Internship Report

on

DETECTION OF PARKINSON'S DISEASE

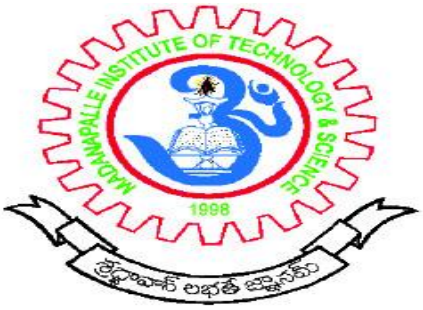
Submitted by
K.LAKSHITHA
20691A3217

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

DATA SCIENCE



MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC – AUTONOMOUS)

(Affiliated to JNTUA, Ananthapuramu)
Accredited by NBA, Approved by AICTE, New Delhi)
AN ISO 9001:2008 Certified Institution
P. B. No: 14, Angallu

2022--2023

✆ : 08571 - 280255, 280590
MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
An Engineering College Sponsored by :
RATAKONDA RANGA REDDY EDUCATIONAL ACADEMY
Approved by AICTE, New Delhi and Affiliated to JNTUA, Anantapur
P.B. No. 14, Angallu, MADANAPALLE - 517 325. Chittoor Dist. A.P.
www.mits.edu

DEPARTMENT OF DATA SCIENCE

BONAFIDE CERTIFICATE

This is to certify that the internship work entitled **" DETECTION OF PARKINSON'S DISEASE"** is a bonafide work carried out by

K.LAKSHITHA - 20691A3217

Submitted in partial fulfillment of the requirements for the award of degree **Bachelor of Technology** in the Department of **Data Science**, **Madanapalle Institute of Technology and Science, Madanapalle,** affiliated to **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** during the academic year 2022-2023

Mrs. Dhanalakshmi.R **Assistant Professor** **Department of AI & DS**	**Mr.Toralkar Pawan** **Assistant Professor** **Internship Coordinator** **Department of AI & DS**	**Dr. K. Chokkanathan** **Head & Associate Professor** **Department of AI & DS**

ACKNOWLEDGEMENT

I sincerely thank the **Management** of **Madanapalle Institute of Technology and Science** for providing excellent infrastructure and lab facilities that helped me to complete this project.

I sincerely thank **Dr. C. Yuvaraj , M.E., Ph.D**., **Principal** for guiding and providing facilities for the successful completion of our project at **Madanapalle Institute of Technology and Science,** Madanapalle.

I express our deep sense of gratitude to **Dr. K. Chokkanathan, M. Tech., Ph.D.,** Associate Professor & Head, Department of AI & DS for his continuous support in making necessary arrangements for the successful completion of the project.

I express our sincere thanks to the **Internship Coordinator**, **Mr Toralkar Pawan, M.E.,** Assistant Professor, Department of AI & DS for his tremendous support for the successful completion of the internship Project.

.

I express our deep gratitude to Mrs. Dhana Lakshmi .R, Designation, Department of AI & DS for her guidance and encouragement that helped us to complete this internship project.

I also wish to place on record my gratefulness to other **Faculty members of Department of AI & DS** and also to our friends and our parents for their help and cooperation during our internship.

CERTIFICATE

Remark
Skill
Education

SUMMER INTERNSHIP APPRECIATION LETTER

Date: 6th August, 2022

TO WHOM SO EVER IT MAY CONCERN

This is to certify that **Ms.** **Kolla lakshitha** has successfully completed **Summer Internship** with **Remark Skill Education** on **Data Science using Python.** During the span of internship with us, the candidate has been actively & diligently involved in the projects and tasks assigned to her. The candidate has exclusively worked over the project which implicated on the practical execution of the course ware and the skills the students had learned with us as part of Remark Skill Online Internship Program during **28th June 2022 to 6th August 2022.**

So hereby with this letter, we acknowledge that the student has done project on the topic of:
DETECTING PARKINSON'S DISEASE

We found her sincere, hardworking, technically sound and result oriented. She has shown an impressive skill towards being a diligent task handler as well as collaborating with the team and being a seamless part of the team during her tenure. She has shown the ability to incorporate the ideologies and concepts of the company and being a strong team player and contributing towards successful result oriented goals. We take this opportunity to thank her to being a part of our organisation and wish her all the best for her future from **Remark Skill Education**.

Madhu Das

Ms. Madhu Das
Mobile: +91 8287-916-037
Managing Head | Remark Skill Education
Office Address: WOLK, Bakshi House 40-41, Nehru Place-110019
Regd. Address: Plot No.1/3989 ground floor, Ram Nagar, Shahadara delhi -110032
Email: info@remarkskill.com **Visit:** www.remarkskill.com

Associated Certification Partners

Microsoft Technology Associate
CERTIFIED ASSOCIATE
AUTODESK. Certified User
EC-Council
Microsoft Office Specialist

DECLARATION

I, the undersigned hereby declare that the results embodied in this Internship **"Detection of Parkinson's Disease"** is a bonafide record of the work done by me in partial fulfillment of the award of **Bachelor of Technology** in **Data Science** from **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu. The content of this report is** not submitted to any other University/Institute for award of any other degree.

Place **: Madanapalle**

Date : 25-11-2022

Name : K.Lakshitha

Roll No : 20691A3217

Department of Data Science

Madanapalle Institute of Technology & Science

Madanapalle

ABSTRACT

In various data repositories, there are large medical datasets available that are used to identifying the diseases. Parkinson's is considered one of the deadliest and progressive nervous system diseases that affect movement. It is the second most common neurological disorder that causes disability, reduces the life span, and still has no cure. Nearly 90% of affected people with this disease have speech disorders.

To increase the lifespan of elderly people the machine learning algorithms are used to detect diseases in the early stages. Speech features are the main concept while taking into consideration the term 'Parkinson's'. In this project, I have used Machine Learning technique like SVM and how these algorithm are used to predict Parkinson's based on the input taken from the user and the input for algorithm is the dataset. Based on these features I have predicted the algorithm that gives more accuracy. To recover the patients from early stages, prediction is important. This process can be done with the help of Machine Learning.

LIST OF FIGURES

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PYTHON

Python is widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software. Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that integrate systems more efficiently. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages.

• **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 techniques of programming that encapsulates code within objects.

1.2 Python Features

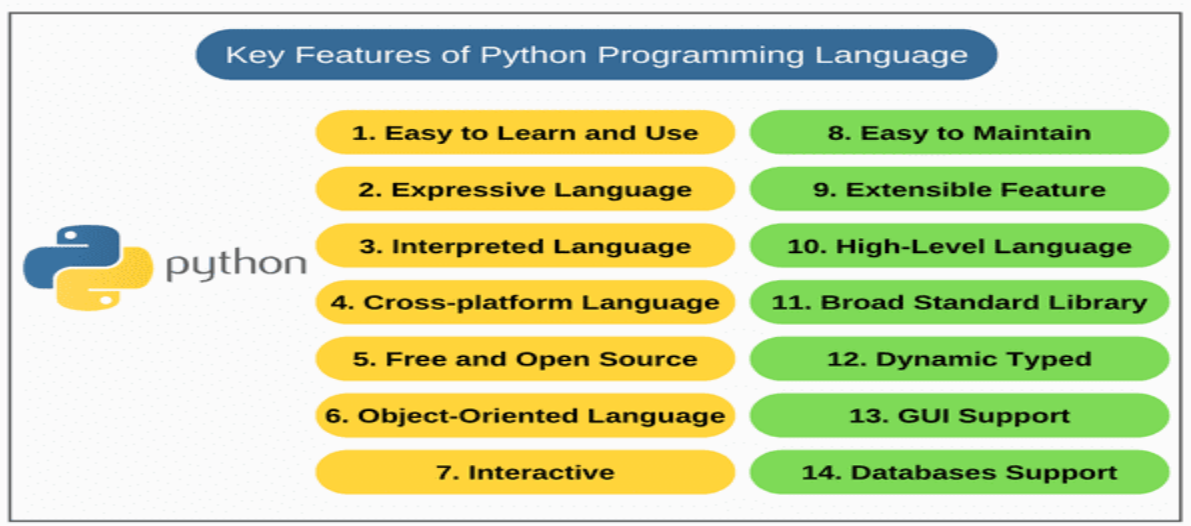


Fig 1.2 Features of Python

Python's features include –

• **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.

• **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

• **Databases** − Python provides interfaces to all major commercial databases.

• **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

1.3 Applications on Python

- Web Development
- Game Development
- Machine Learning and Artificial Intelligence
- Data Science and Data Visualization
- Desktop GUI
- Web Scraping Applications
- Business Applications
- Audio and Video Applications
- CAD Applications
- Embedded Applications

1.4 ABOUT MACHINE LEARNING

Machine learning is a sub-domain of computer science. It uses data and artificial intelligence in its area of applications. It is considered as the top-notch pass to the most interesting and growing careers in the current world. It is used to make predictions and gain insights. This can be achieved by providing the data to train the model. At a broad level, machine learning can be classified into three types:

- Supervised Machine Learning
- Unsupervised Machine Learning
- Reinforcement learning

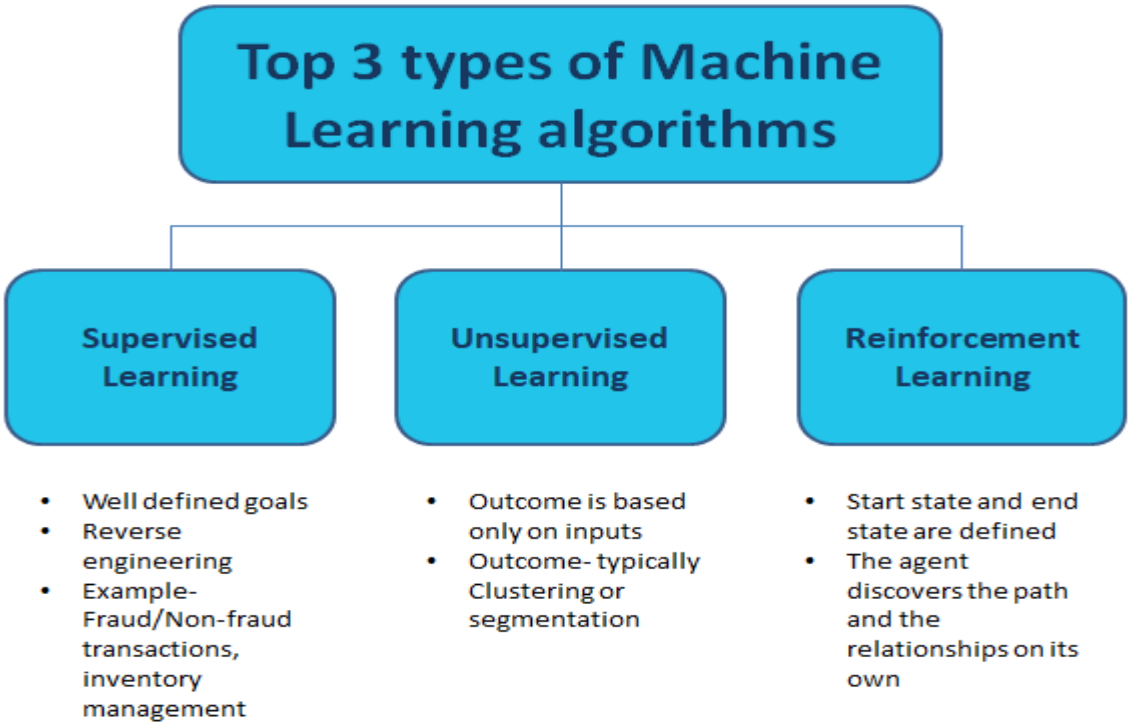


Fig 1.4 Types of Machine Learning

- **Supervised Machine Learning**

In supervised learning technique, we train the machines using the "labelled" dataset, and based on the training, the machine predicts the output. Here, the labelled data specifies that some of the inputs are already mapped to the output. More preciously, we can say; first, we train the machine with the input and corresponding output, and then we ask the machine to predict the output using the test dataset.

Let's understand supervised learning with an example. Suppose we have an input dataset of cats and dog images. So, first, we will provide the training to the machine to understand the images, such as the shape & size of the tail of cat and dog, Shape of eyes, color, height (dogs are taller, cats are smaller), etc. After completion of training, we input the picture of a cat and ask the machine to identify the object and predict the output. Now, the machine is well trained, so it will check all the features of the object, such as height, shape, color, eyes, ears, tail, etc., and find that it's a cat. So, it will put it in the Cat category.

This is the process of how the machine identifies the objects in Supervised Learning The main goal of the supervised learning technique is to map the input variable(x) with the output variable(y). Some real-world applications of supervised learning are Risk Assessment, Fraud Detection, Spam filtering, etc.

Categories of Supervised Machine Learning

Supervised machine learning can be classified into two types of problems, which are given below:

- Classification
- Regression

Classification

Classification algorithms are used to solve the classification problems in which the output variable is categorical, such as "Yes" or No, Male or Female, Red or Blue, etc**.** The classification algorithms predict the categories present in the dataset. Some real-world examples of classification algorithms are Spam Detection, Email filtering, etc.

Some popular classification algorithms are given below :

- Random Forest Algorithm
- Decision Tree Algorithm
- Logistic Regression Algorithm
- Support Vector Machine Learning Algorithm

Regression

Regression algorithms are used to solve regression problems in which there is a linear relationship between input and output variables. These are used to predict continuous output variables, such as market trends, weather prediction, etc.

Some popular Regression algorithms are given below

- Linear Regression Algorithm
- Non-Linear Regression Algorithm
- Polynomial Regression Algorithm
- Bayesian Regression Algorithm

Advantages and Disadvantages of Supervised Learning

Advantages

- Since supervised learning work with the labelled dataset so we can have an exact idea about the classes of objects.
- These algorithms are helpful in predicting the output on the basis of prior experience.

Disadvantages

- These algorithms are not able to solve complex tasks.
- It may predict the wrong output if the test data is different from the training data.
- It requires lots of computational time to train the algorithm.

Applications of Supervised Learning

Some common applications of Supervised Learning are given below:

- **Image Segmentation :** Supervised Learning algorithms are used in image segmentation. In this process, image classification is performed on different image data with pre-defined labels.
- **Medical Diagnosis :** Supervised algorithms are also used in the medical field for diagnosis purposes. It is done by using medical images and past labelled data with labels for disease conditions. With such a process, the machine can identify a disease for the new patients.
- **Fraud Detection :** Supervised Learning classification algorithms are used for identifying fraud transactions, fraud customers, etc. It is done by using historic data to identify the patterns that can lead to possible fraud.
- **Spam detection :** In spam detection & filtering, classification algorithms are used. These algorithms classify an email as spam or not spam. The spam emails are sent to the spam folder.

- **Speech Recognition :** Supervised learning algorithms are also used in speech recognition. The algorithm is trained with voice data, and various identifications can be done using the same, such as voice-activated passwords, voice commands, etc.

Un Supervised Machine Learning Algorithm

In unsupervised machine learning, the machine is trained using the unlabelled dataset, and the machine predicts the output without any supervision . In unsupervised learning, the models are trained with the data that is neither classified nor labelled, and the model acts on that data without any supervision.

Categories of Unsupervised Machine Learning

Unsupervised Learning can be further classified into two types, which are given below:

- Clustering
- Association

Clustering

The clustering technique is used when we want to find the inherent groups from the data. It is a way to group the objects into a cluster such that the objects with the most similarities remain in one group and have fewer or no similarities with the objects of other groups. An example of the clustering algorithm is grouping the customers by their purchasing behaviour.
Some of the popular clustering algorithms are given below:

- K-Means Clustering algorithm
- Mean-shift algorithm
- DBSCAN Algorithm
- Principal Component Analysis
- Independent Component Analysis

Association

Association rule learning is an unsupervised learning technique, which finds interesting relations among variables within a large dataset. The main aim of this learning algorithm is to find the dependency of one data item on another data item and map those variables accordingly

so that it can generate maximum profit. This algorithm is mainly applied in Market Basket analysis, Web usage mining, continuous production.

Some popular algorithms of Association rule learning are Apriori Algorithm, Eclat, FP-growth algorithm.

Advantages and Disadvantages of Unsupervised Learning Algorithm

Advantages

- These algorithms can be used for complicated tasks compared to the supervised ones because these algorithms work on the unlabeled dataset.
- Unsupervised algorithms are preferable for various tasks as getting the unlabeled dataset is easier as compared to the labelled dataset.

Disadvantages

The output of an unsupervised algorithm can be less accurate as the dataset is not labelled, and algorithms are not trained with the exact output in prior.

- Working with Unsupervised learning is more difficult as it works with the unlabeled dataset that does not map with the output.

Applications of Unsupervised Learning

- **Network Analysis:** Unsupervised learning is used for identifying plagiarism and copyright in document network analysis of text data for scholarly articles.
- **Recommendation Systems:** Recommendation systems widely use unsupervised learning techniques for building recommendation applications for different web applications and e-commerce websites.
- **Anomaly Detection:** Anomaly detection is a popular application of unsupervised learning, which can identify unusual data points within the dataset. It is used to discover fraudulent transactions.
- **Singular Value Decomposition:** Singular Value Decomposition or SVD is used to extract information from the database. For example, extracting information of each user located at a particular location.

Reinforcement Learning

Reinforcement learning works on a feedback-based process, in which an AI agent (A software component) automatically explore its surrounding by hitting & trail, taking action, learning from experiences, and improving its performance. Agent gets rewarded for each good action and get punished for each bad action; hence the goal of reinforcement learning agent is to maximize the rewards .In reinforcement learning, there is no labelled data like supervised learning, and agents learn from their experiences only.

The reinforcement learning process is similar to a human being; for example, a child learns various things by experiences in his day-to-day life. An example of reinforcement learning is to play a game, where the Game is the environment, moves of an agent at each step define states, and the goal of the agent is to get a high score. Agent receives feedback in terms of punishment and rewards. Due to its way of working, reinforcement learning is employed in different fields such as Game theory, Operation Research, Information theory, multi-agent systems.

Advantages and Disadvantages of Reinforcement Learning

Advantages

- It helps in solving complex real-world problems which are difficult to be solved by general techniques.
- The learning model of RL is similar to the learning of human beings; hence most accurate results can be found.
- Helps in achieving long term results.

Disadvantages

- RL algorithms are not preferred for simple problems.
- RL algorithms require huge data and computations.
- Too much reinforcement learning can lead to an overload of states which can weaken the results.

1.5 IMPORTANCE AND APPLICATIONS OF MACHINE LEARNING

With the growing economy, the world is changing, and the internet has become the data generation machine. Machine learning helps the data analysts to organize and handle this data. It helps in analyzing the data and provides valuable insights. Machine learning allows the software's to become more accurate.

It is known to everyone that large companies are describing the Machine Learning as "The future". It has many applications in various domains. Few of them are listed below :

1) Image Recognition
2) Automatic language Translation
3) Medical Diagnosis
4) Stock market Trading
5) Online Fraud Detection
6) Virtual Personal Assistant
7) Email Spam and Malware Filtering
8) Self-driving cars
9) Recommendation Systems (Movie recommendation, Music Recommendation)
10) Image recognition.

CHAPTER 2

TOOLS AND TECHNOLOGLY

2.1 SOFTWARE REQUIREMENTS

Here, I have used Colab platform for executing my code. Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited by your team members just the way you edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in your notebook.

As a programmer, you can perform the following using Google Colab.

- Write and execute code in Python
- Document your code that supports mathematical equations
- Create/Upload/Share notebooks
- Import/Save notebooks from/to Google Drive
- Import/Publish notebooks from GitHub
- Import external datasets e.g., from Kaggle
- Integrate PyTorch, TensorFlow, Keras, OpenCV
- Free Cloud service with free GPU

2.2 HARDWARE REQUIREMNTS

Device name : HP Laptop 15p-da0xxx
Processor : intel® Core ™ i5 -82504 CPU @ 1.60GHz 1.80
Installed Ram : 8.00 GB
System type : 64-bit Operating System , X64- based processor.

CHAPTER 3

PROJECT ANALYSIS

3.1 PROBLEM STATEMENT

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

The technique used in this problem is Support Vector Machine (SVM) .The experimental study is performed on the voice dataset of Parkinson's patients which is downloaded from the Kaggle. The prediction is evaluated using evaluation metrics like confusion matrix, precision, recall and accuracy. I have used feature selection where the important features are taken into consideration to detect Parkinson Disease

3.2 PROJECT OVERVIEW

In this project Detection of Parkinson's Disease is a progressive, neuro degenerative disorder that affects both motor and cognitive functions. Parkinson Disease is treated with Deep Brain Stimulation (DBS). With DBS ,electrodes are surgically implemented into the brain. They send electrical impulses into the brain that control movement. Several drugs are now available for the treatment of Parkinson's Disease that include levodopa , Mao-B inhibitors .Parkinson's Disease symptoms can be different for everyone Early signs are mild that gets unnoticed**.**

Symptoms usually begin on one side of our body and gets worse on another side .Parkinson disease affects the person above the 50 to 60 years of age **.** There isn't a specific test to diagnose Parkinson's disease. A doctor trained in nervous system conditions (neurologist) will diagnose. Parkinson's disease based on your medical history, a review of your signs and symptoms ,analyze neurological and physical examination. Parkinson's Disease has motor and non-motor symptoms.

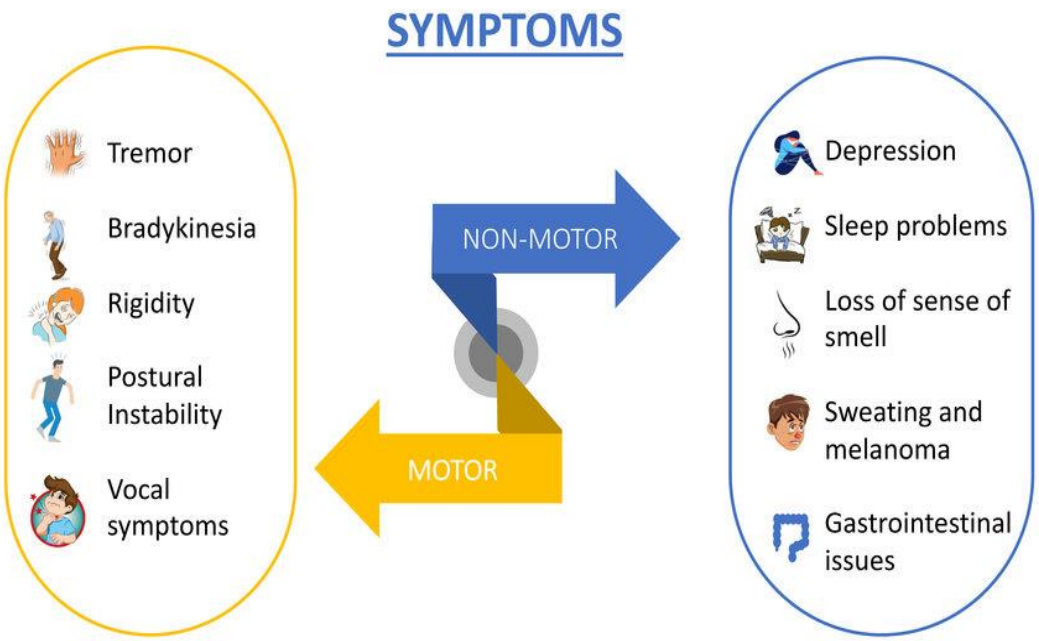


Fig 3.2 Motor and Non-motor Symptoms

Parkinson's disease can't be cured, but medications can help control the symptoms, often dramatically. In some more advanced cases, surgery may be advised. Your health care provider may also recommend lifestyle changes, especially ongoing aerobic exercise. In some cases, physical therapy that focuses on balance and stretching is important. A speech-language pathologist may help improve speech problems.

As well as the main symptoms of movement problems, people with Parkinson's disease can experience a wide range of additional symptoms that may need to be treated separately. Diagnosis of Parkinson's disease (PD) is commonly based on medical observations and assessment of clinical signs, including the characterization of a variety of motor symptoms. 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

3.3 ALGORITHM

I have used , Support Vector Machine (SVM) for detection of Parkinson’s Disease .The accuracy I got using this algorithm is 87%.SVMs use input-output pairs during training to construct an N-dimensional hyperplane that optimally separates the data into two categories .

After training, the SVM can be used to predict which category a new input vector belongs to. Soft margin SVMs were trained the data set . The performance of SVM operating on both two dimensional and three-dimensional input attributes vectors were compared.

SVM performs classification task by finding decision boundary called hyper-plane. This plan is the optimal hyper plane. The goal with the algorithm is to find the optimal level that can be controlled when the situation is exact. It is used to separate two different classes: PD and NPD. After that, test the data with this network and an output is obtained. Then obtained output and trained output is compared and choose the minimum distance and a class is obtained. That is whether a person has a Parkinson's disease or non-Parkinson's disease.

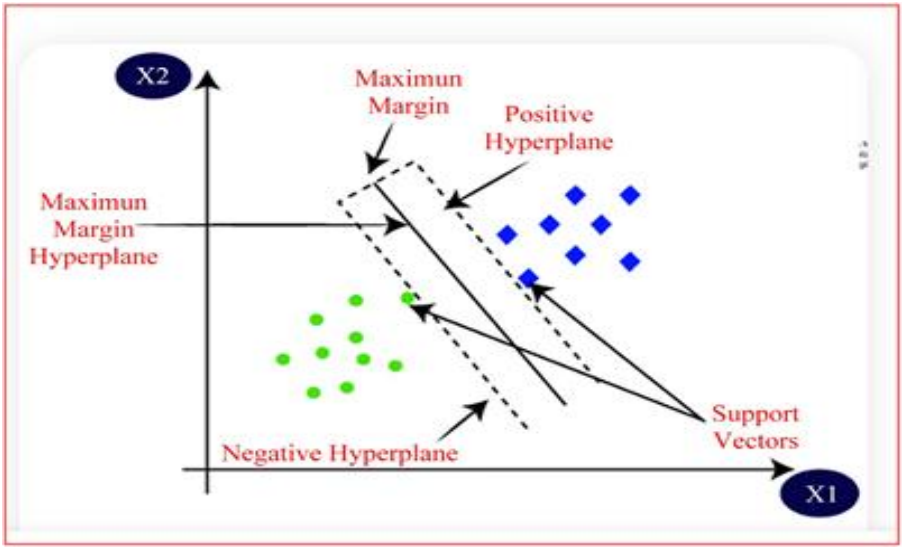


Fig 3.3 Support Vector Machine

3.4 SYSTEM ARCHITECTURE

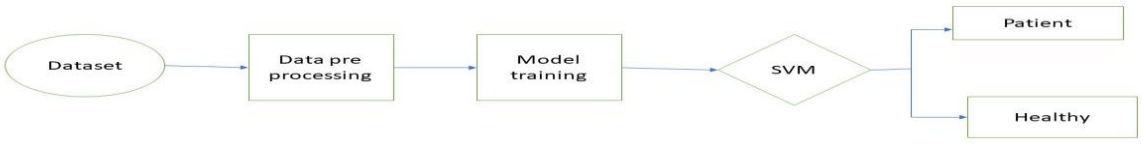


Fig 3.4 System Architecture

3.5 MODULES

The modules used for code are

- Import Data set
- Data pre-processing data
- Data Standardization
- Model training
- Model Evaluation
- Building a predictive system

- **Import data set**

Python libraries make it very easy for us to handle the data and perform typical and complex tasks with a single line of code.

- **Pandas** – This library helps to load the data frame in a 2D array format and has multiple functions to perform analysis tasks in one go.
- **NumPy –** NumPy arrays are very fast and can perform large computations in a very short time.
- **Matplotlib/Seaborn** – This library is used to draw visualizations.
- **Sk learn** – This module contains multiple libraries having pre-implemented functions to perform tasks from data pre-processing to model development and evaluation.

- **Data pre-processing**

In the second part, we will perform data preprocessing to treat outliers, impute missing values, remove duplicate entries, Gaussian transformation for a non-normal distribution, balance the imbalanced data, feature scaling (standardization and normalization), feature binning (converting continuous data to categorical values), feature encoding (converting categorical data to continuous values), and so on.

There are lots of tasks to consider for preprocessing, and this is the most important and time-consuming task of all the stages in machine learning. The data which is obtained from the primary sources is termed the raw data and required a lot of pre-processing before we can derive any conclusions from it or to some modelling on it. Those pre-processing steps are known as data cleanin**g** and it includes, outliers removal, null value imputation, and removing discrepancies of any sort in the data inputs.

- **Data standardization**

  Data standardization is the process of converting data to a common format to enable users to process and analyse it. Data standardization means your data is internally consistent each of your data sources has the same format and labels. When your data is neatly organized with logical descriptions and labels, everyone in your organization can understand it and put it to use.

- **Model training**

  Now we will separate the features and target variables and split them into training and the testing data by using which we will select the model which is performing best on the validation data.

- **Model evaluation**

  From the above accuracies, we can say that Support Vector Machine classifier perform better on the validation data with less difference between the validation and training data. Let's plot the confusion matrix as well for the validation data using the Support Vector Machine model.

- **Building a predictive system**

  Predictive analytics involves certain manipulations on data from existing data sets with the goal of identifying some new trends and patterns. These trends and patterns are then used to predict future outcomes and trends. By performing predictive analysis, we can predict future trends and performance. It is also defined as the prognostic analysis; the word prognostic means prediction. Predictive analytics uses the data, statistical algorithms and machine learning techniques to identify the probability of future outcomes based on historical data.

CHAPTER 4
CONCLUSION

Parkinson's disease is one of the most common neurodegenerative diseases affecting the aging population and is associated with an increased morbidity and mortality. Awareness of the disease manifestations, the treatments, and the progressive long-term course of the disease is necessary for the optimal management of the cases. Tremendous progress has been made in understanding the neuropathology of PD and its progression throughout the nervous system. However, none of these treatments is curative. PD remains a progressive disorder that eventually causes severe disability due to the increasing severity of treatment-resistant motor problems and non-motor symptoms. Modifying factors that lead to the disease progression and in further delaying its disability are the key unmet needs to be addressed by the current and future research efforts

CHAPTER 5
BIBLIOGRAPHY

[1]. Goldman SM, Tanner C. Etiology of Parkinson's disease. In: Jankovic J, Tolosa E, editors. Parkinson's disease and movement disorders, 3rd ed. Baltimore, MD: Lippincott-Williams and Wilkins; 1998. p. 133-58.

[2]. Schrag A, Schott JM. Epidemiological, clinical, and genetic characteristics of early-onset Parkinsonism, Lancet Neuron 2006. 5:355-63.

[3]. Van Den Eeden SK, Tanner CM, Bernstein AL, Fross RD, Leimpeter A, et al. Incidence of Parkinson's disease: Variation by age, gender, and race/ethnicity. Am J Epidemiol 2003;157:1015-22.

[4]. Razdan S, Kaul RL, Motta A, Kaul S, Bhatt RK. Prevalence and pattern of major neurological disorders in rural Kashmir (India) in 1986. Neuroepidemiology 1994;13:113-9.

[5]. Gourie Devi M, Gururaj P, Satishchandra P. Bangalore urban and rural neurological survey. Report submitted to the Indian Council of Medical Research;1995.

[6]. Das SK, Sanyal K. Neuroepidemiology of major neurological disorders in rural Bengal. Neural India 1996;49:47-58.

APPENDIX

A.SOURCE CODE

IMPORTING LIBARIES

```
import NumPy as np
import pandas as pd
import seaborn as sns
import matplotlib.  pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.pre processing import StandardScaler
from sklearn import svm
from sklearn. Metrics import accuracy score
from sklearn import metrics
from sklearn.linear_model import Logistic Regression
from sklearn.ensemble import RandomForestClassifier
from sklearn. Metrics import accuracy score, confusion matrix, classification report
import os
```

PARKINSONS DATA

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

display(parkinsons_data)
```

FIVE ROWS OF DATASET

```
parkinsons_data.head()
```

INFORMATION ABOUT DATASET

```
parkinsons_data.info()
```

DESCRIBE ABOU DATAET

```
parkinsons_data.describe()
```

DATA PRE-PROCESSING

Separate the features & Target

```
X = parkinsons_data.drop(columns=['name','status'], axis=0
```

Y = parkinsons_data['status']

print(X)

Model Training

Support Vector Machine Model

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

#Training the SVM model with training data

model. Fit(X_train, Y_train)

Model Evaluation

Accuracy Score

 # Accuracy score on training data

X_train_prediction = model.predict(X_train)

training_data_accuracy = accuracy_score(Y_train, X_train_prediction)

print('Acuuracy score of training data :', training_data_accuracy)

 # Accuracy score on training data

X_test_prediction = model.predict(X_test)

test_data_accuracy = accuracy_score(Y_test, X_test_prediction)

print('Acuuracy score of test data :', test_data_accuracy)

Building a predictive system

input_data=(124.44500,135.06900,117.49500,0.00431,0.00003,0.00141,0.00167,0.00422,0.02184,0.19700,0.01241,0.01024,0.01685,0.03724,0.00479,25.13500,0.553134,0.775933,-6.650471,0.254498,1.840198,0.103561)

 # Changing input data to a numpy array

input_data_as_numpy_array = np.as array(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] == 1):

```
    print("The Person does not have Parkinsons Disease")
else:
    print("The Person has Parkinsons")

input_data=(124.44500,135.06900,117.49500,0.00431,0.00003,0.00141,0.00167,0.00422,0.0
2184,0.19700,0.01241,0.01024,0.01685,0.03724,0.00479,25.13500,0.553134,0.775933,-
6.650471,0.254498,1.840198,0.103561)

# 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")
```

B.OUTPUT SCREENSHOTS

	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	...	Shimmer:DDA	NHR
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109	0.04374	...	0.06545	0.02211
1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	...	0.09403	0.01929
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633	0.05233	...	0.08270	0.01309
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	...	0.08771	0.01353
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	...	0.10470	0.01767
...	...	...	...	...	...	...	...	...	...	...	...	...	...
190	phon_R01_S50_2	174.188	230.978	94.261	0.00459	0.00003	0.00263	0.00259	0.00790	0.04087	...	0.07008	0.02764
191	phon_R01_S50_3	209.516	253.017	89.488	0.00564	0.00003	0.00331	0.00292	0.00994	0.02751	...	0.04812	0.01810
192	phon_R01_S50_4	174.688	240.005	74.287	0.01360	0.00008	0.00624	0.00564	0.01873	0.02308	...	0.03804	0.10715
193	phon_R01_S50_5	198.764	396.961	74.904	0.00740	0.00004	0.00370	0.00390	0.01109	0.02296	...	0.03794	0.07223
194	phon_R01_S50_6	214.289	260.277	77.973	0.00567	0.00003	0.00295	0.00317	0.00885	0.01884	...	0.03078	0.04398

195 rows × 24 columns

Fig.B.1 Parkinson's data

Out[31]:

	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	...	Shimmer:DDA	NHR
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109	0.04374	...	0.06545	0.02211
1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	...	0.09403	0.01929
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633	0.05233	...	0.08270	0.01309
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	...	0.08771	0.01353
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	...	0.10470	0.01767

5 rows × 24 columns

Fig .B.2 Top five rows of a dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 24 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   name              195 non-null    object
 1   MDVP:Fo(Hz)       195 non-null    float64
 2   MDVP:Fhi(Hz)      195 non-null    float64
 3   MDVP:Flo(Hz)      195 non-null    float64
 4   MDVP:Jitter(%)    195 non-null    float64
 5   MDVP:Jitter(Abs)  195 non-null    float64
 6   MDVP:RAP          195 non-null    float64
 7   MDVP:PPQ          195 non-null    float64
 8   Jitter:DDP        195 non-null    float64
 9   MDVP:Shimmer      195 non-null    float64
 10  MDVP:Shimmer(dB)  195 non-null    float64
 11  Shimmer:APQ3      195 non-null    float64
 12  Shimmer:APQ5      195 non-null    float64
 13  MDVP:APQ          195 non-null    float64
 14  Shimmer:DDA       195 non-null    float64
 15  NHR               195 non-null    float64
 16  HNR               195 non-null    float64
 17  status            195 non-null    int64
 18  RPDE              195 non-null    float64
 19  DFA               195 non-null    float64
 20  spread1           195 non-null    float64
 21  spread2           195 non-null    float64
 22  D2                195 non-null    float64
 23  PPE               195 non-null    float64
dtypes: float64(22), int64(1), object(1)
memory usage: 36.7+ KB
```

Fig.B.3 Information about dataset

Out[35]:

	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	MDVP:Shimmer(dB)	...	Shimmer:DDA
count	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	...	195.000000
mean	154.228641	197.104918	116.324631	0.006220	0.000044	0.003306	0.003446	0.009920	0.029709	0.282251	...	0.046993
std	41.390065	91.491548	43.521413	0.004848	0.000035	0.002968	0.002759	0.008903	0.018857	0.194877	...	0.030459
min	88.333000	102.145000	65.476000	0.001680	0.000007	0.000680	0.000920	0.002040	0.009540	0.085000	...	0.013640
25%	117.572000	134.862500	84.291000	0.003460	0.000020	0.001660	0.001860	0.004985	0.016505	0.148500	...	0.024735
50%	148.790000	175.829000	104.315000	0.004940	0.000030	0.002500	0.002690	0.007490	0.022970	0.221000	...	0.038360
75%	182.769000	224.205500	140.018500	0.007365	0.000060	0.003835	0.003955	0.011505	0.037885	0.350000	...	0.060795
max	260.105000	592.030000	239.170000	0.033160	0.000260	0.021440	0.019580	0.064330	0.119080	1.302000	...	0.169420

8 rows × 23 columns

Fig.B.4 Description about dataset

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

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

     MDVP:Shimmer(dB)  ...  MDVP:APQ  Shimmer:DDA      NHR     HNR      RPDE  \
0               0.426  ...   0.02971      0.06545  0.02211  21.033  0.414783
1               0.626  ...   0.04368      0.09403  0.01929  19.085  0.458359
2               0.482  ...   0.03590      0.08270  0.01309  20.651  0.429895
3               0.517  ...   0.03772      0.08771  0.01353  20.644  0.434969
4               0.584  ...   0.04465      0.10470  0.01767  19.649  0.417356
..                ...  ...       ...          ...      ...     ...       ...
190             0.405  ...   0.02745      0.07008  0.02764  19.517  0.448439
191             0.263  ...   0.01879      0.04812  0.01810  19.147  0.431674
```

Fig. B.5 Data Pre-Processing

Support Vector Machine Model

In [50]:
```
model = svm.SVC(kernel='linear')
#training the SVM model with training data
model.fit(X_train, Y_train)
```

Out[50]:
```
SVC(kernel='linear')
```

Fig.B.6 Using SVM Classifier

```
Acuuracy score of training data : 0.8717948717948718
```

In [52]:
```
# accuracy score on training data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(Y_test, X_test_prediction)
print('Acuuracy score of test data :', test_data_accuracy)
```

```
Acuuracy score of test data : 0.8717948717948718
```

Fig.B.7 Accuracy Score

Building a Predicive System

In [77]:
```
input_data = (124.44500,135.06900,117.49500,0.00431,0.00003,0.00141,0.00167,0.00422,0.02184,0.19700,0.01241,0.01024,0.01685,0.03724,0.00479,25.13500,0.

# 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] == 1):
    print("The Person does not have Parkinsons Disease")
else:
    print("The Person has Parkinsons")
```

```
[1]
The Person does not have Parkinsons Disease
```

Fig.B.8 Predictive System