ABSTRACT

In that paper, we proposed as checking the whole patient Alzheimer using naive Bayes classification. Alzheimer's a progressive disease that demolishes brain's memory and it's regular functioning as well. There isn't any single test till date to diagnose this disease and brain scans alone can't determine whether the person is possessed by it. Currently, the physician believes that a person is affected by Alzheimer's by based the reports of the family members about the behavioural tendencies and observations of the past medical history. AI combined with Machine Learning algorithms might now be able to change this situation. Big Data processing, as the information comes from multiple, heterogeneous, autonomous sources with complex and evolving relationships, and keeps growing. So in that, we will take results of how much percentage patients get disease as a positive information and negative information. The proposed shows a Bi processing model, from the data mining perspective. Using classifiers, we are processing Alzheimer percentage and values are showing as a confusion matrix. We proposed a new classification scheme which can effectively improve the classification performance in the situation that training dataset is available. In that dataset, we have nearly 500 patient details. We will get all that details from there. Then we will good and bad values are using Naive Bayes classifier.

CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCRION

1.1 OVERVIEW

To developed and implement a Machine learning approaches that characterizes the features of the Alzheimer, and proposes an Alzheimer processing model, from the machine learning perspective using SVM, Decision Tree, Naive Bayes Classification and Random Forest.

1.2 INTRODUCTION

Alzheimer disease is caused by both genetic and environmental factors, those affects the brain of a person over time. The genetic changes guarantee a person will develop this disease.

This disease breaks the brain tissue over time. It occurs to people over age 65. However people live with this disease for about 9 years and about 1 among 8 people of age 65 and over have this disease. MMSE (Mini Mental State Examination) score is the main parameter used for prediction of the disease. This score reduces periodically if the person is affected. Those people having MCI have a serious risk of growing dementia.

When the fundamental MCI results in a loss of memory, the situation expects to develop to dementia due to this kind of disease. There is no treatment to cure Alzheimer's disease. In advanced stages of the disease, complications like dehydration, malnutrition or infection occurs which leads to death. The diagnosis at MCI stage will help the person to focus on healthy approach of life, and good planning to take care of memory loss.

1.3 EXISTING SYSTEM

In this existing method is we used Naive Bayes Algorithm for predict that Alzheimer details. Naive Bayes algorithm makes only clustering process and it will segregate the person who are all suffering from Alzheimer**.** Naive Bayes Algorithm is slow process for classify all the given details. In that main disadvantage is time efficiency. Patient's Alzheimer details start with large-volume, heterogeneous, autonomous sources with distributed and decentralized control, and seek to explore complex and evolving relationships among data.

LIMITATIONS / DISADVANTAGES:

- It will take more time to calculate our dataset which we having our Alzheimer dataset.
- Discover patterns using traditional data mining tools may not use for predictions.
- They concluded that drug treatment for patients in the young age group can be delayed whereas; patients in the old age group should be prescribed drug treatment immediately.
- Prediction and classification of various type of Alzheimer using classification algorithm was carried out in Alzheimer dataset.

1.4 PROPOSED SYSTEM

The raw data set is given as input to the system. The unstructured voluminous input data can be obtained from various Electronic Health Record (EHR) / Patient Health Record (PHR), Clinical systems and external sources (government sources, laboratories, pharmacies, insurance companies etc.). Algorithms used is, SVM and Decision Tree algorithm to predict the results accurately. To explore Big Data, proposed system analysed several challenges at the data, model, and system levels. To support Big Data mining, high-performance computing platforms are required, which impose systematic designs to unleash the full power of the Big Data.

At the data level, the autonomous information sources and the variety of the data collection environments, often result in data with complicated conditions, such as missing/uncertain values. In other situations, privacy concerns, noise, and errors can be introduced into the data, to produce altered data copies.

ADVANTAGES:

- Fastest path to business value from raw big data.
- Machine learning models to detect the Alzheimer early.
- Discover patterns using machine learning algorithm that identify the best mode of treatment for Alzheimer across different age.
- Novel, high value business insights driving growth and profitability
- Leverage existing skills and investments
- Minimal time, cost and effort spent

- These technical challenges are common across a large variety of application domains, and therefore not cost-effective to address in the context of one domain alone.

1.5 SYSTEM ARCHITECTURE:

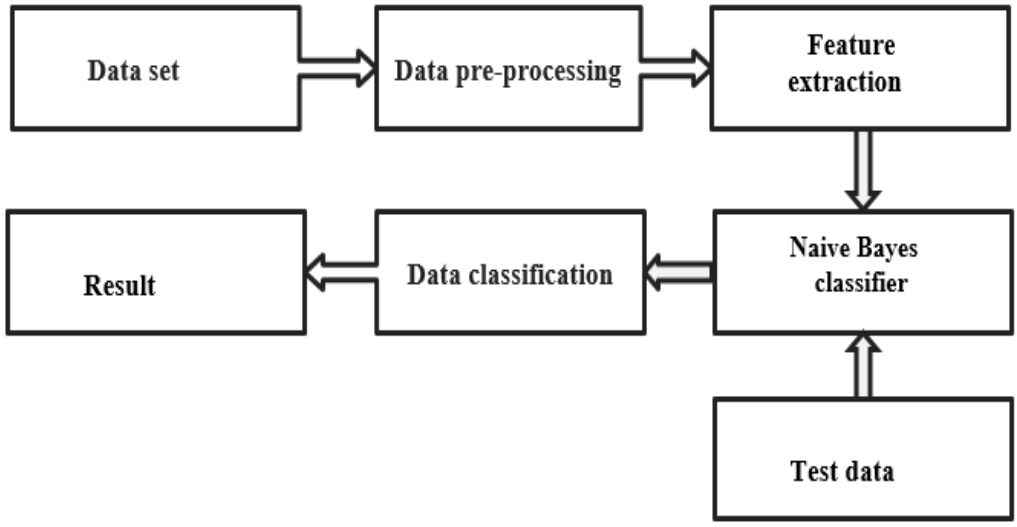

Fig.1.1: System Architecture.

1.6 HARDWARE AND SOFTWARE SPECIFICATION:

a. Hardware:

i. 1 GB RAM
ii. 80 GB Hard Disk
iii. Intel Processor
iv. LAN

b. Software:

i. Windows OS
ii. 2Python GUI or Anaconda Navigator

c. System Requirement:

Operating System: Windows 7 Ultimate 32 bit / Windows XP.

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

- **[2] Author:** M. Graña et al.
- **Title:** Computer Aided Diagnosis system for Alzheimer Disease using brain Diffusion Tensor Imaging features selected by Pearson's correlation.
- **Abstract:** We present a new automated system for the detection of brain magnetic resonance images (MRI) affected by Alzheimer's disease (AD). The MRI is analysed by means of multiscale analysis (MSA) to obtain its fractals at six different scales. The extracted fractals are used as features to differentiate healthy brain MRI from those of AD by a support vector machine (SVM) classifier. The result of classifying 93 brain MRIs consisting of 51 images of healthy brains and 42 of brains affected by AD, using leave-one-out cross-validation method, yielded 99.18% ± 0.01 classification accuracy, 100% sensitivity, and 98.20% ± 0.02 specificity. These results and a processing time of 5.64 seconds indicate that the proposed approach may be an efficient diagnostic aid for radiologists in the screening for AD.
- **[3] Author:** M. Dyrba et al
- **Title:** Combining DTI and MRI for the automated detection of Alzheimer's disease using a large European multi-center dataset.
- **Abstract:** Diffusion tensor imaging (DTI) allows assessing neuronal fiber tract integrity in vivo to support the diagnosis of Alzheimer's disease (AD). It is an open research question to which extent combinations of different neuroimaging techniques increase the detection of AD. In this study we examined different methods to combine DTI data and structural T 1-weighted magnetic resonance imaging (MRI) data. Further, we applied machine learning techniques for automated detection of AD. We used a sample of 137 patients with clinically probable AD (MMSE 20.6 ±5.3) and 143 healthy elderly controls, scanned in nine different scanners, obtained from the recently created framework of the European DTI study on Dementia (EDSD). For diagnostic classification we used the DTI derived indices fractional anisotropy (FA) and mean diffusivity (MD) as well as grey matter density (GMD) and white matter density (WMD) maps from anatomical MRI. We performed voxel-based classification using a

Support Vector Machine (SVM) classifier with tenfold cross validation. We compared the results from each single modality with those from different approaches to combine the modalities. For our sample, combining modalities did not increase the detection rates of AD. An accuracy of approximately 89% was reached for GMD data alone and for multimodal classification when GMD was included. This high accuracy remained stable across each of the approaches. Evaluating the classification performance when adding other modalities, e.g. functional MRI or FDG-PET.

- **[4] Author:** S. Haller et al.
- **Title:** Individual classification of mild cognitive impairment subtypes by support vector machine analysis of white matter DTI
- **Abstract:** MCI was recently subdivided into sd-aMCI, sd-fMCI, and md-aMCI. The current investigation aimed to discriminate between MCI subtypes by using DTI. MATERIALS AND METHODS: Sixty-six prospective participants were included: 18 with sd-aMCI, 13 with sd-fMCI, and 35 with md-aMCI. Statistics included group comparisons using TBSS and individual classification using SVMs. RESULTS: The group-level analysis revealed a decrease in FA in md-aMCI versus sd-aMCI in an extensive bilateral, right-dominant network, and a more pronounced reduction of FA in md-aMCI compared with sd-fMCI in right inferior fronto-occipital fasciculus and inferior longitudinal fasciculus. The comparison between sd-fMCI and sd-aMCI, as well as the analysis of the other diffusion parameters, yielded no significant group differences. The individual-level SVM analysis provided discrimination between the MCI subtypes with accuracies around 97%. The major limitation is the relatively small number of cases of MCI. CONCLUSIONS: Our data show that, at the group level, the md-aMCI subgroup has the most pronounced damage in white matter integrity. Individually, SVM analysis of white matter FA provided highly accurate classification of MCI subtypes.
- **[5] Author:** W. Lee, B. Park, and K. Han.
- **Title:** "Classification of diffusion tensor images for the early detection of Alzheimer's disease".
- **Abstract:** Early detection of Alzheimer's disease (AD) is important since treatments are more efficacious when used at the beginning of the disease. Despite significant advances in diagnostic methods for AD, there is no single diagnostic method for AD

with high accuracy. We developed a support vector machine (SVM) model that classifies mild cognitive impairment (MCI) and normal control subjects using probabilistic tractography and tract-based spatial statistics of diffusion tensor imaging (DTI) data. MCI is an intermediate state between normal aging and AD, so finding MCI is important for an early diagnosis of AD. The key features of DTI data we identified through extensive analysis include the fractional anisotropy (FA) values of selected voxels, their average FA value, and the volume of fiber pathways from a pre-defined seed region. In particular, the volume of the fiber pathways to thalamus is the most powerful single feature in classifying MCI and normal subjects regardless of the age of the subjects. The best performance achieved by the SVM model in a 10-fold cross validation and in independent testing was sensitivity of 100%, specificity of 100% and accuracy of 100%.

- **[6] Author:** T. M. Nir et al.
- **Title:** "Diffusion weighted imaging-based maximum density path analysis and classification of Alzheimer's disease".
- **Abstract:** Characterizing brain changes in Alzheimer's disease (AD) is important for patient prognosis and for assessing brain deterioration in clinical trials. In this diffusion weighted imaging study, we used a new fiber-tract 6odelling method to investigate white matter integrity in 50 elderly controls (CTL), 113 people with mild cognitive impairment, and 37 AD patients. After clustering tractography using a region-of-interest atlas, we used a shortest path graph search through each bundle's fiber density map to derive maximum density paths (MDPs), which we registered across subjects. We calculated the fractional anisotropy (FA) and mean diffusivity (MD) along all MDPs and found significant MD and FA differences between AD patients and CTL subjects, as well as MD differences between CTL and late mild cognitive impairment subjects. MD and FA were also associated with widely used clinical scores. As an MDP is a compact low-dimensional representation of white matter organization, we tested the utility of diffusion tensor imaging measures along these MDPs as features for support vector Machine based classification of AD.

CHAPTER-3

MODULES AND DATAFLOW & UML DIAGRAMS

MODULES

1. DATA COLLECTION
2. DATA PRE-PROCESSING
3. FEATURE EXTRATION
4. EVALUATION MODEL

3.1 DATA COLLECTION

Data used in this paper is a set of product reviews collected from credit card transactions records. This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which you already know the target answer is called labelled data.

3.2 DATA PRE-PROCESSING

Organize your selected data by formatting, cleaning and sampling from it.

Three common data pre-processing steps are:

- **Formatting:** The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file.
- **Cleaning:** Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be anonymized or removed from the data entirely.
- **Sampling:** There may be far more selected data available than you need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative

sample of the selected data that may be much faster for exploring and prototyping solutions before considering the whole dataset.

3.3 FEATURE EXTRATION

Next thing is to do Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally, our models are trained using Classifier algorithm. We use classify module on Natural Language Toolkit library on Python. We use the labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data. The chosen classifiers were Random-forest. These algorithms are very popular in text classification tasks.

3.4 EVALUATION MODEL

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and over fitted models. There are two methods of evaluating models in data science, Hold-Out and Cross-Validation. To avoid over fitting, both methods use a test set (not seen by the model) to evaluate model performance. Performance of each classification model is estimated base on its averaged. The result will be in the visualized form. Representation of classified data in the form of graphs. Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

3.5 DATAFLOW DIAGRAMS

LEVEL 0

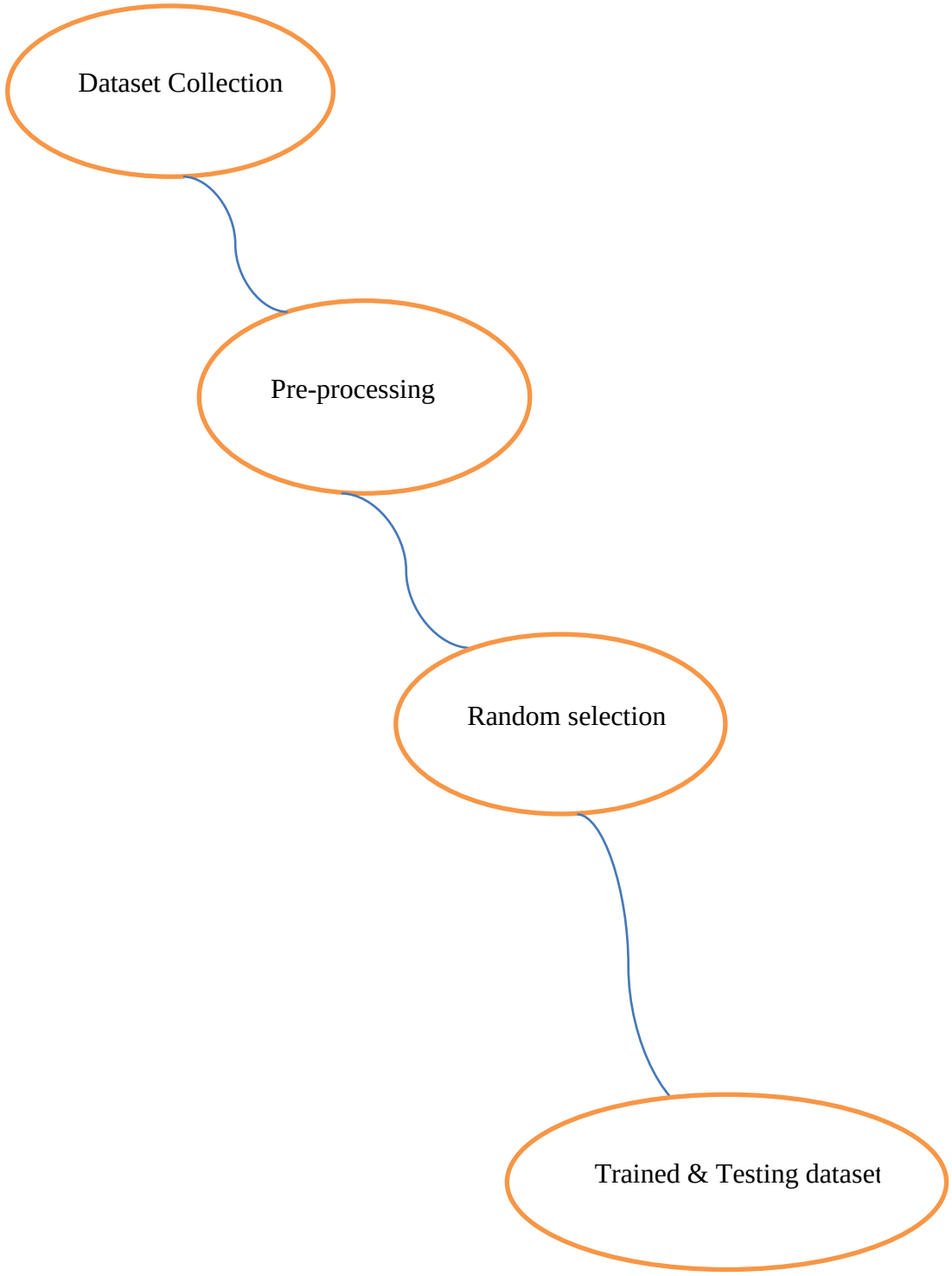

Fig.3.1: Level-0 flow diagram.

LEVEL 1

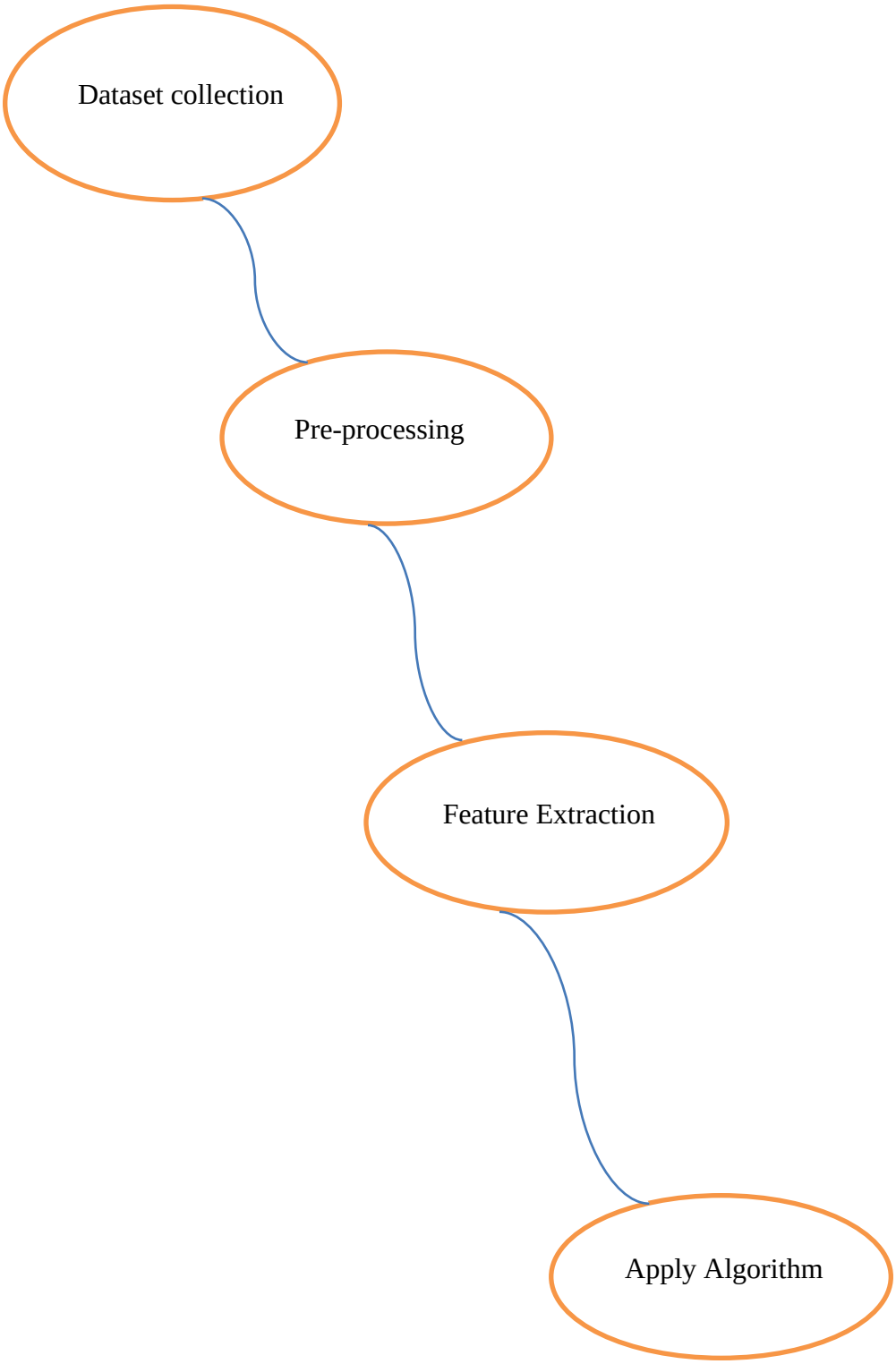

Fig.3.2: Level-1 flow diagram.

LEVEL 2

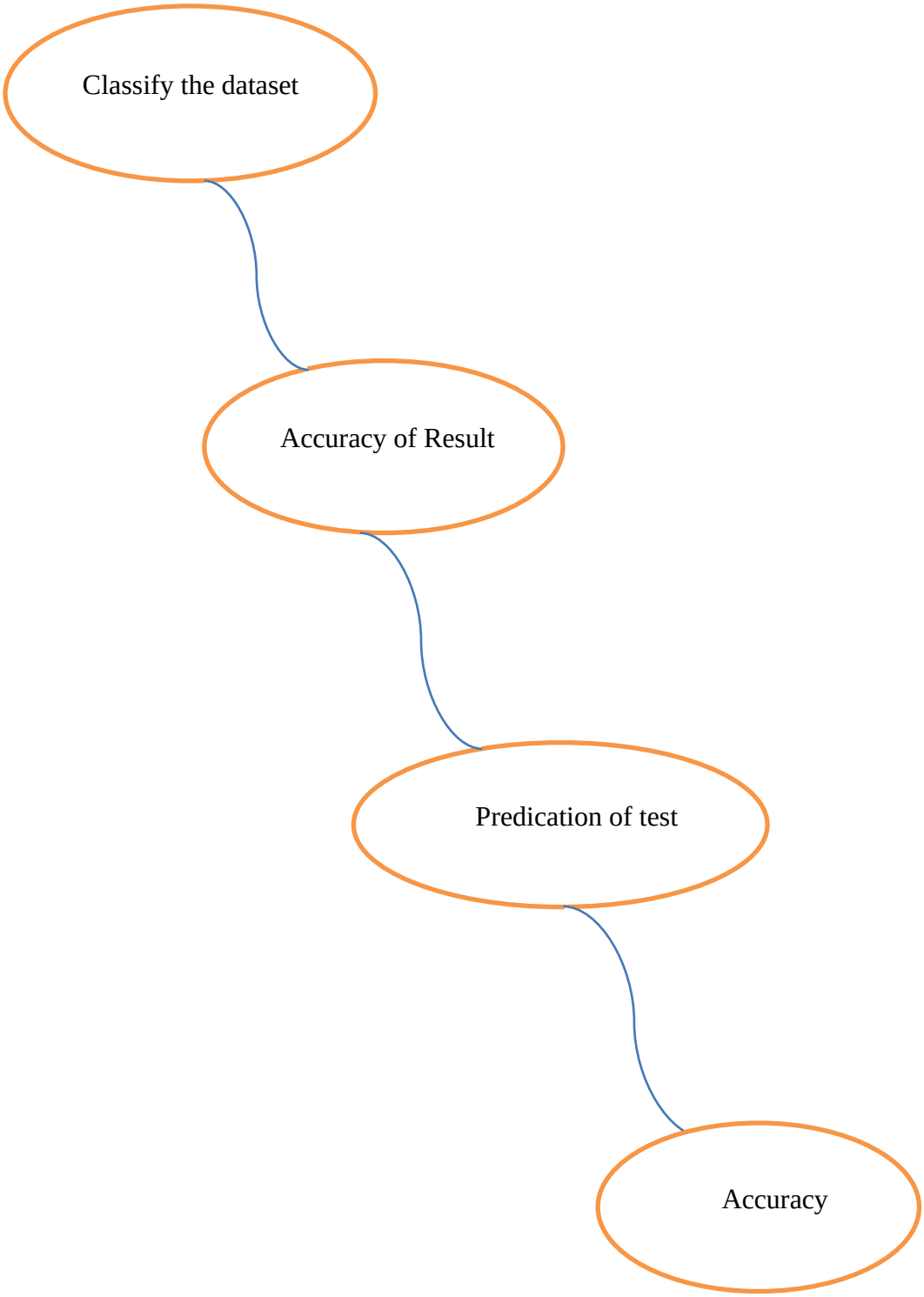

Fig.3.3: Level-2 flow diagram.

3.6 UML DIAGRAMS

The Unified Modelling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

- Actors
- business processes
- (logical) components
- Activities
- programming language statements
- database schemas, and
- Reusable software components.

UML combines best techniques from data modelling (entity relationship diagrams), business modelling (work flows), object modelling, and component modelling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML has synthesized the notations of the Booch method, the Object-modelling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modelling language. UML aims to be a standard modelling language which can model concurrent and distributed systems.

USE CASE DIAGRAM:

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997. OMG is continuously putting effort to make a truly industry standard. UML stands for Unified Modelling Language.

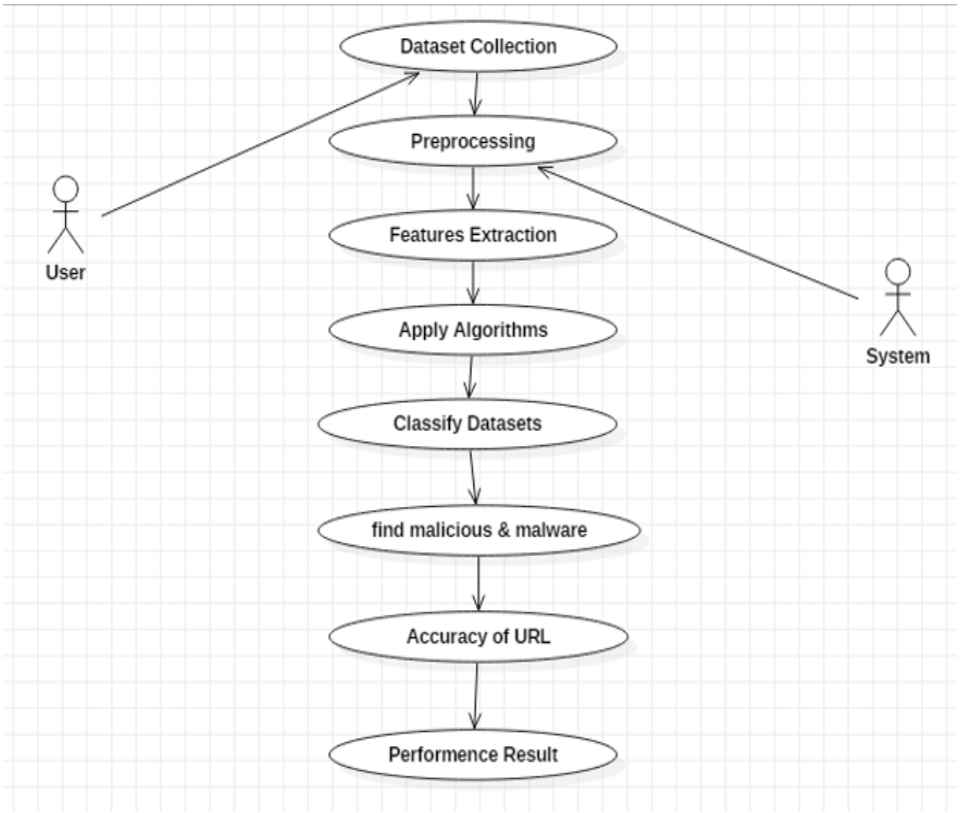

Fig.3.4: Use case diagram.

CLASS DIAGRAM:

The class diagram is the main building block of object-oriented modelling. It is used for general conceptual modelling of the systematic of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modelling.[1] The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

The top compartment contains the name of the class. It is printed in bold and centred, and the first letter is capitalized. The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase. The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

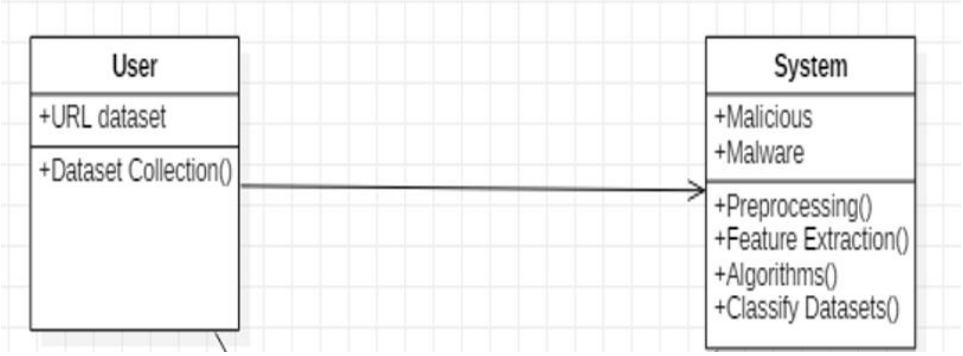

Fig.3.5: Class diagram.

SEQUENCE DIAGRAM:

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioural classifier that represents a declaration of an offered behaviour. Each use case specifies some behaviour, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behaviour of the subject without reference to its internal structure. These behaviours, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behaviour, including exceptional behaviour and error handling.

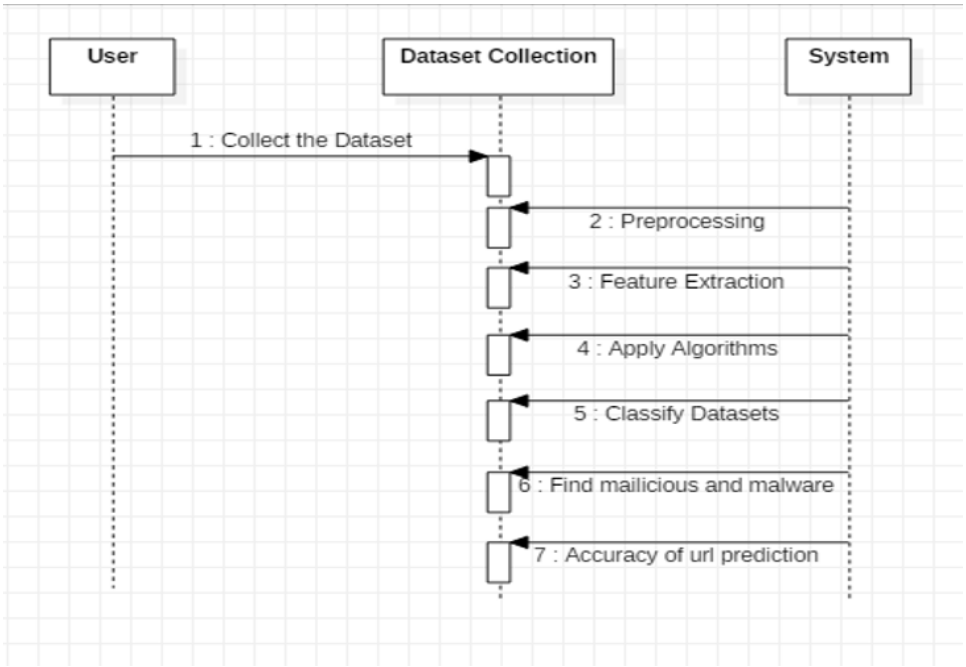

Fig.3.6: Sequence diagram.

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

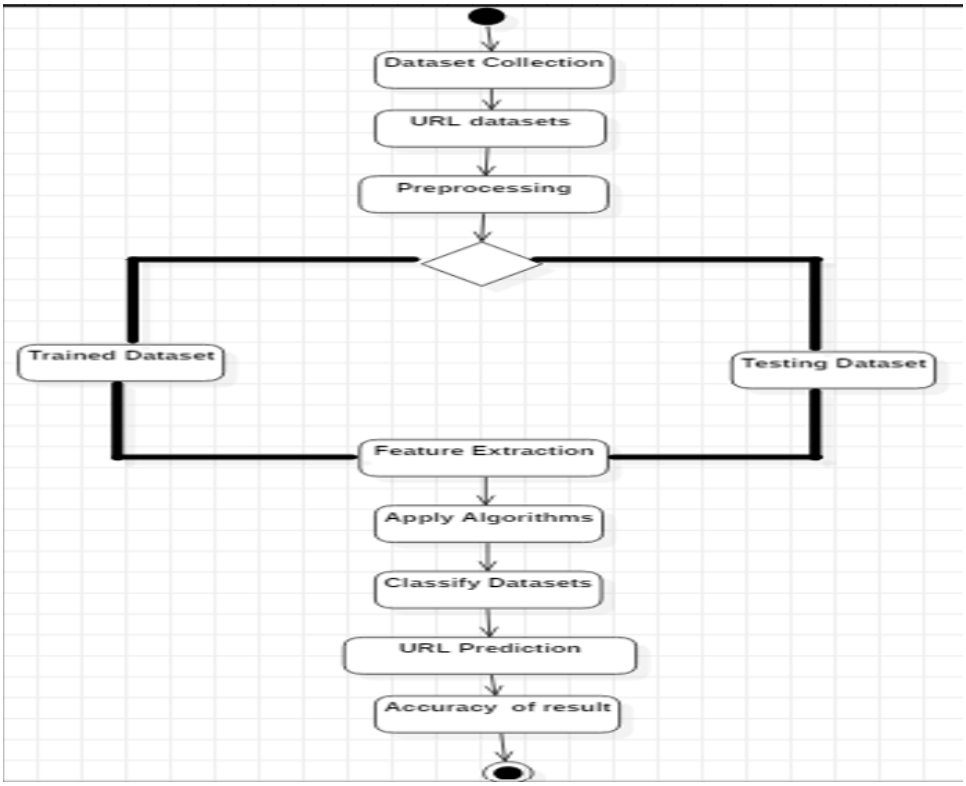

Fig.3.7: Activity diagram.

CHAPTER-4

SOFTWARE REQUIREMENTS

4.1 PYTHON OVERVIEW

4.1.1 WHAT IS PYTHON?

Python is a general purpose, dynamic, high-level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.

4.1.2 PYTHON HISTORY AND VERSIONS:

- Python laid its foundation in the late 1980s.
- The implementation of Python was started in December 1989 by Guido Van Rossum at CWI in Netherland.
- In February 1991, Guido Van Rossum published the code (labeled version 0.9.0) to alt.sources.
- In 1994, Python 1.0 was released with new features like lambda, map, filter, and reduce.
- Python 2.0 added new features such as list comprehensions, garbage collection systems.
- On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify the fundamental flaw of the language.
- ABC programming language is said to be the predecessor of Python language, which was capable of Exception Handling and interfacing with the Amoeba Operating System.

The following programming languages influence Python:

- ABC language.
- Modula-3

4.1.3 PYTHON FEATURES:

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.
- **Scalable:** Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below:

- IT supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

4.1.4 WHERE TO USE PYTHON?

Even though Python started as a general-purpose programming language with no particular application as its focus, it has emerged as the language of choice for developers in some application areas over the last few years. Some important applications of Python are summarized below:

- **Data Science:** Python experienced a recent emergence in popularity charts, mainly because of its Data science libraries. A huge amount of data is being generated today by web applications, mobile applications, and other devices. Companies need business insights from this data. Today Python has become the language of choice for data scientists. Python libraries like NumPy, Pandas, and Matplotlib are extensively used in the process of data analysis, including the collection, processing and cleansing of data sets, applying mathematical algorithms, and generating visualizations for the benefit of users. Commercial and community Python distributions by third-parties such as Anaconda and Active-State provide all the essential libraries required for data science.
- **Machine Learning:** This is another key application area of Python. Python libraries such as Scikit-learn, Tensor flow and NLTK are widely used for the prediction of trends like customer satisfaction, projected values of stocks, etc. Some of the real-world applications of machine learning include medical diagnosis, statistical arbitrage, basket analysis, sales prediction, etc.
- **Web Development:** This is another application area in which Python is becoming popular. Web application framework libraries like Django, Pyramid, Flask, etc. make it very easy to develop and deploy simple as well as complex web applications. These frameworks are used extensively by various IT companies. Dropbox, for example, uses Django as a backend to store and synchronize local folders. Today, most of the web servers are compatible with WSGI (Web Server Gateway Interface) - a specification for the universal interface between Python web frameworks and web servers. All leading web servers such as Apache, IIS, Nginx etc can now host Python web applications. Google's App Engine hosts web applications built with almost all Python web frameworks.

- **Image Processing:** The OpenCV library is commonly used for face detection and gesture recognition. OpenCV is a C++ library but has been ported to Python. Because of the rapid development of this feature, Python is a very popular choice from image processing.
- **Game Development:** Python is a popular choice for game developers. The PyGame library is extensively used for building games for desktop as well as for mobile platforms. PyGame applications can be installed on Android too.
- **Embedded Systems and IoT:** Another important area of Python application is in embedded systems. Raspberry Pi is a very popular yet low-cost single-board computer. It is extensively used in automation products, robotics, IoT, and kiosk applications. Popular microcontrollers like Arduino are used in many IoT products and are being programmed with Python. A lightweight version of Python called Micropython has been developed, especially for microcontrollers. A special Micropython-compatible controller called PyBoard has also been developed.
- **Android Apps:** Although Android apps are predominantly developed using Android SDK, which is similar to Java, Python can also be used to develop Android apps. Python's Kivy library has all the functionalities required for a mobile application.
- **Automated Jobs:** Python is extremely useful and widely used for automating CRON (Command Run ON) jobs. Certain tasks like backups, defined in Python scripts, can be scheduled to be invoked automatically by the operating system scheduler to be executed at predefined times. Python is embedded as a scripting language in many popular software products. This is similar to VBA used for writing macros in Excel, PowerPoint, etc. Python API is integrated with Maya, PaintShop Pro, etc.
- **Rapid Development Tool:** The standard distribution of Python, as developed by Rossum and maintained by Python Software Foundation, is called CPython, which is a reference implementation. Its alternative implementations - Jython, the JRE implementation of Python and IronPython - the .NET implementation, interact seamlessly with Java and C#, respectively. For example, Jython can use all Java libraries such as Swing. So the development time can be minimized by using simpler Python syntaxes and Java libraries for prototyping the software product.

4.2 PYTHON ENVIRONMENT

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

4.2.1 PYTHON WINDOWS INSTALLATION:

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to https://www.python.org/downloads/.
- Follow the link for the Windows installer python-XYZ.msi file where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

4.2.2 SETTING PATH AT WINDOWS:

- To add the Python directory to the path for a particular session in Windows-
- At the command prompt − type path %path%;C:\Python and press Enter.

Note − C:\Python is the path of the Python directory.

4.3 ANACONDA NAVIGATOR

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, mac OS and Linux.

Why use Navigator?

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages, and use multiple

environments to separate these different versions. The command line program conda is both a package manager and an environment manager, to help data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages and update them, all inside Navigator.

What applications can i access using navigator?

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QTConsole
- Spyder
- VSCode
- Glueviz
- Orange 3 App
- Rodeo
- RStudio

Advanced conda users can also build your own Navigator applications.

How can I run code with Navigator?

The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute your code. You can also use Jupyter Notebooks the same way. Jupyter Notebooks are an increasingly popular system that combine your code, descriptive text, output, images and interactive interfaces into a single notebook file that is edited, viewed and used in a web browser.

What's new in 1.9?

- Add support for Offline Mode for all environment related actions.
- Add support for custom configuration of main windows links.

- Numerous bug fixes and performance enhancements.

4.4 ANACONDA ENVIRONMENT:

Steps to Download Anaconda for windows:

Step 1: Visit www.anaconda.com. https://www.anaconda.com/

Step 2: In top Left corner Click on Product. From the dropdown menu select Individual Edition option.

Step 3: Now, Download the Anaconda edition you need by clicking on download button.

Step 4: After downloading, you all will find that the .exe file for Anaconda has been downloaded. Click on the file and click Next.

Step 5: Agree to the license and click the next button. Also, select the location where you want your file to be downloaded and click next.

Step 6: Click on install.

Step 7: Click on Next button. Then Finish the process and enjoy your Anaconda distribution.

Setting path at Windows:

- After the installation is done, we need to set up the environment variable.
- Go to Control Panel -> System and Security -> System
- Under the Advanced System Setting option click on Environment Variables.
- Now, we have to alter the "Path" variable under System variables so that it also contains the path to the Anaconda environment. Select the "Path" variable and click on the Edit button.
- We will see a list of different paths, click on the New button and then add the path where Anaconda is installed.
- Click on OK, Save the settings and it is done!!
- Now to check whether the installation is done correctly, open the command prompt and type anaconda-navigator. It will start the anaconda navigator App if installed correctly.

4.5 PYTHON STANDARD LIBRARIES

4.5.1 NUMPY

Python NumPy is a general-purpose array processing package which provides tools for handling the n-dimensional arrays. It provides various computing tools such as comprehensive mathematical functions, linear algebra routines. NumPy provides both the flexibility of Python and the speed of well-optimized compiled C code. It's easy to use syntax makes it highly accessible and productive for programmers from any background.

This NumPy tutorial helps you learn the fundamentals of NumPy from Basics to Advance, like operations on NumPy array, matrices using a huge dataset of NumPy – programs and projects.

What is NumPy?

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases. Installation:

```
pip install numpy
```

Importing numpy:

```
import numpy as np
```

Advantages of NumPy:

Below are the points that explain the advantages of NumPy:

- The core of Numpy is its arrays. One of the main advantages of using Numpy arrays is that they take less memory space and provide better runtime speed when compared with similar data structures in python (lists and tuples).
- Numpy support some specific scientific functions such as linear algebra. They help us in solving linear equations.
- Numpy support vectorized operations, like elementwise addition and multiplication, computing Kronecker product, etc. Python lists fail to support these features.
- It is a very good substitute for MATLAB, OCTAVE, etc as it provides similar functionalities and supports with faster development and less mental overhead (as python is easy to write and comprehend)
- NumPy is very good for data analysis.

4.5.2 PANDAS

Pandas is an open-source library that is built on top of NumPy library. It is a Python package that offers various data structures and operations for manipulating numerical data and time series. It is mainly popular for importing and analyzing data much easier. Pandas is fast and it has high-performance & productivity for users.

This Pandas Tutorial will help learning Pandas from Basics to advance data analysis operations, including all necessary functions explained in detail.

History:

Pandas were initially developed by Wes McKinney in 2008 while he was working at AQR Capital Management. He convinced the AQR to allow him to open source the Pandas. Another AQR employee, Chang She, joined as the second major contributor to the library in 2012. Over time many versions of pandas have been released. The latest version of the pandas is 1.4.1

Advantages:

- Fast and efficient for manipulating and analyzing data.
- Data from different file objects can be loaded.
- Easy handling of missing data (represented as NaN) in floating point as well as non-floating point data.

- Size mutability: columns can be inserted and deleted from DataFrame and higher dimensional objects
- Data set merging and joining.
- Flexible reshaping and pivoting of data sets.
- Provides time-series functionality.
- Powerful group by functionality for performing split-apply-combine operations on data sets.

Getting Started:

The first step of working in pandas is to ensure whether it is installed in the Python folder or not. If not then we need to install it in our system using pip command. Type cmd command in the search box and locate the folder using cd command where python-pip file has been installed. After locating it, type the command:

```
pip install pandas
```

After the pandas have been installed into the system, you need to import the library. This module is generally imported as:

```
import pandas as pd
```

Here, pd is referred to as an alias to the Pandas. However, it is not necessary to import the library using the alias, it just helps in writing less amount code every time a method or property is called. Pandas generally provide two data structures for manipulating data, They are:

1. Series:

Pandas Series is a one-dimensional labeled array capable of holding data of any type (integer, string, float, python objects, etc.). The axis labels are collectively called indexes. Pandas Series is nothing but a column in an excel sheet. Labels need not be unique but must be a hashable type. The object supports both integer and label-based indexing and provides a host of methods for performing operations involving the index.

2. DataFrame:

Pandas DataFrame is a two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-

dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the data, rows, and columns.

Why Pandas is used for Data Science?

Pandas are generally used for data science but have you wondered why? This is because pandas are used in conjunction with other libraries that are used for data science. It is built on the top of the NumPy library which means that a lot of structures of NumPy are used or replicated in Pandas. The data produced by Pandas are often used as input for plotting functions of Matplotlib, statistical analysis in SciPy, and machine learning algorithms in Scikit-learn.

Pandas program can be run from any text editor but it is recommended to use Jupyter Notebook for this as Jupyter given the ability to execute code in a particular cell rather than executing the entire file. Jupyter also provides an easy way to visualize pandas data frames and plots.

4.5.3 MATPLOTLIB

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Installation:

Windows, Linux and macOS distributions have matplotlib and most of its dependencies as wheel packages. Run the following command to install matplotlib package:

python -mpip install -U matplotlib

Importing matplotlib:

from matplotlib import pyplot as plt (or) **import matplotlib.pyplot as plt**

Basic plots in Matplotlib:

Matplotlib comes with a wide variety of plots. Plots helps to understand trends, patterns, and to make correlations. They're typically instruments for reasoning about quantitative information. Some of the sample plots are covered here.

Advantages Of Matplotlib:

- **Matplotlib Is Open Source –** One of the primary advantages of Matplotlib is that it is an open-source package. Because of this, you can use it in whatever way you want. You don't need to pay any money for this tool to anyone. You can also use it for both academic and commercial purposes.
- **Written In Python** – Yet another benefit of Matplotlib is that it is all written in Python. As you use Python programming language to do data processing, plotting its result in Python again makes it so much more easier.
- **Customizable & Extensible** – As its written in Python, you will also be able to customize the package (if required) to suit your requirements. In addition to this, you can always extend its functionalities and contribute it back to the open-source community. Since Python also has other useful packages, you can also make use of those packages' functionalities to extend Matplotlib.
- **Portable & Cross Platform** – Since its written in Python it is easily portable to any system which can run Python. It also works smoothly on Windows, Linux & Mac OS.
- **Easy to learn** – Because the Python language is much easier to learn, any packages written using this language becomes so much more easier. You will not find Matplotlib any different either when it comes to this.

4.5.4 SKLEARN

What is Scikit-Learn (Sklearn)

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction via

a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Origin of Scikit-Learn

It was originally called scikits.learn and was initially developed by David Cournapeau as a Google summer of code project in 2007. Later, in 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, and Vincent Michel, from FIRCA (French Institute for Research in Computer Science and Automation), took this project at another level and made the first public release (v0.1 beta) on 1st Feb. 2010.

Installation

If you already installed NumPy and Scipy, following are the two easiest ways to install scikit-learn −

Using pip

Following command can be used to install scikit-learn via pip −

```
pip install -U scikit-learn
```

Using conda

Following command can be used to install scikit-learn via conda −

```
conda install scikit-learn
```

On the other hand, if NumPy and Scipy is not yet installed on your Python workstation then, you can install them by using either pip or conda.

Another option to use scikit-learn is to use Python distributions like Canopy and Anaconda because they both ship the latest version of scikit-learn.

4.5.5 SEABORN

Seaborn in python issued to create graphics which is easy to manage. Seaborn is a library provided by python, which basically helps to visualize the data and make it more and more undertakable by the user. With the help of the library, we can plot our data and make a graphical representation of it. Internally this library uses matplotlib; in short, it is based on matplotlib only. This also makes it efficient to create attractive and more informative graphics representations of our data. This library is integrated with the panda's

data structure. In the coming section of the tutorial, we will see how to use a seaborn library to make interactive graphics and understand its working in detail. Seaborn is a library mostly used for statistical plotting in Python. It is built on top of Matplotlib and provides beautiful default styles and color palettes to make statistical plots more attractive.

Installing Seaborn and getting started

Before using Seaborn, we need to install it and here I am going to show various ways of installing it on your computer.

Using Pip Installer

pip is a de facto standard package-management system used to install and manage software packages written in Python.

```
pip install seaborn
```

Using Anaconda

Anaconda is a package manager, an environment manager, and Python distribution that contains a collection of many open source packages. If you need additional packages after installing Anaconda, you can use Anaconda's package manager or conda to install those packages.

```
conda install seaborn
```

Advantages:

We have some advantages of using seaborn in our application which is as follows;

- By using the seaborn library, we can easily represent our data on a plot.
- This library is used to visualize our data; we do not need to take care of the internal details; we just have to pass our data set or data inside the relplot() function, and it will calculate and place the value accordingly.
- Inside this, we can switch to any other representation of data using the 'kind' property inside it.
- It creates an interactive and informative plot to representation our data; also, this is easy for the user to understand and visualize the records on the application.
- It uses static aggregation for plot generation in python.

CHAPTER-5

DOMAIN SPECIFICATION AND ALGORITHM

5.1 MACHINE LEARNING

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results.

Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive modelling. The machine receives data as input, use an algorithm to formulate answers.

A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation. Machine learning is also used for a variety of task like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

Machine Learning vs. Traditional Programming:

Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

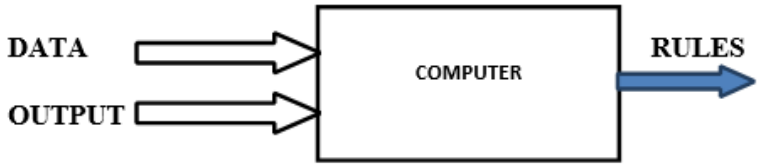

Fig.5.1: Machine Learning model.

How does Machine learning work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the learning and inference. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the data. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a feature vector. You can think of a feature vector as a subset of data that is used to tackle a problem. The machine uses some fancy algorithms to simplify the reality and transform this discovery into a model. Therefore, the learning stage is used to describe the data and summarize it into a model.

For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant:

This is the model:

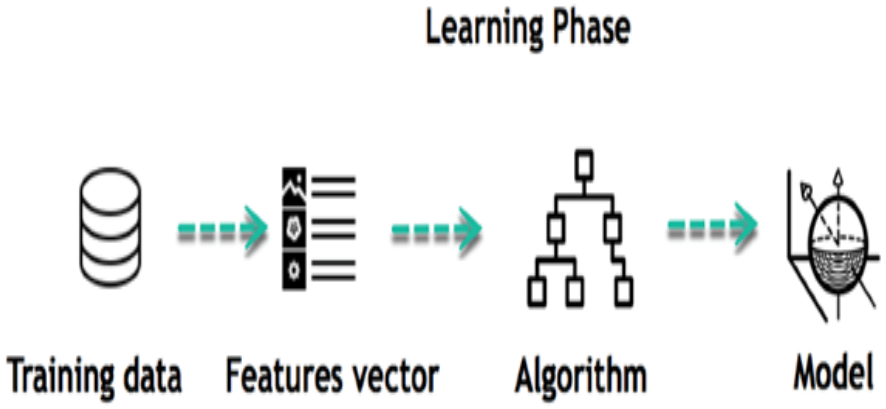

Fig.5.2: Learning Phase Diagram

Inferring:

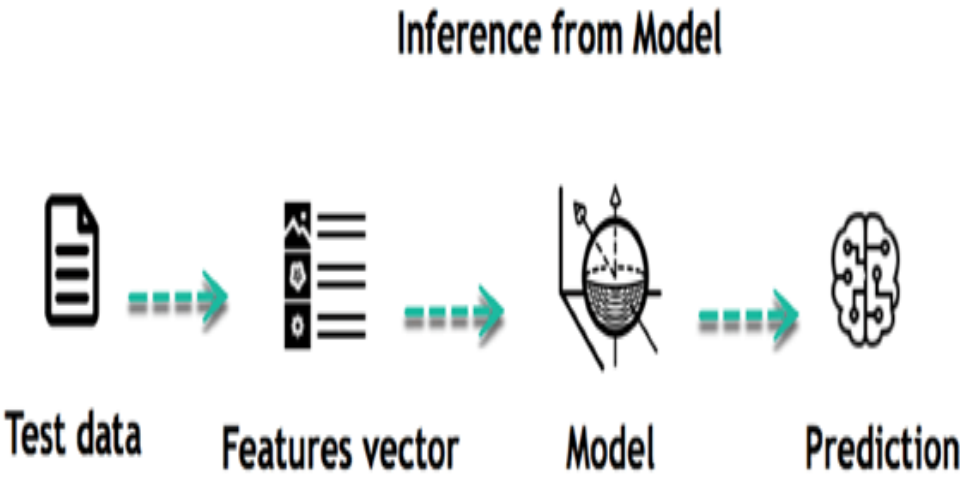

Fig.5.3: Inference from Model.

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

The life of Machine Learning programs is straightforward and can be summarized in the following points:

1. Define a question
2. Collect data
3. Visualize data
4. Train algorithm
5. Test the Algorithm
6. Collect feedback
7. Refine the algorithm
8. Loop 4-7 until the results are satisfying
9. Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

Machine learning Algorithms and where they are used?

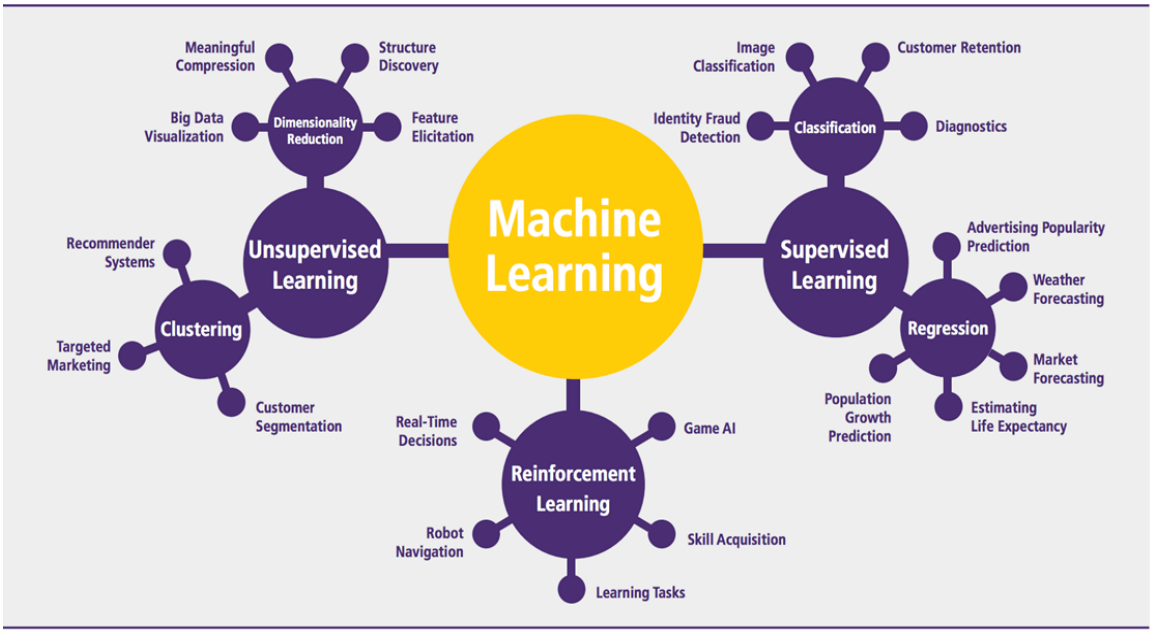

Fig.5.4: Machine Learning Architecture.

Machine learning can be grouped into two broad learning tasks: Supervised and Unsupervised. There are many other algorithms

1 Supervised learning:

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans.

You can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

I. Classification task

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male

or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female. The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class).

II. Regression task

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

Algorithm Name	Description	Type
Linear regression	Finds a way to correlate each feature to the output to help predict future values.	Regression
Logistic regression	Extension of linear regression that's used for classification tasks. The output variable 3is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colors)	Classification
Decision tree	Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (e.g., if a feature is a color, each possible color becomes a new branch) until a final decision output is made	Regression Classification

Naive Bayes	The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event.	Regression Classification
Support vector machine	Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divided the classes. It is best used with a non-linear solver.	Regression (not very common) Classification
Random forest	The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest generates many times simple decision trees and uses the 'majority vote' method to decide on which label to return. For the classification task, the final prediction will be the one with the most vote; while for the regression task, the average prediction of all the trees is the final prediction.	Regression Classification
AdaBoost	Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome	Regression Classification
Gradient-boosting trees	Gradient-boosting trees is a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it.	Regression Classification

2 Unsupervised learning:

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns).

You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you.

Algorithm	Description	Type
K-means clustering	Puts data into some groups (k) that each contains data with similar characteristics (as determined by the model, not in advance by humans)	Clustering
Gaussian mixture model	A generalization of k-means clustering that provides more flexibility in the size and shape of groups (clusters	Clustering
Hierarchical clustering	Splits clusters along a hierarchical tree to form a classification system. Can be used for Cluster loyalty-card customer	Clustering
Recommender system	Help to define the relevant data for making a recommendation.	Clustering
PCA/T-SNE	Mostly used to decrease the dimensionality of the data. The algorithms reduce the number of features to 3 or 4 vectors with the highest variances.	Dimension Reduction

5.1.1 APPLICATION OF MACHINE LEARNING:

- **Augmentation:** Machine learning, which assists humans with their day-to-day tasks, personally or commercially without having complete control of the output. Such machine learning is used in different ways such as Virtual Assistant, Data analysis, software solutions. The primary user is to reduce errors due to human bias.

- **Automation:** Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots performing the essential process steps in manufacturing plants.
- **Finance Industry:** Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.
- **Government organization:** The government makes use of ML to manage public safety and utilities. Take the example of China with the massive face recognition. The government uses Artificial intelligence to prevent jaywalker.
- **Healthcare industry:** Healthcare was one of the first industry to use machine learning with image detection.
- **Marketing:** Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to estimate the value of a customer. With the boom of data, marketing department relies on AI to optimize the customer relationship and marketing campaign.

Example of application of Machine Learning in Supply Chain:

Machine learning gives terrific results for visual pattern recognition, opening up many potential applications in physical inspection and maintenance across the entire supply chain network.

Unsupervised learning can quickly search for comparable patterns in the diverse dataset. In turn, the machine can perform quality inspection throughout the logistics hub, shipment with damage and wear.For instance, IBM's Watson platform can determine shipping container damage. Watson combines visual and systems-based data to track, report and make recommendations in real-time.

In past year stock manager relies extensively on the primary method to evaluate and forecast the inventory. When combining big data and machine learning, better forecasting techniques have been implemented (an improvement of 20 to 30 % over traditional forecasting tools). In term of sales, it means an increase of 2 to 3 % due to the potential reduction in inventory costs.

Example of Machine Learning Google Car:

For example, everybody knows the Google car. The car is full of lasers on the roof which are telling it where it is regarding the surrounding area. It has radar in the front, which is informing the car of the speed and motion of all the cars around it. It uses all of that data to figure out not only how to drive the car but also to figure out and predict what potential drivers around the car are going to do. What's impressive is that the car is processing almost a gigabyte a second of data.

5.2 ALGORITHMS

5.2.1 RANDOM FOREST

Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithms of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

How the Random Forest works

The following are the basic steps involved in performing the random forest algorithm.

1. Pick N random records from the dataset.
2. Build a decision tree based on these N records.
3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
4. For classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

Advantages of using Random Forest:

Pros of using random forest for classification and regression.

1. The random forest algorithm is not biased, since, there are multiple trees and each tree is trained on a subset of data. Basically, the random forest algorithm relies on

the power of "the crowd"; therefore, the overall biasedness of the algorithm is reduced.

2. This algorithm is very stable. Even if a new data point is introduced in the dataset the overall algorithm is not affected much since new data may impact one tree, but it is very hard for it to impact all the trees.
3. The random forest algorithm works well when you have both categorical and numerical features.
4. The random forest algorithm also works well when data has missing values or it has not been scaled we.

5.2.2 DECISION TREE:

Decision Tree Classification Algorithm:

- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
- In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.
- Below diagram explains the general structure of a decision tree:

Why use Decision Trees?

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

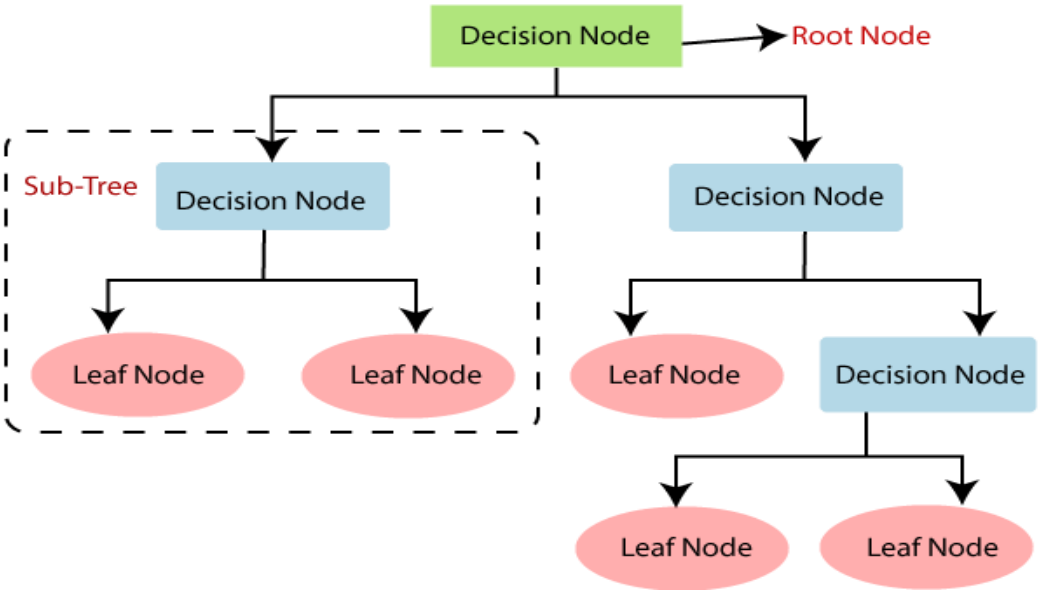

Fig.5.5: Decision Tree.

Decision Tree Terminologies:

- **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.
- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- **Branch/Sub Tree:** A tree formed by splitting the tree.
- **Pruning:** Pruning is the process of removing the unwanted branches from the tree.

- **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

Python Implementation of Decision Tree:

Now we will implement the Decision tree using Python. For this, we will use the dataset "**user_data.csv,**" which we have used in previous classification models. By using the same dataset, we can compare the Decision tree classifier with other classification models such as KNN SVM, Logistic Regression, etc.

Steps will also remain the same, which are given below:

- Data Pre-processing step
- Fitting a Decision-Tree algorithm to the Training set
- Predicting the test result
- Test accuracy of the result (Creation of Confusion matrix)
- Visualizing the test set result.

5.2.3 SUPPORT VECTOR MACHINE ALGORITHM:

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:

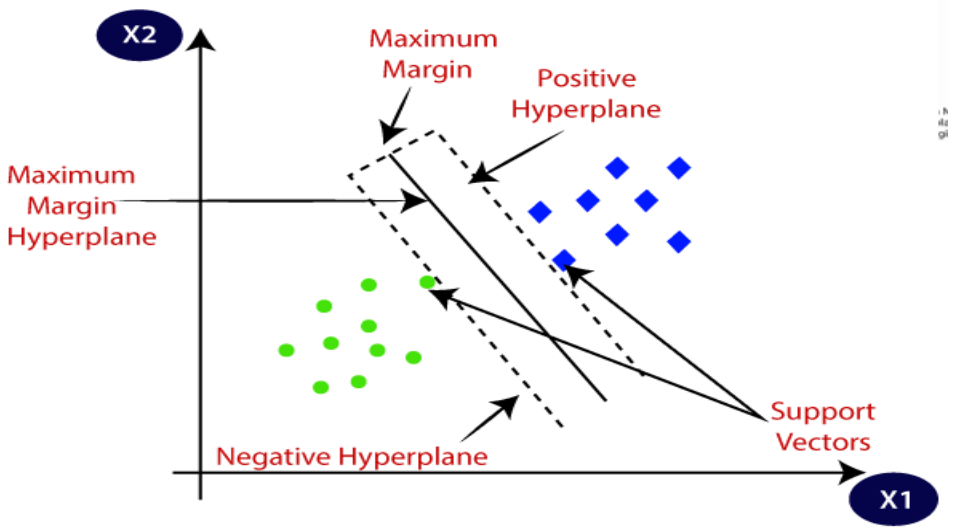

Fig.5.6: SVM.

Types of SVM:

SVM can be of two types:

- **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.
- **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.

Python Implementation of Support Vector Machine:

Now we will implement the SVM algorithm using Python. Here we will use the same dataset **user_data**, which we have used in Logistic regression and KNN classification.

- Data Pre-processing step
- Predicting the test set result
- Creating the confusion matrix
- Visualizing the test set result

5.2.4 NAIVE BAYES CLASSIFIER ALGORITHM:

Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naive Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

Why is it called Naive Bayes?

The Naive Bayes algorithm is comprised of two words Naive and Bayes, Which can be described as:

- **Naive:** It is called Naive because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- **Bayes:** It is called Bayes because it depends on the principle of Bayes' Theorem.

Working of Naive Bayes' Classifier:

Working of Naive Bayes' Classifier can be understood with the help of the below example:

Suppose we have a dataset of weather conditions and corresponding target variable "Play". So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions. So to solve this problem, we need to follow the below steps:

- Convert the given dataset into frequency tables.
- Generate Likelihood table by finding the probabilities of given features.
- Now, use Bayes theorem to calculate the posterior probability.

Python Implementation of the Naive Bayes algorithm:

Now we will implement a Naive Bayes Algorithm using Python. So for this, we will use the "**user_data**" dataset, which we have used in our other classification model. Therefore we can easily compare the Naive Bayes model with the other models.

Steps to implement:

- Data Pre-processing step
- Fitting Naive Bayes to the Training set
- Predicting the test result
- Test accuracy of the result (Creation of Confusion matrix)
- Visualizing the test set result.

CHAPTER-6

RESULT AND CONCLUSION

6.1 RESULT:

Data mining is a process to extract knowledge from existing data. It is used as a tool in banking and finance, in general, to discover useful information from the operational and historical data to enable better decision-making. It is an interdisciplinary field, the confluence of Statistics, Database technology, Information science, Machine learning, and Visualization. It involves steps that include data selection, data integration, data transformation, data mining, pattern evaluation, knowledge presentation.

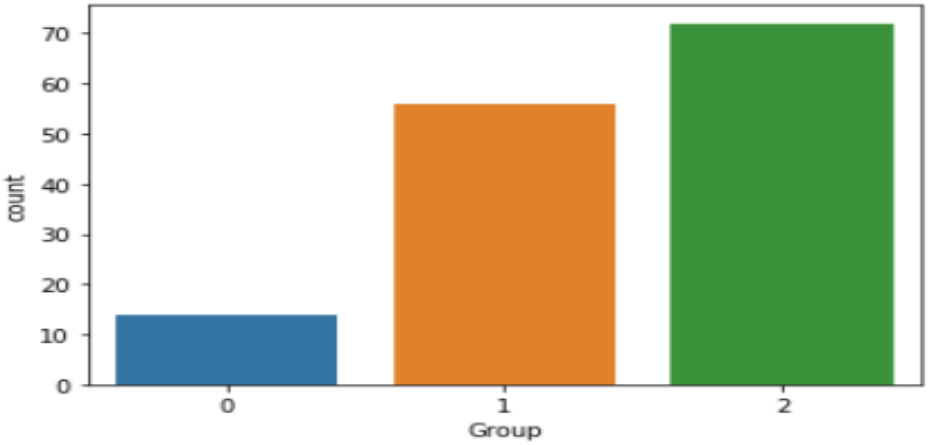

Fig.6.1: Group count graphical output.

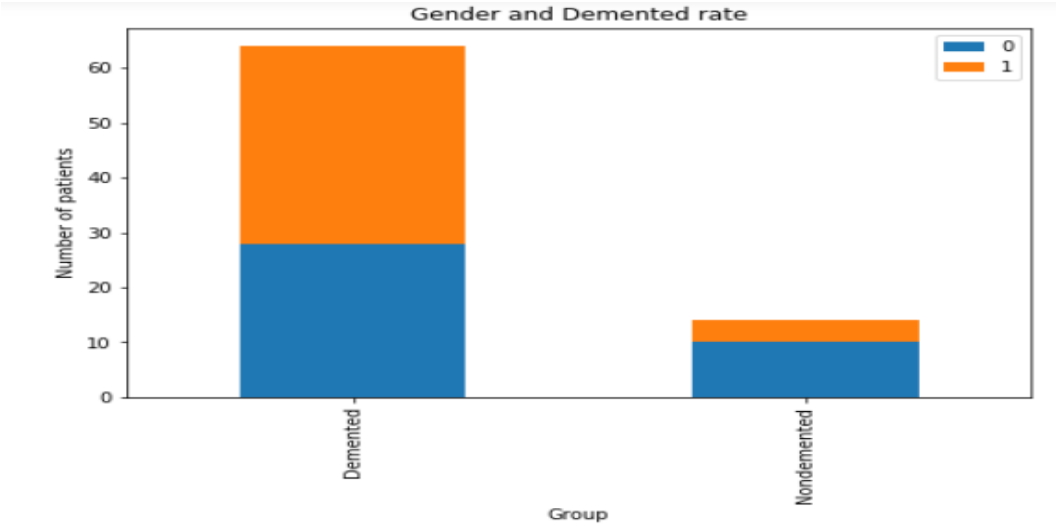

Fig.6.2: Gender and Demented rate.

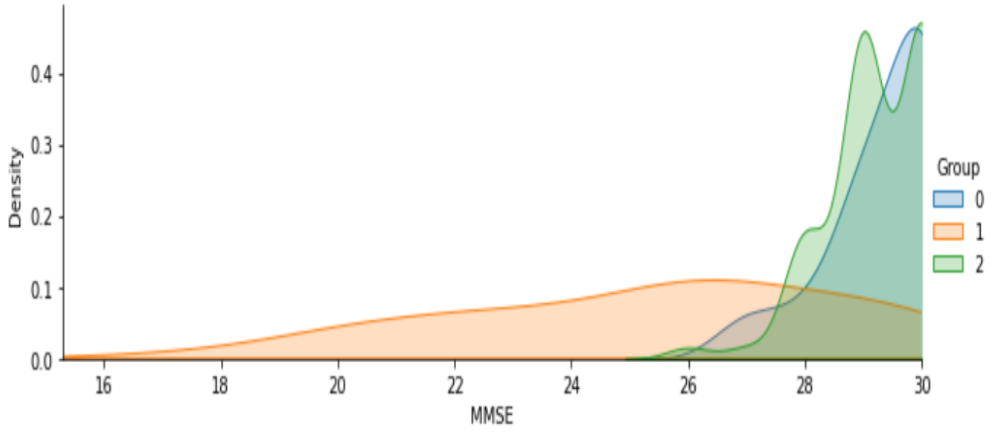

Fig.6.3: MMSE graphical output.

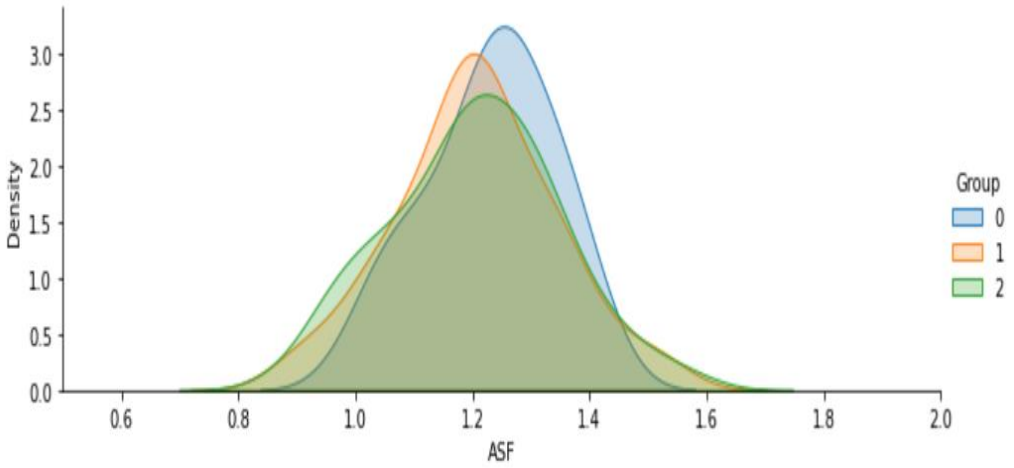

Fig.6.4: ASF graphical output..

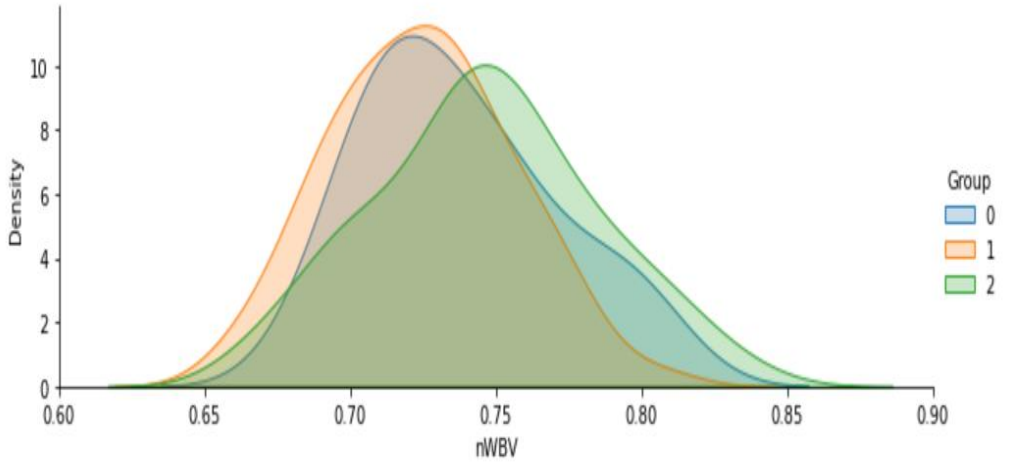

Fig.6.5: nWBV graphical output.

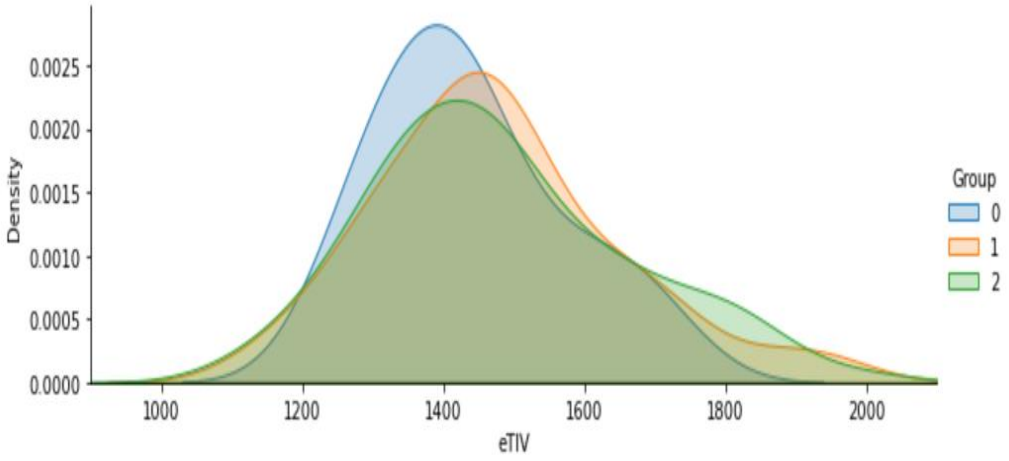

Fig.6.6: eTIV graphical output.

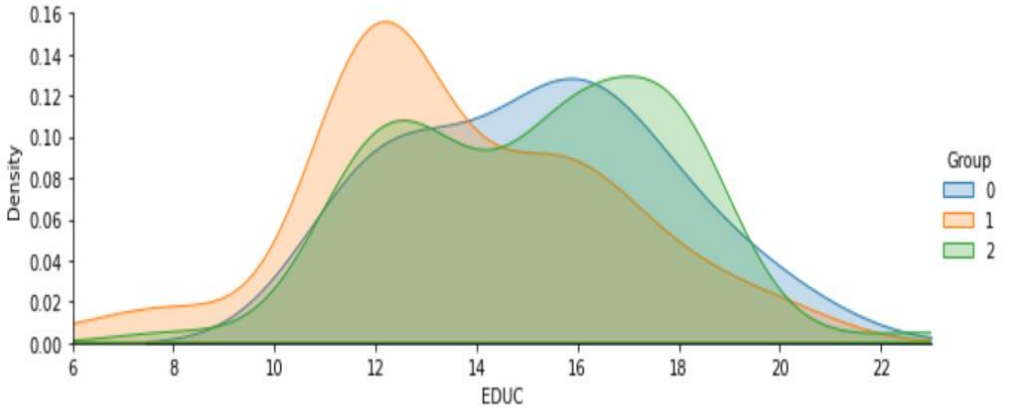

Fig.6.7: EDUC graphical output.

```
Accuracy:
0.46511627906976744

 Classification report of SVM:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         5
           1       0.00      0.00      0.00        18
           2       0.47      1.00      0.63        20

    accuracy                           0.47        43
   macro avg       0.16      0.33      0.21        43
weighted avg       0.22      0.47      0.30        43
```

Fig.6.8: SVM output.

Accuracy:
0.8837209302325582

Classification report of Random Forest:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	5
1	1.00	1.00	1.00	18
2	0.80	1.00	0.89	20
accuracy			0.88	43
macro avg	0.60	0.67	0.63	43
weighted avg	0.79	0.88	0.83	43

Fig.6.9: Random Forest output.

Accuracy:
0.7674418604651163

Classification report of Decision Tree:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	5
1	1.00	1.00	1.00	18
2	0.75	0.75	0.75	20
accuracy			0.77	43
macro avg	0.58	0.58	0.58	43
weighted avg	0.77	0.77	0.77	43

Fig.6.10: Decision Tree output.

Accuracy:
0.7674418604651163

Classification report of Naive Bayes:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	5
1	1.00	1.00	1.00	18
2	0.75	0.75	0.75	20
accuracy			0.77	43
macro avg	0.58	0.58	0.58	43
weighted avg	0.77	0.77	0.77	43

Fig.6.11: Naïve Bayes output.

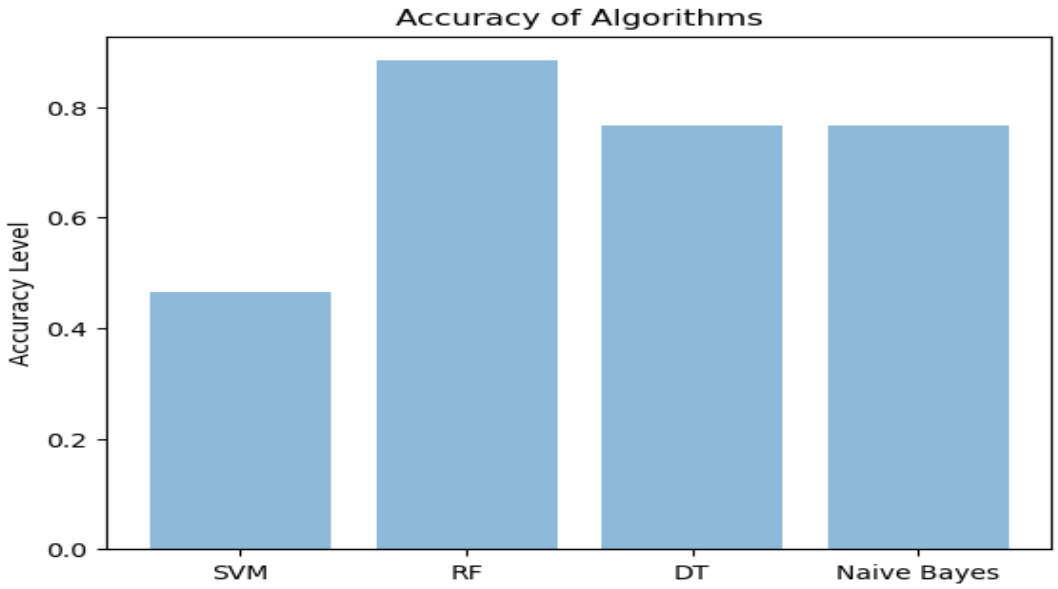

Fig.6.12: Accuracy Level of Algorithms.

Webpage output:

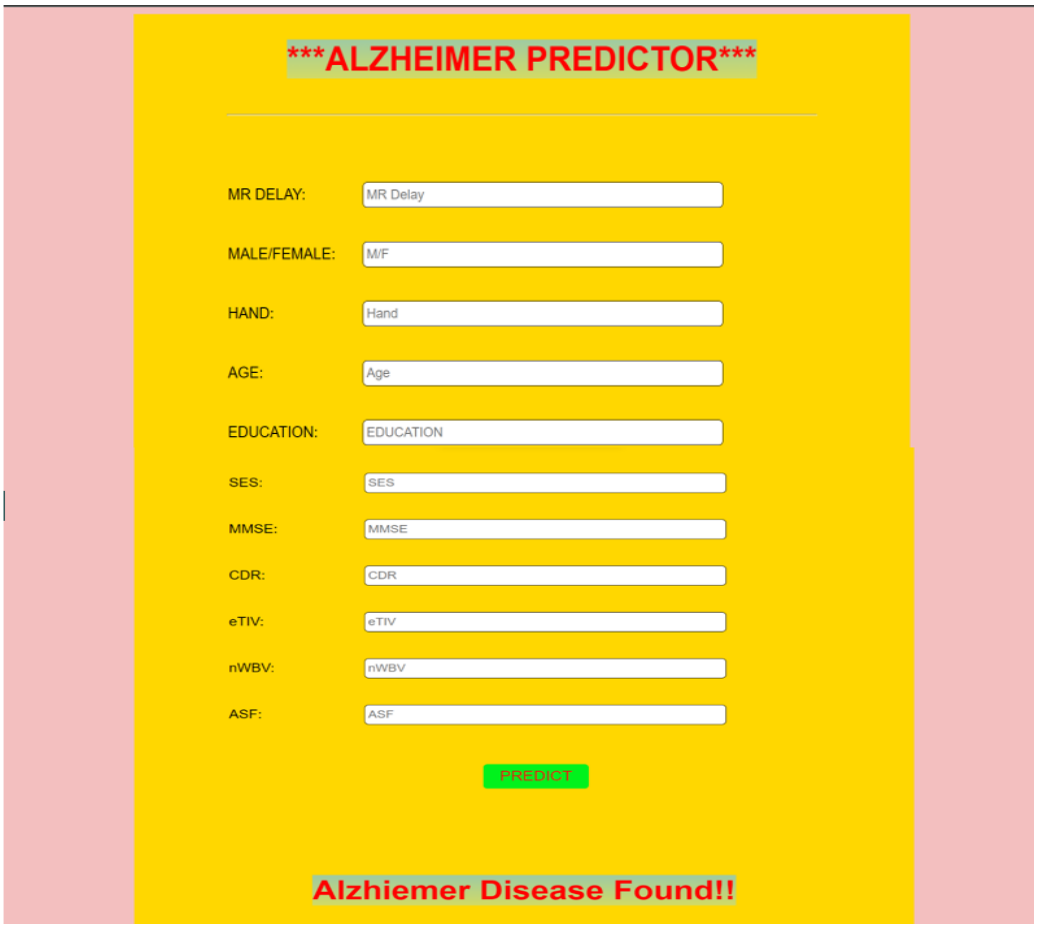

Fig.6.13: Final output of the patient.

6.2 CONCLUSION:

Machine learning approach to predict the Alzheimer disease using machine learning algorithms is successfully implemented and gives greater prediction accuracy results. The model predicts the disease in the patient and also distinguishes between the cognitive impairment.

The future work can be done by combining both brain MRI scans and the psychological parameters to predict the disease with higher accuracy using machine learning algorithms. When they are combined, the disease could be predicted with a higher accuracy in the earlier stage itself.

6.3 FUTURE SCOPE:

Alzheimer's disease (AD) is poised to become the scourge of the next century, bringing with it enormous social and personal costs. Depending on the methods of assessment used, estimates of the prevalence of dementia due to AD in Americans 65 and older range from 6% to 10% (1-3). The prevalence of the disease doubles every 5 years after the age of 60 (4-6). For the population 85 and older, estimates of the prevalence have been as high as 30-47% (1-3). As many as 4 million Americans may suffer from a clinical dementia of the Alzheimer's type, with an annual cost of approximately $100 billion (7). Based on current rates, and in the absence of effective prevention, it is estimated that in 50 years, there will be as many as 14 million cases of clinically diagnosed Alzheimer's disease in the United States alone. While AD is a major public health problem, it also has a very private face that causes tremendous suffering to families. For the elderly, it one of the most dreaded afflictions that threatens to rob them of their independence and dignity at the end of life.

Every person with Alzheimer's experiences the disease differently, but people tend to experience a similar trajectory from the beginning of the illness to its end. The precise number of stages of Alzheimer's is somewhat arbitrary. Some experts use a simple three-phase model (early, moderate and end), while others have found a granular breakdown to be a more useful aid to understanding the progression of the illness.

We used machine learning algorithms along with data pre-processing and principal component analysis (PCA) and feature selection techniques to classify patients with mild

AD and MCI from health volunteers using data obtained from MRI images taken from ADNI datasets. The CNN algorithm has been employed to carry out he prediction. Compared with traditional methods, the proposed method has achieved about 20% of improvement on the classification accuracy, suggesting that neural network is a powerful tool for the diagnosis of neurological diseases. Based on our work, the same or similar methods can be used to diagnose other neurological diseases providing a intelligent healthcare systems. Potential future work includes evaluating our method on larger data sets and applying it to the diagnosis of other neurological diseases.

REFERENCES:

1. [1] M. Prince, A. Wimo, M. Guerchet, A. Gemma-Claire, Y.-T. Wu, and M. Prina, "World Alzheimer Report 2015: The Global Impact of Dementia - An analysis of prevalence, incidence, cost and trends," Alzheimer's Dis. Int., p. 84, 2015.
2. [2] M. Graña et al., "Computer Aided Diagnosis system for Alzheimer Disease using brain Diffusion Tensor Imaging features selected by Pearson's correlation," Neurosci. Lett., vol. 502, no. 3, pp. 225–229, 2011.
3. [3] M. Dyrba et al., "Combining DTI and MRI for the automated detection of Alzheimer's disease using a large European multicenter dataset," in International Workshop on Multimodal Brain Image Analysis, 2012, pp. 18–28.
4. [4] S. Haller et al., "Individual classification of mild cognitive impairment subtypes by support vector machine analysis of white matter DTI," Am. J. Neuroradiol., vol. 34, no. 2, pp. 283–291, 2013.
5. [5] W. Lee, B. Park, and K. Han, "Classification of diffusion tensor images for the early detection of Alzheimer's disease," Comput. Biol. Med., vol. 43, no. 10, pp. 1313–1320, 2013.
6. [6] T. M. Nir et al., "Diffusion weighted imaging-based maximum density path analysis and classification of Alzheimer's disease," Neurobiol. Aging, vol. 36, no. S1, pp. S132–S140, 2015.
7. [7] R. Cuingnet et al., "Automatic classification of patients with Alzheimer's disease from structural MRI: A comparison of ten methods using the ADNI database.," Neuroimage, vol. 56, no. 2, pp. 766–81, 2010.
8. [8] E. Westman, J.-S. Muehlboeck, and A. Simmons, "Combining MRI and CSF measures for classification of Alzheimer's disease and prediction of mild cognitive impairment conversion," Neuroimage, vol. 62, no. 1, pp. 229–238, 2012.
9. [9] C. Aguilar et al., "Different multivariate techniques for automated classification of MRI data in Alzheimer's disease and mild cognitive impairment," Psychiatry Res. - Neuroimaging, vol. 212, no. 2, pp. 89–98, 2013.
10. [10] A. Ortiz, J. M. Górriz, J. Ramírez, F. J. Martínez-Murcia, and A. D. N. Initiative, "LVQ-SVM based CAD tool applied to structural MRI for the diagnosis of the Alzheimer's disease," Pattern Recognit. Lett., vol. 34, no. 14, pp. 1725–1733, 2013.
11. [11] L. Khedher, J. Ramírez, J. M. Górriz, A. Brahim, and F. Segovia, "Early diagnosis of Alzheimer's disease based on partial least squares, principal component analysis and support vector machine using segmented MRI images," Neurocomputing, vol. 151, no. P1, pp. 139–150, 2015.