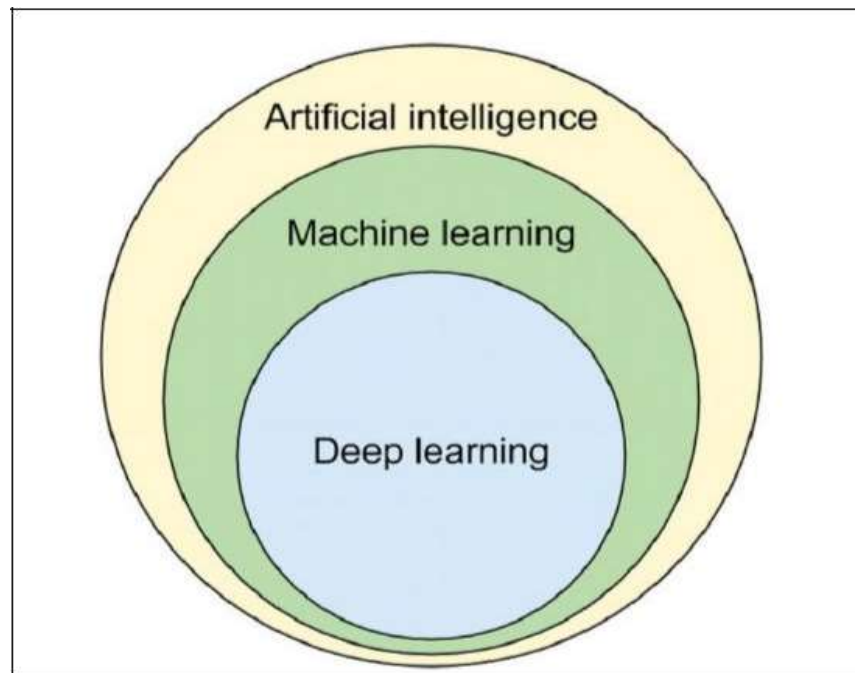
2.1. GANTT CHART-

A Gantt chart is a horizontal bar chart developed as a production control tool in 1917 by Henry L. Gantt, an American engineer and social scientist. It is a chart in which a series of horizontal lines shows the amount of work done or production completed in certain periods of time in relation to the amount planned for those periods.



2.2. MACHINE LEARNING

Machine learning (ML) is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a 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 in medicine, email filtering, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed task



In addition to an informed, working definition of machine learning (ML), we detail the challenges and limitations of getting machines to ‘think,’ some of the issues being tackled today in deep learning (the frontier of machine learning), and key takeaways for developing machine learning applications for business use-cases. Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

All of these things mean it's possible to quickly and automatically produce models that can analyze bigger, more complex data and deliver faster, more accurate results – even on a very large scale. And by building precise models, an organization has a better chance of identifying profitable opportunities – or avoiding unknown risks.

- **Supervised learning:**

In this type of machine learning, data scientists supply algorithms with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.

Examples of Supervised Learning: Regression, Decision Tree, Random Forest, KNN, Logistic Regression etc.

- **Unsupervised learning:**

This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.

Examples of Unsupervised Learning: Apriori algorithm, K-means.

- **Semi-supervised learning:**

This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labeled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.

- **Reinforcement learning:**

Data scientists typically use reinforcement learning to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task.

But for the most part, the algorithm decides on its own what steps to take along the way.

Example of Reinforcement Learning: Markov Decision Process.

2.3 DATA PREPROCESSING

There are seven significant steps in data pre-processing in Machine Learning:

1. Acquire the dataset

To build and develop Machine Learning models, you must first acquire the relevant dataset. This dataset will be comprised of data gathered from multiple and disparate sources which are then combined in a proper format to form a dataset. Dataset formats differ according to use cases

2. Import all the crucial libraries

Since Python is the most extensively used and also the most preferred library by Data Scientists around the world, we'll show you how to import Python libraries for data preprocessing in Machine Learning. Read more about [Python libraries for Data Science here](#). The predefined Python libraries can perform specific data preprocessing jobs. The three core Python libraries used for this data preprocessing in Machine Learning are:

- **NumPy**

NumPy is the fundamental package for scientific calculation in Python. Hence, it is used for inserting any type of mathematical operation in the code. Using NumPy, you can also add large multidimensional arrays and matrices in your code.

- **Pandas**

Pandas is an excellent open-source Python library for data manipulation and analysis. It is extensively used for importing and managing the datasets. It packs in high-performance, easy-to-use data structures and data analysis tools for Python.

- **Matplotlib**

Matplotlib is a Python 2D plotting library that is used to plot any type of charts in Python. It can deliver publication-quality figures in numerous hard copy formats and interactive environments across platforms (IPython shells, Jupyter notebook, web application servers, etc.).

3. Import the dataset

In this step, you need to import the dataset/s that you have gathered for the ML project at hand. However, before you can import the dataset/s, you must set the current directory as the working directory. Once you've set the working directory containing the relevant dataset, you can import the dataset using the "read_csv()" function of the Pandas library. This function can read a CSV file (either locally or through a URL) and also perform various operations on it. The read_csv() is written as:

```
dataset= pd.read_csv('Dataset.csv')
```

4. Identifying and handling the missing values

In data preprocessing, it is pivotal to identify and correctly handle the missing values, failing to do this, you might draw inaccurate and faulty conclusions and inferences from the data. Needless to say, this will hamper your ML project.

Basically, there are two ways to handle missing data:

- **Deleting a particular row** – In this method, you remove a specific row that has a null value for a feature or a particular column where more than 75% of the values are missing. However, this method is not 100% efficient, and it is recommended that you use it only when the dataset has adequate samples. You must ensure that after deleting the data, there remains no addition of bias.
- **Calculating the mean** – This method is useful for features having numeric data like age, salary, year, etc. Here, you can calculate the mean, median, or mode of a particular feature or column or row that contains a missing value and replace the result for the missing value. This method can add variance to the dataset, and any loss of data can be efficiently negated. Hence, it yields better results compared to the first method (omission of rows/columns). Another way of approximation is through the deviation of neighbouring values. However, this works best for linear data.

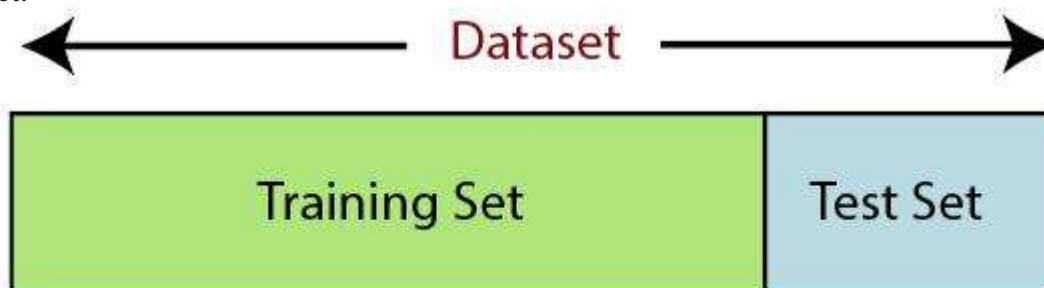
5. Encoding the categorical data

Categorical data refers to the information that has specific categories within the dataset. In the dataset cited above, there are two categorical variables – country and purchased.

Machine Learning models are primarily based on mathematical equations. Thus, you can intuitively understand that keeping the categorical data in the equation will cause certain issues since you would only need numbers in the equations.

6. Splitting the dataset

Every dataset for Machine Learning model must be split into two separate sets – training set and test set.



Training set denotes the subset of a dataset that is used for training the machine learning model. Here, you are already aware of the output. A test set, on the other hand, is the subset of the dataset that is used for testing the machine learning model. The ML model uses the test set to predict outcomes.

7. Feature scaling

Feature scaling marks the end of the *data preprocessing in Machine Learning*. It is a method to standardize the independent variables of a dataset within a specific range. In other words, feature scaling limits the range of variables so that you can compare them on common grounds.

2.4. PYTHON

- Python is a popular object-oriented programming language having the capabilities of high-level programming language. Its easy to learn syntax and portability capability makes it popular these days. The following facts given us the introduction to Python
- Python was developed by Guido van Rossum at Stichting Mathematisch Centrum in the Netherlands.
- It was written as the successor of programming language named 'ABC'.
- Its first version is released in 1991.
- The name Python was picked by Guido van Rossum from a TV show named Monty Python's Flying Circus.
- It is an open-source programming language which means that we can freely download it and use it to develop programs. It can be downloaded from www.python.org.
- Python programming language is having the features of Java and C both. It is having the elegant 'C' code and on the other hand, it is having classes and objects like java for object-oriented programming
- It is an interpreted language, which means the source code of Python program would be first converted into bytecode and then executed by Python virtual machine.

Why to learn "Python?"

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other language

Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning Python

Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP

Python is Interactive – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python is Object-Oriented – Python supports Object-Oriented style or technique of programming that encapsulates code within objects

Python is a Beginner's Language – Python is a great language for the beginner level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games

Characteristics of Python

Following are important characteristics of **Python Programming**

- It supports functional and structured programming methods as well as OOP.
- It provides very high-level dynamic data types, supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Applications of Python

As mentioned before, Python is one of the most widely used language over the web. I'm going to list few of them here:

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes
- **Easy-to-maintain** - Python's source code is fairly easy-to-maintain

A broad standard library – Python's bulk of the library is very portable and cross platform compatible on UNIX, Windows, and Macintosh

Interactive Mode – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code

Portable – Python can run on a wide variety of hardware platforms and has the same interface on all platforms

GUI Programming – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix

Scalable – Python provides a better structure and support for large programs than shell scripting

NUMPY

NumPy is one of the most powerful Python libraries. It is used in the industry for array computing. This article will outline the core features of the NumPy library. It will also provide an overview of the common mathematical functions in an easy-to-follow manner. Numpy is gaining popularity and is being used in a number of production systems.

PANDAS

pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

- Data Frame object for data manipulation with integrated indexing.
- Tools for reading and writing data between in-memory data structures and different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of data sets.
- Dataset merging and joining

SKLEARN

Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

Components of scikit-learn:

- Supervised learning algorithms
- Cross-validation
- Unsupervised learning algorithms
- Various toy datasets
- Feature extraction

MATPLOTLIB

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. Matplotlib is one of the most popular Python packages used for

data visualization. It is a cross-platform library for making 2D plots from data in arrays. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, Wx Python or Tkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.`

2.5. MOTIVATION OF THE WORK

In this project we tried to develop a system for Parkinson's prediction using Parkinson's dataset and their symptoms and also to evaluate the symptoms and their vocal frequency of the person. The symptoms of the human plays a major role in this and medical life.

Classification algorithms are used by many companies during the prediction process. They are designed to help patients for early prediction more insight into each candidate work style and performances. So this project will help the patients and medical industry to decide whether the person is affected from PD or not who are attending for this process. Persons can take this test and can check their health status by sitting at their home they don't need to go to anywhere.

As a result people suffer from this disease for many years before diagnosis. The estimated results have shown that there are 7-10 million people are affected by Parkinson's disease (PD) worldwide. People with age above 50 are the one who has the higher possibility of getting Parkinson's disease but still an estimated 4 percentage of people who are under the age 50 are diagnosed with Parkinson's disease. There is no cure or prevention for PD. However, the disease can be controlled in early stage. The Machine Learning Classification Algorithm techniques is going to use as a effective way for early detection and diagnosis of the disease. ML techniques in medicine is a research area that combines sophisticated representational and computing techniques with the insights of expert physicians to produce tools for improving healthcare. ML is a statistical method for finding hidden patterns in datasets by constructing predictive or classification models that can be learned from past experience and applied in future cases, so there is a need for a more accurate, objective means of early detection, ideally one which can be used by individuals in their home setting.

2.6. PROBLEM STATEMENT

The Parkinson's disease is due to a loss of neurons that produce a chemical messenger in the brain called dopamine. when there is a decrease in level of the amino acid named dopamine it leads to the abnormal brain activity, which leads to Parkinson's disease. The cause of Parkinson's disease is still a question mark, but several factors appear to play a role, including:

- Genes
- Environmental
- Triggers

As a result people suffer from this disease for many years before diagnosis. The estimated results have shown that there are 7-10 million people are affected by Parkinson's disease (PD) worldwide. People with age above 50 are the one who has the higher possibility of getting Parkinson's disease but still an estimated 4 percentage of people who are under the age 50 are diagnosed with Parkinson's disease. There is no cure or prevention for PD. However, the disease can be controlled in early stage. The Machine Learning Classification Algorithm techniques is going to use as a effective way for early detection and diagnosis of the disease. ML techniques in medicine is a research area that combines sophisticated representational and computing techniques with the insights of expert physicians to produce tools for improving healthcare. ML is a statistical method for finding hidden patterns in datasets by constructing predictive or classification models that can be learned from past experience and applied in future cases, so there is a need for a more accurate, objective means of early detection, ideally one which can be used by individuals in their home setting.

CHAPTER 3: LITERATURE SURVEY

This section describes the theoretical background of this project, starting with an explanation of Parkinson's disease, followed by overviews of machine learning, deep learning, related work and finally Parkinson's diagnosis (PD) problems. The detection of PD is extremely important at the first stage. The detection can be performed using ML technique.

Jie Mei et al . used all basic algorithms of Machine learning techniques for the detection of PD. Like SVM, RF, Decision Tree, ANN, KNN, Radial Basis Function Networks (RBF) and Deep Belief Networks (DBN) etc. The early identification of Parkinson's disease is critical. The identification can be performed with the use of a data mining technique. The techniques for detecting PD, such as Naive Bayes, support vector machine, multilayer perceptron neural network, and decision tree, are theoretically explained in this study. This study uses speech input from acoustic devices to predict Parkinson's disease. People from various areas and speech factors are investigated in this article in order to predict Parkinson's disease among patients.

Gabriel Solana-Lavalle et al. uses the algorithms such as Multilayer Perceptron (MLP), Random Forest (RF), K-Nearest Neighbor (KNN). For the prediction of Parkinson disease, three set of experiences were conducted to obtain the features with highest contribution to PD. This three sets are 1. a population with male and female subjects (balanced), 2 male subjects (balanced and unbalanced), and 3. Female subjects (balanced and unbalanced). In this study, the researchers used acoustic devices to collect speech parameters from 50 persons with Parkinson's disease and fifty healthy people. They employed the k-fold cross validation method for testing and claim that it can deliver good accuracy.

Kazi Amit Hasan et al . used different classification methods RF, KNN, Decision Tree, Logistic Regression (LR), SVM, and Naïve Bayes for detection of PD. The best result achieved by Decision Tree and Random Forest (RF) classification methods. The data mining techniques may be a more popular in many field of medical, business, railway, education etc. They are most commonly used for medical diagnosis and disease prediction at the early stage. The data mining is employed for healthcare sector in industrial societies.

Shail Raval et al. For the detection of PD they include all the aspects such as biological data, chemical data and genetic data. In this paper they mainly focused on the symptoms like rigidity, Tremor at rest, changing voice etc. The secure data transmission is proposed through authentication check, duplication check and faulty node detection. The proposed method is applicable to long ranges of transmission. It is also supporting a retransmission concept.

Mosarrat Rumman et al. based on Image Processing and Artificial Neural Network (ANN) classification algorithm According to ANN prediction, if value closer to 1 then suggests PD and value closer to 0 then suggest normal. Parkinson disease is a global public health issue. Machine learning technique would be a best solution to classify individuals and individuals with Parkinson's sickness (PD). This paper gives an entire review for the forecast of Parkinson disease by utilizing the machine learning based methodologies. A concise presentation of varied computational system based methodologies utilized for the forecast of Parkinson disease are

introduced. This paper likewise displays the outline of results acquired by different scientists from accessible information to predict the Parkinson disease.

Some studies applied supervised machine learning approaches for the classification of psychopathy. For example, Keshtkar et al. exploited different machine learning algorithms, namely, (i) SVM, (ii) NB, and (iii) DT, where the main emphasis of the proposed system was to automatically identify the personality traits of students in an educational game. The experimental results showed that the n-gram gave the best performance compared with the other feature sets.

Prediction of Parkinson disorder is one of the most important problem that has to be detected in 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 ND'S (Neurodegenerative disorders) using various machine learning techniques. Lots of pre-processing feature selection and classification techniques have been implemented and developed in the past decades.

CHAPTER 4: PROPOSED METHODOLOGY

Deep learning may be used to forecast how severe Parkinson's disease. Voice data from individuals with Parkinson's disease is first gathered for analysis. Signal error drop standardization is then used to normalize the data. An input layer, hidden layers, and output layer are created in the following phase of deep neural network architecture. A predetermined amount of input data characteristics determines how many neurons are in the input layer. Neurons in the output layer correlate to "severe" and "no severe" categories. The normalized data is sent into a deep neural network for training and testing. UCI Machine Learning Repository's Parkinson's Telemonitoring Voice Data Set was utilized. 42 patients' biological voice measures are included in the collection. Subject number, age, gender, time interval, Motor UPDRS, Total UPDRS, and 16 biomedical voice measurements are some of the data's properties. These patients' voices may be heard in 5,875 audio files in the collection. CSV files are used to store the information. Each patient provides an average of around 200 recordings (identification can be done through the first attribute-subject number)

Step 1. — (data pre-processing). As suggested by previous studies, the data is pre-processed to have a more accurate prediction of UPDRS. The goal of data pre-processing in this study is to handle the dataset's null values. In general, we included the pre-processing stage in the proposed method because it is typically completed during the first step of data analysis. The data is then deployed in the data analysis stages, such as SVM and prediction. The datasets are created with null values for method evaluation. Before clustering and classification tasks, these values must be imputed.

Step 2. — (dimensionality reduction). To remove the noise of data, the PCA method was used in this phase to lower the dimensionality of the data. Multicollinearity has a considerable impact on the accuracy of predictors and is a major issue in the field of disease diagnosis. The accuracy of SVR predictors has been affected by the multicollinearity of the data. We, therefore, use PCA to solve the multicollinearity problem as the most popular technique for noise removal.

Step 3. —(prediction). This stage was performed to predict according to the input features. In contrast to the previous prediction methods for PD diagnosis, we used ensembles of SVR to perform this task. SVR is trained to build prediction models with training datasets. It is a common practice to seek the advice of several doctors who are experts in the field in various clinical settings. The ultimate decision for a specific therapy is thus normally made through consultation and a combination of opinions of a committee of specialists. Ensemble learning systems serve a similar function in the machine learning context in general, ensemble learning systems can be utilized effectively in classification and regression problems and provide more reliable predictions than any individual learning model in fact, several weak hypotheses are combined in ensemble learning systems to form a stronger theory. Note that the success and effectiveness of ensemble learning approaches are heavily dependent on the diversity of the individual predictors that construct the ensemble. The total error can be reduced by combining the output of different prediction models through an algebraic expression (e.g., mean value of the predictions), as the various errors of the prediction models are averaged out.

The system architecture gives an overview of the working of the system. The working of the system starts with the collection of data from the data base and here we divide the data into training data and testing data, selecting the attributes. And then pre-processing the required data is done so that it removes duplicate data and the error data. Firstly, user have to login and then write the personality test there are 50 questions each trait consists of 10 questions, user have to answer those 50 questions, based up on the answers the algorithms are applied and the model is trained using the training data. Here we are big five personality traits and then classify the personality type. Accuracy is measured by testing the system using testing data. So, after that Personality is predicted.

CHAPTER 5: EXPERIMENTAL ANALYSIS AND RESULTS

5.1 MODULE DIVISION

Module Division is the process of dividing collection of source files required in the project into discrete units of functionality. Each module can be independently built, tested and debugged. Below are the modules which are divided in our project.

- 1.Data collection
- 2.Attribute selection
- 3.Pre-processing of data
- 4.Prediction of personality

5.1.1 Data Collection

First step for prediction system is data collection and deciding about the training and testing dataset. In this project we have imported dataset from Kaggle website which includes 70% of training dataset and 30% of testing dataset. Data collection is defined as the procedure of collecting, measuring and analyzing accurate insights for research using standard validated techniques. A researcher can evaluate their hypothesis on the basis of collected data. In most cases, data collection is the primary and most important step for research, irrespective of the field of research. The approach of data collection is different for different fields of study, depending on the required information.

TRAINING DATASET:

In a dataset, a training set is implemented to build up a model, while a test (or validation) set is to validate the model built. Here, you have the complete training dataset. You can extract features and train to fit a model and so on.

TESTING DATASET:

Here, once the model is obtained, you can predict using the model obtained on the training set. Some data may be used in a confirmatory way, typically to verify that a given set of input to a given function produces some expected result. Other data may be used in order to challenge the ability of the program to respond to unusual, extreme, exceptional, or unexpected input.

5.1.2 Attribute Selection

Attribute of dataset are property of dataset which are used for system and for personality many attributes are like heart gender of the person, age of the person ,Big five traits like Openness, Neuroticism, Extraversion, Agreeableness, Consciousness(value 1 -10). The importance of

feature selection can best be recognized when you are dealing with a dataset that contains a vast number of features. This type of dataset is often referred to as a *high dimensional* dataset. Now, with this high dimensionality, comes a lot of problems such as - this high dimensionality will significantly increase the training time of your machine learning model, it can make your model very complicated which in turn may lead to Overfitting.

5.1.3 Pre-Processing of Data

Pre-processing needed for achieving best result from the machine learning algorithms. In this, we gathered dataset and it was pre-processed before it is sent to training stage. Sampling is a very common method for selecting a subset of the dataset that we are analysing. In most cases, working with the complete dataset can turn out to be too expensive considering the memory. Using a sampling algorithm can help us reduce the size of the dataset to a point where we can use a better, but more *expensive*, machine learning algorithm. When we talk about data, we usually think of some large datasets with huge number of rows and columns. While that is a likely scenario, it is not always the case — data could be in so many different forms: Structured Tables, Images, Audio files, Videos etc. Machines don't understand free text, image or video data as it is, they understand 1s and 0s. So we pre-process the data.

5.1.4 Prediction of Parkinson's Classification

In this, system we used machine learning algorithms is performed and whichever algorithm is used which it gives best accuracy for personality prediction. By applying all this modules finally the personality is predicted and the final result is personality of the user. by using the training and testing dataset the personality of the user is classified.

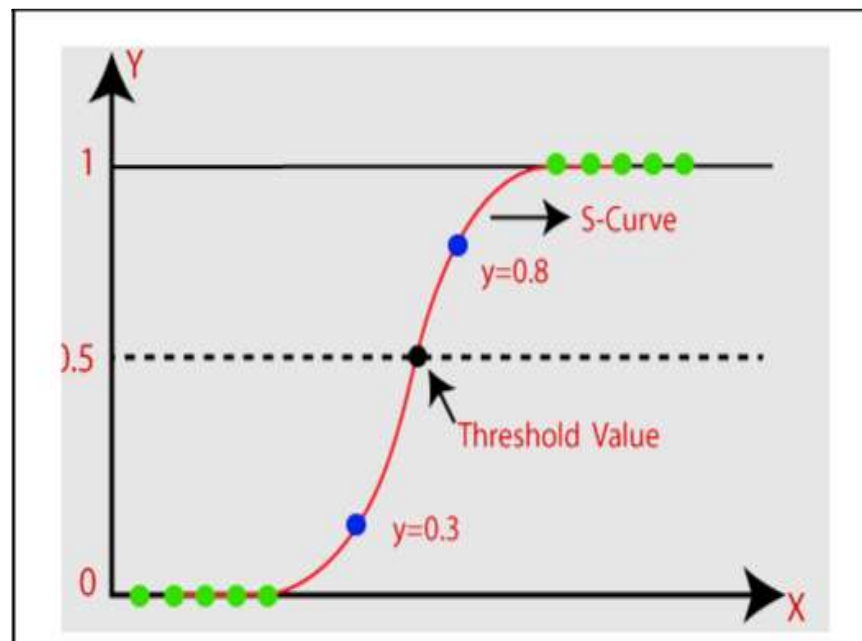
5.2 ALGORITHM

5.2.1 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.

- o Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- o Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

- o Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.
- o In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- o The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- o Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- o Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:



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 –

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.

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. 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 Assumptions

Before diving into the implementation of logistic regression, we must be aware of the following assumptions about the same –

- In case of binary logistic regression, the target variables must be binary always and the desired outcome is represented by the factor level 1.
- There should not be any multi-collinearity in the model, which means the independent variables must be independent of each other.
- We must include meaningful variables in our model.
- We should choose a large sample size for logistic regression.

Regression Models

- Binary Logistic Regression Model – The simplest form of logistic regression is binary or binomial logistic regression in which the target or dependent variable can have only 2 possible types either 1 or 0.
- Multinomial Logistic Regression Model – Another useful form of logistic regression is multinomial logistic regression in which the target or dependent variable can have 3 or more possible *unordered* types i.e., the types having no quantitative significance.

Logistic Regression Predicts Probabilities (Technical Interlude)

Logistic regression models the probability of the default class (e.g., the first class). For example, if we are modeling people’s sex as male or female from their height, then the first class could be male and the logistic regression model could be written as the probability of male given a person’s height, or more formally:

$$P(\text{sex}=\text{male}/\text{height})$$

Written another way, we are modeling the probability that an input (X) belongs to the default class (Y=1), we can write this formally as:

$$P(X) = P(Y=1/X)$$

We're predicting probabilities. I thought logistic regression was a classification algorithm.

Note that the probability prediction must be transformed into a binary values (0 or 1) in order to actually make a probability prediction. More on this later when we talk about making prediction. Logistic regression is a linear method, but the predictions are transformed using the logistic function. continuing on from above, the model can be stated as:

$$p(X) = e^{(b0 + b1 * X)} / (1 + e^{(b0 + b1 * X)})$$

I don't want to dive into the math too much, but we can turn around the above equation as follows (remember we can remove the e from one side by adding a natural logarithm (ln) to the other):

$$\ln(p(X) / 1 - p(X)) = b0 + b1 * X$$

This is useful because we can see that the calculation of the output on the right is linear again (just like linear regression), and the input on the left is a log of the probability of the default class.

This ratio on the left is called the odds of the default class (it's historical that we use odds, for example, odds are used in horse racing rather than probabilities). Odds are calculated as a ratio of the probability of the event divided by the probability of not the event, e.g. 0.8/(1-0.8) which has the odds of 4. So, we could instead write:

$$\ln(odds) = b0 + b1 * X$$

Because the odds are log transformed, we call this left hand side the log-odds or the probit. It is possible to use other types of functions for the transform (which is out of scope_, but as such it is common to refer to the transform that relates the linear regression equation to the probabilities as the link function, e.g. the probit link function.

We can move the exponent back to the right and write it as:

$$odds = e^{(b0 + b1 * X)}$$

All of this helps us understand that indeed the model is still a linear combination of the inputs, but that this linear combination relates to the log-odds of the default class.

5.2.2. DECISION TREE

Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split. which are outcomes like either 'fit', or 'unfit'. In this case this was a binary classification problem (a yes no type problem). There are two main types of Decision Trees:

1. Classification trees (Yes/No types)

What we've seen above is an example of classification tree, where the outcome was a variable like 'fit' or 'unfit'. Here the decision variable is **Categorical**.

2. Regression trees (Continuous data types)

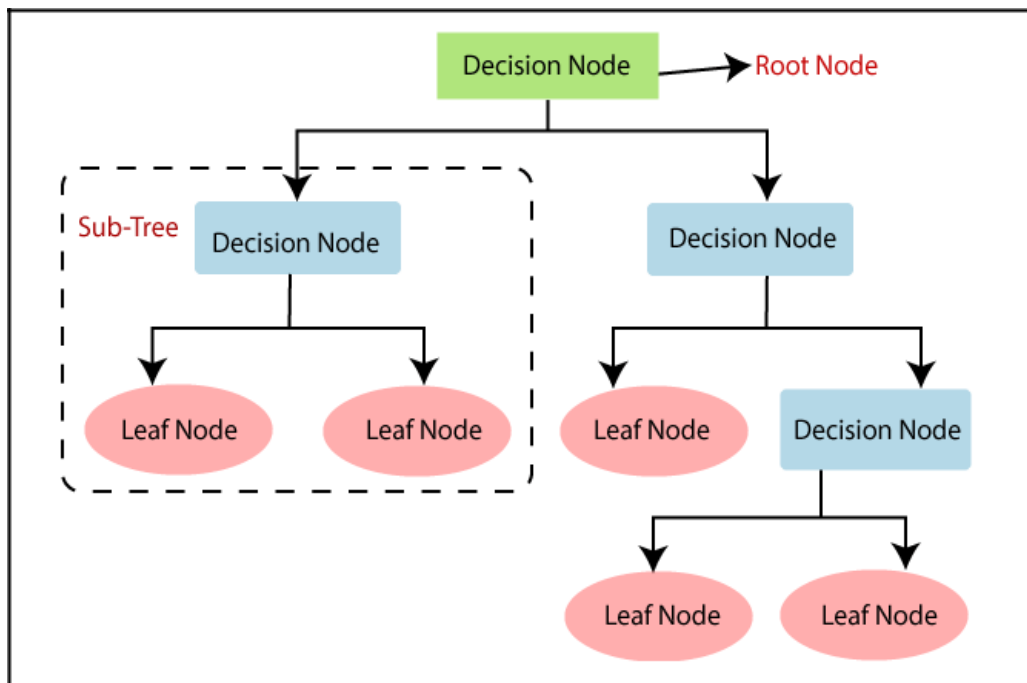
Here the decision or the outcome variable is **Continuous**, e.g. a number like 123 **Working** Now that we know what a Decision Tree is, we'll see how it works internally.

There are many algorithms out there which construct Decision Trees, but one of the best is called as **ID3 Algorithm**. ID3 Stands for **Iterative Dichotomiser 3**. Before discussing the ID3 algorithm, we'll go through few definitions. **Entropy** Entropy, also called as Shannon Entropy is denoted by $H(S)$ for a finite set S , is the measure of the amount of uncertainty or randomness in data.

We'll build a decision tree to do that using **ID3 algorithm**.

ID3 Algorithm will perform following tasks recursively

1. Create root node for the tree
2. If all examples are positive, return leaf node 'positive'
3. Else if all examples are negative, return leaf node 'negative'
4. Calculate the entropy of current state $H(S)$
5. For each attribute, calculate the entropy with respect to the attribute 'x' denoted by $H(S, x)$
6. Select the attribute which has maximum value of $IG(S, x)$
7. Remove the attribute that offers highest IG from the set of attributes
8. Repeat until we run out of all attributes, or the decision tree has all leaf nodes.



Decision Tree Terminologies

1.Root Node:

Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

2.Leaf Node:

Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

3.Splitting:

Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

4.Branch/Sub Tree:

A tree formed by splitting the tree.

5.Pruning:

Pruning is the process of removing the unwanted branches from the tree.

6. Parent/Child node:

The root node of the tree is called the parent node, and other nodes are called the child nodes.

How does the Decision Tree algorithm Work?

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

Step-1: Begin the tree with the root node, says S, which contains the complete dataset.

Step-2: Find the best attribute in the dataset using **Attribute Selection Measure (ASM)**.

Step-3: Divide the S into subsets that contains possible values for the best attributes.

Step-4: Generate the decision tree node, which contains the best attribute.

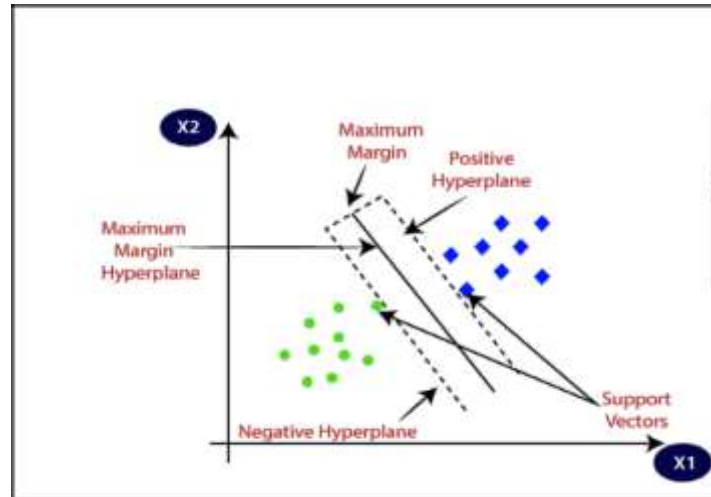
Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

5.2.3. SUPPORT VECTOR MACHINE

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



Types of SVM

SVM can be of two types:

- o **Linear SVM:**

Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

- o **Non-linear SVM:**

Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors in the SVM algorithm:

Hyperplane:

There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

Support Vectors:

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

This is a type of machine which is basically used for analysis the data which receive from supervised learning and identify the patterns for classification [8]. Training data set is taken and checked that whether the test data belongs to existing class or not for personality classification and classification.

Data is represented by Support Vector Machine model in the form of a point commonly in space which further classified in a line or in a hyper plane. The main idea behind the support vector machine algorithm is that if a classifier performs well at the most challenging comparisons, then it will definitely perform even better at the most easy comparisons. Support Vector Machine which is a nonlinear classifier often produces superior classification results than other classifier methods. Support Vector Machine is based on non-linearly mapping the input data to some high dimensional space where the data is separated linearly, thus giving accurate classification results.

The steps involved in Support Vector Machine are:

1. Create vectors for given question answers.
2. Then calculate the weights of the vectors.
3. Get the vectors with highest value and find value of personality
4. Finally predict personality type

PERFORMANCE METRICS

Evaluating machine learning algorithm is an essential part of a project. In this , We have used machine learning algorithms like Logistic Regression , Support Vector Machine , Decision Tree are used to predict the personality system. In this, we have 7 attributes and one attribute label as personality for predicting the behaviour from the test data. Various attributes like Name, Fi(Hz), FO(Hz), status, jitter, jitter abs etc. Conscientiousness are used in this system. Parkinson's Traits in our system and calculated performance metrics per each trait. We have Calculated precision, recall, f1- score, accuracy per each trait. For each classification algorithm we will be getting the performance metrics.

Confusion Matrix

It is the easiest way to measure the performance of a classification problem where the output can be of two or more type of classes. A confusion matrix is nothing but a table with two dimensions viz. “Actual” and “Predicted” and furthermore, both the dimensions have “True Positives (TP)”, “True Negatives (TN)”, “False Positives (FP)”, “False Negatives (FN)” as shown below –

		Actual	
		1	0
Predicted	1	True Positive(TP)	False Positive(FP)
	0	False Negative(FN)	True Negative(TN)

Explanation of the terms associated with confusion matrix are as follows –

True Positive: It is defined as model predicted as positive class and it is True. It is denoted by TP.

False Positive: It is defined as model predicted as positive class and it is false. It is denoted by FP.

False Negative: It is defined as model predicted as negative class and it is False. It is denoted by FN.

True Negative: It is defined as model predicted as negative class and it is True. It is denoted by TN.

Precision

It is the fraction of correctly classified instances to the total number of instances Precision, used in document retrievals, may be defined as the number of correct documents returned by our ML model. We can easily calculate it by confusion matrix with the help of following formula –

$$Precision = TP / (TP + FP)$$

Recall or Sensitivity

It is the fraction of correctly classified instances to the total number of instances Recall may be defined as the number of positives returned by our ML model. We can easily calculate it by confusion matrix with the help of following formula –

$$Recall = TP / (TP + FN)$$

Support

Support may be defined as the number of samples of the true response that lies in each class of target values.

F1 Score

This score will give us the harmonic mean of precision and recall. Mathematically, F1 score is the weighted average of the precision and recall. The best value of F1 would be 1 and worst would be 0. We can calculate F1 score with the help of following formula –

$$F1\ Score = (2 * Precision * Recall) / (Precision + Recall)$$

F1 score is having equal relative contribution of precision and recall.

Accuracy

Accuracy is the simple ratio between number of correctly classified points to the total number of points. It is most common performance metric for classification algorithms. It may be defined as the number of correct predictions made as a ratio of all predictions made. We can easily calculate it by confusion matrix with the help of following formula –

$$Accuracy = TP + TN / (TP + FP + FN + TN)$$

After performing the classification algorithms like Support Vector Machine, Decision Tree and Logistic Regression algorithms for training and testing data we find that Decision Tree is the Best algorithms with highest accuracy compared to both Support Vector Machine and Logistic Regression. We found the accuracy by calculating TP, FP, FN, TN is given and using the equation we got the accuracy. Among all the above three algorithms used, Accuracy comparisons are:

Support vector machine : .88%

Decision Tree : 85%

Logistic Regression : 78.%

5.3 SAMPLE CODE

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn import svm
from sklearn.metrics import accuracy_score
%matplotlib inline
import matplotlib.pyplot as plt
from scipy import stats
import seaborn as sns

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

parkinsons_data.head()

parkinsons_data.shape
    # number of rows and columns in the dataframe

parkinsons_data.info()
# getting more information about the dataset

parkinsons_data.isnull().sum()
# checking for missing values in each column

sns.heatmap(parkinsons_data.isnull())

correlation = parkinsons_data.corr()
print(correlation)
correlation

plt.figure(figsize=(20,20))
cor = parkinsons_data.corr()
sns.heatmap(cor,annot=True,cmap=plt.cm.CMRmap_r)
plt.show()
\

parkinsons_data.describe(include=['object'])

parkinsons_data.describe()
    # getting some statistical measures about the data

parkinsons_data['status'].value_counts()
# distribution of target Variable
```

```

parkinsons_data.groupby('status').mean()
    # Grouping the data based on the target variable

X = parkinsons_data.drop(columns=['name','status'], axis=1)
Y = parkinsons_data['status']
# Setting the independent and dependent variable

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
# Dividing the data into test and train

scaler = StandardScaler()
scaler.fit(X_train)
#Scaling the data in 0-1 range

X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)

model = svm.SVC(kernel='linear')
    # training the SVM model with training data

model.fit(X_train, Y_train)
# Fiting our data on train dataset

X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(Y_train, X_train_prediction)
# accuracy score on training data

X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(Y_test, X_test_prediction)
    # accuracy score on training data

input_data
(107.33200,113.84000,104.31500,0.00290,0.00003,0.00144,0.00182,0.00431,0.01567,0.13400,0
.00829,0.00946,0.01256,0.02487,0.00344,26.89200,0.637420,0.763262,-
6.167603,0.183721,2.064693,0.163755)

input_data_as_numpy_array = np.asarray(input_data)
    # changing input data to a numpy array

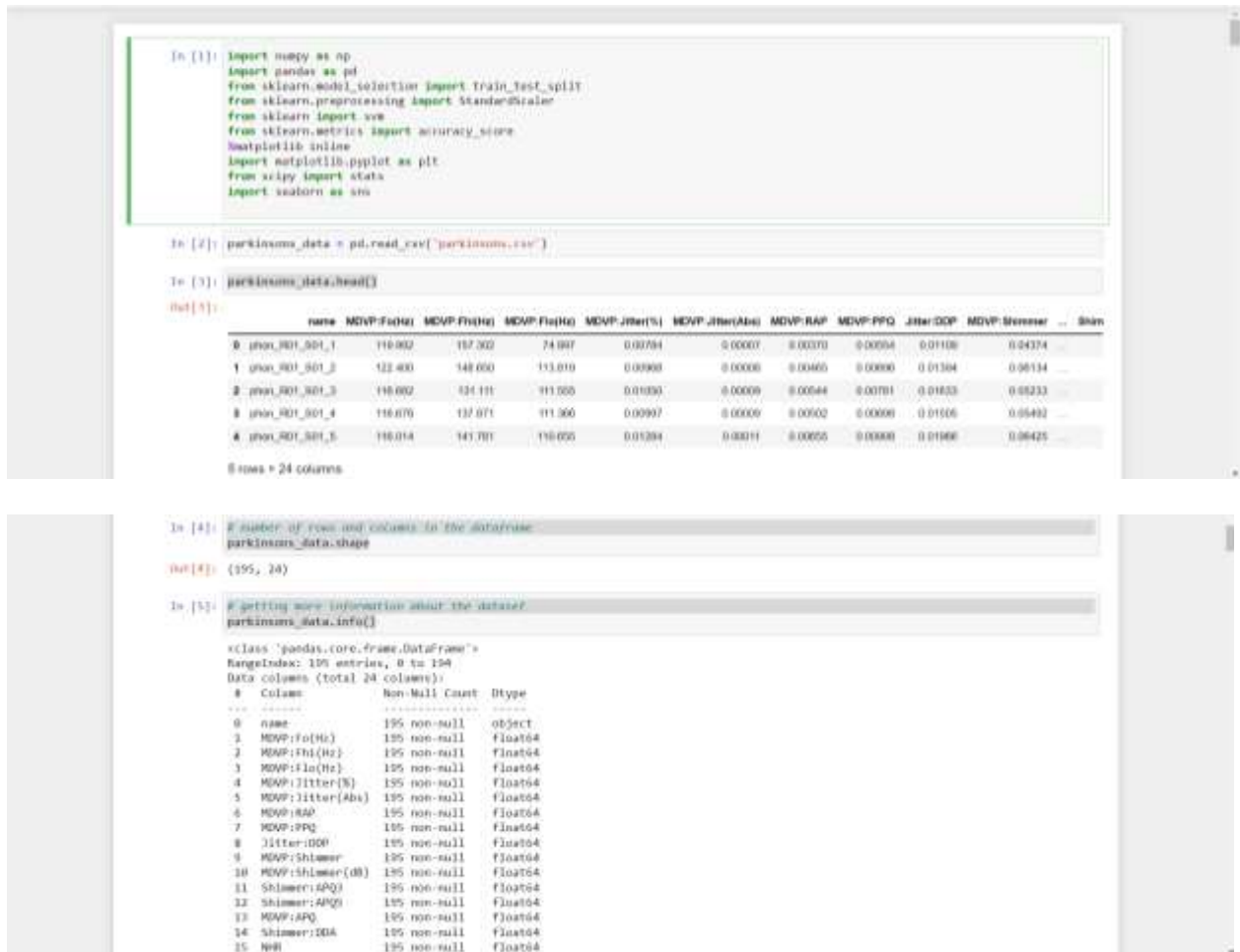
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
    # reshape the numpy array

std_data = scaler.transform(input_data_reshaped)
    # standardize the data

```

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

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



```
In [10]: correlation = parkinsons_data.corr()
print(correlation)
```

```

MDVP:F0(Hz)      MDVP:F1(Hz)      MDVP:F1a(Hz)      MDVP:Jitter(%)      \
MDVP:F0(Hz)      1.000000      0.400985      0.506546      -0.118001
MDVP:F1(Hz)      0.400985      1.000000      0.084951      0.102086
MDVP:F1a(Hz)     0.506546      0.084951      1.000000      -0.139919
MDVP:Jitter(%)   -0.118001      0.102086      -0.139919      1.000000
MDVP:Jitter(Abs) -0.382827      -0.629198      -0.277835      0.035734
MDVP:RAP         -0.076294      0.097377      -0.100519      0.090276
MDVP:PPQ         -0.112165      0.091126      -0.095828      0.074256
Jitter:DDP       -0.076293      0.097350      -0.100488      0.090276
MDVP:Shimmer     -0.096378      0.002281      -0.146543      0.769863
MDVP:Shimmer(dB) -0.075742      0.045465      -0.110889      0.004289
Shimmer:APQ5     -0.094717      -0.005743      -0.150747      0.746625
Shimmer:APQ6     -0.076682      -0.009997      -0.101095      0.725541
MDVP:APQ         -0.077774      -0.004937      -0.107293      0.758255
Shimmer:DBA      -0.094712      -0.003733      -0.130737      0.746635

```

```
In [12]: plt.figure(figsize=(20,20))
cor = parkinsons_data.corr()
sns.heatmap(cor, annot=True, cmap=plt.cm.CM_rmap_r)
plt.show()
```



```
In [15]: parkinsons_data.describe(include=['object'])
```

```

Out[15]:
      name
count    195
unique    195
top  phn1_f0f1_S18_8
freq      1

```

```
In [16]: # getting some statistical measures about the data
parkinsons_data.describe()
```

```

Out[16]:
      MDVP:F0(Hz)  MDVP:F1(Hz)  MDVP:F1a(Hz)  MDVP:Jitter(%)  MDVP:Jitter(Abs)  MDVP:RAP  MDVP:PPQ  Jitter:DDP  MDVP:Shimmer  MDVP:Shimmer(dB)
count    195.000000    195.000000    195.000000    195.000000    195.000000    195.000000    195.000000    195.000000    195.000000    195.000000
mean     154.229041    167.104010    196.324031         0.000220         0.000044         0.000300         0.003440         0.009920         0.029709         0.242251
std       41.300005     61.601548     43.521413         0.004040         0.000036         0.000200         0.002790         0.008805         0.018817         0.194877
min       90.113333     90.140000     90.470000         0.000000         0.000000         0.000000         0.000000         0.000000         0.000000         0.000000
max       240.113333    240.140000    240.470000         0.000000         0.000000         0.000000         0.000000         0.000000         0.000000         0.000000

```

```
In [20]: X_train = scaler.transform(X_train)
```

```
X_test = scaler.transform(X_test)
```

```
In [21]: print(X_train)
```

```

[[ 0.63239631 -0.02731881 -0.87985849 ... -0.97586547 -0.55160318
  0.07709494]
 [-1.05512719 -0.03337041 -0.5204778 ... 0.1981888 -0.61014073
  0.39291782]
 [ 0.02996187 -0.29531068 -1.12211187 ... -0.43937044 -0.62849605
  -0.50948489]
 ...
 [-0.9096795 -0.0637382 -0.160638 ... 1.22801922 -0.47488629
  -0.2159483 ]
 [-0.25977689 0.19731822 -0.79063679 ... -0.17896029 -0.47272835
  0.28181221]
 [ 1.01957056 0.19922317 -0.91914972 ... -0.710232  1.23052066
  -0.05629386]]

```

0 Model Training

Support Vector Machine Model

```
In [30]: # accuracy score on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(Y_train, X_train_prediction)
```

```
In [31]: print("Accuracy score of training data : ", training_data_accuracy)

Accuracy score of training data :  0.8846153846153846
```

```
In [32]: # accuracy score on testing data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(Y_test, X_test_prediction)
```

```
In [33]: print("Accuracy score of test data : ", test_data_accuracy)

Accuracy score of test data :  0.8717948717948718
```

Building a Predictive System

```
In [34]: input_data = (107.33300,111.84000,104.21500,0.00200,0.00003,0.00146,0.00202,0.00431,0.01507,0.13400,0.00625,0.00948,0.01256,0.0)

# changing input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the numpy array
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

# standardize the data
std_data = scaler.transform(input_data_reshaped)
prediction = model.predict(std_data)
print(prediction)

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

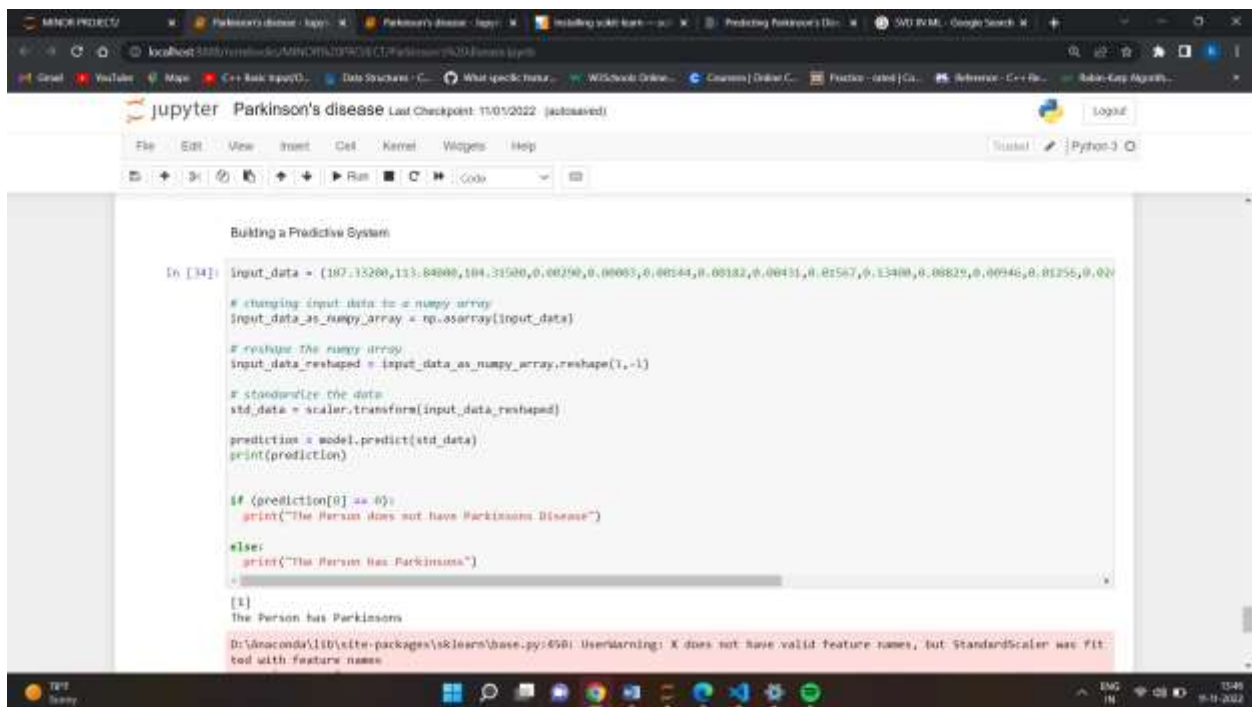
+ ~~~~~~

[1]
The Person has Parkinsons
```

5.5 RESULTS

ML has been used for medical disease detection lately and particularly Parkinson's disease (PD) treatment. This can be explained by the convenient performance and accurate results of ML techniques. Classification of diseases is a significant type of predictive modelling. It is considered an important data mining approach because it clusters the population referring to a predetermined criterion. It is vital to compare the outcomes of various classification methods to decide which approach presents the best performance. Hence, the main goal of this research is to assess several approaches that are utilized for PD prediction and classification. Even though ML methods have been assessed in several studies separately, the evaluation of these methods based on various datasets makes it complex to perform an accurate comparison among the deployed methodologies. Hence, it is vital to evaluate these methods in one comparative study based on a chosen dataset

5.5.1 OUTPUT IMAGES



```
Building a Predictive System

In [34]: input_data = [107.15200,133.84000,104.31500,0.00290,0.00003,0.00144,0.00182,0.00431,0.01567,0.13400,0.00829,0.00946,0.01256,0.02

# changing input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the numpy array
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

# standardize the data
std_data = scaler.transform(input_data_reshaped)

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

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

[1]
The Person has Parkinsons

D:\Anaconda\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but StandardScaler was fit-
ted with feature names
```


CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1. Conclusion

Parkinson's disease affects the CNS of the brain and has yet no treatment unless it's detect early. Late detection leads to no treatment and loss of life, so its early detection is significant. For early detection of the disease, we utilized machine learning algorithms such as Classification Random Forest. We checked our Parkinson disease data and find out Classification Random forest will be the best Algorithm to predict the disease which will enable early treatment and save a life.

6.2. Future Scope

In future work, we can focus on different techniques to predict the Parkinson disease using different datasets. In this research, we using binary attribute (1-diseased patients, 0-non-diseased patients) for patient's classification. In the future we will use different types of attributes for the classification of patients and also identify the different stages of Parkinson's disease.

APPENDIX

SAMPLE INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

INPUT- Answer the given questions

SAMPLE OUTPUT DESIGN

Inspecting the computer to read data from a Data base and answers given by the user or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

A quality output is one, which meets the requirements of the end user and presents the users personality clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making

OUTPUT-Early prediction whether the person is affected or not .

REFERENCES:

- [1] Adrien Payan, Giovanni Montana, Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks.
- [2] Alemami, Y. and Almazaydeh, L. (2014) Detecting of Parkinson Disease through Voice Signal Features. Journal of American Science,
- [3] Fayao Liu, Chunhua Shen, Learning Deep Convolutional Features for MRI Based Alzheimer's Disease Classification.
- [4] Hadjahamadi, A.H. and Askari, T.J. (2012) A Detection Support System for Parkinson's Disease Diagnosis Using Classification and Regression Tree. Journal of Mathematics and Computer Science , 4, 257-263.
- [5] Little, M.A., McSharry, P.E., Hunter, E.J. and Ramig, L.O. (2008), Suitability of Dysphonia Measurements for Telemonitoring of Parkinson's disease. IEEE Transactions on Biomedical Engineering, 56, 1015-1022.
- [6] Muhlenbach, F. and Rakotomalala, R. (2015) Discretization of Continuous Attributes. In: Wang, J., Ed., Encyclopedia of Data Warehousing and Mining, Idea Group Reference, 397-402.
- [7] Olanrewaju, R.F., Sahari, N.S., Musa, A.A. and Hakiem, N. (2014) Application of Neural Networks in Early Detection and Diagnosis of Parkinson's Disease. International Conference on Cyber and IT Service Management.
- [8] Saman Sarraf, Danielle D. De Souza, John Anderson, Ghassem To fight, Deep AD: Alzheimer's Disease Classification via Deep Convolutional Neural Networks using MRI and fMRI, Cold Spring Harbor Laboratory Press.