EARLY DETECTION OF ALZHEIMER'S DISEASE USING MEDICAL DATA

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Upasana Bhasin(161347)
Simran Paul(161352)

Under the supervision of

Dr. Rajinder Sandhu

to

Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Candidate’s Declaration

I hereby declare that the work presented in this report entitled **“Generation of Alzheimer’s Disease Data Using Synthetic Data Generator”** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from January 2020 to June 2020 under the supervision of **Dr. Rajinder Sandhu, Assistant Professor (Senior Grade) Computer Science & Engineering and Information Technology.** The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Simranpaul

UBhasin

Upasana Bhasin, 161347
Simran Paul, 161352

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Rajinder Sandhu

Assistant Professor (Senior Grade)

Computer Science & Engineering and Information Technology

Date:

ACKNOWLEDGEMENT

It is our privilege to express our deep gratitude and regards to our project supervisor Assistant Professor Dr. Rajinder Sandhu for his mentoring, valuable inputs, guidance and constructive criticism throughout the duration of this project. We also express our sincere thanks for encouraging and allowing us to present the project on the topic "**Generation of Alzheimer's Disease Data using Synthetic Data Generator**" for the partial fulfilment of the requirements leading to the award of B.Tech. degree.

We would also like to thank Dr Satya Prakash Ghrera, Head of Department (CSE) for providing us a great opportunity to work on such an interesting project.

Last but not least we would like to express our sincere gratitude to our family members who stood by us and stopped us in every phase of this project and gave us the much required moral support in carrying out this project successfully.

Upasana Bhasin, 161347

Simran Paul, 161352

Table of content:

LIST OF ABBREVIATIONS

EDUC	Years of Education
SES	Socioeconomic Statu
MMSE	Mini Mental State Examination
CDR	Clinical Dementia Rating
eTIV	Estimated Total Intracranial Volume
nWBV	Normalized Whole Brain Volume
ASF	Atlas Scaling Factor

LIST OF FIGURES

ABSTRACT

The increasing aging population and the promises, of disease modifying treatments have indeed made early diagnosis of the Alzheimer’s disease a subject of significant investigation intrigue. Analysis of Alzheimer's disease is frequently missed or deferred in clinical practice; along these lines, strategies to improve early discovery would give chances to early mediation, symptomatic treatment, and improved patient capacity. Access to information is basic for research, execution and training. In any case, acquiring genuine information can be exorbitant and frequently introduces moral difficulties, for example, security concerns. This is especially difficult and challenging in healthcare field, where medical records contain exceptionally sensitive data and are carefully ensured by laws and hierarchical policies. The aim and primary point of this project is generation of synthetic data for early diagnosis of the disease. We aim at developing a model that will indeed help the clinicians to predict the possible chances of dementia and also help in its early diagnosis.

Chapter – 1

INTRODUCTION

Introduction

This report showcases a project which is based on the logics of Machine Learning and their applications in the field of Medical Sciences. The project aims to generate large number of instances, which further can be used for research purposes. The idea of this project came from the scarcity of datasets available online in Medical field basically for Alzheimer's Disease, the symptoms of which are somehow ignored by the practioners due to lack of research. Firstly, a model is proposed by training small set of data with Machine Learning algorithms and labeling is done. This part is necessary as the synthetic data generated have to be verified. The verification can be done using this model. The model's accuracy, precision, f1-score and support are being obtained and are being compared to different algorithms.

Synthetic data

Synthetic data is the data that is manufactured artificially rather than through real world or genuine events. Synthetic data is produced algorithmically, and it is utilized as a sub for test-datasets of generational or operational information, to approve numerical models and to train the machine learning models.

The advantages of using synthetic data include:

- Reducing the constraints associated with using regulated or sensitive data.

- Tailoring the information needs to specific conditions that can't be acquired with legitimate information
- Producing datasets for quality affirmation processes and software testing.

Money related administrations and medicinal services are two enterprises that profit by synthetic data techniques. The methods can be utilized to make information with comparable attributes to the actual regulated and sensitive information. This empowers information experts to utilize and share information more freely. For instance, synthetic data empowers medicinal services data experts to permit open utilization of record-level information yet at the same time keep up patient confidentiality.

Deep learning, a best in class AI approach, has demonstrated exceptional execution over conventional AI in distinguishing unpredictable structures in complex high-dimensional information, particularly in the area of computer vision. The use of deep learning for automated classification and early detection of Alzheimer's sickness (AD) has lately gained impressive consideration, as fast progress in neuroimaging methods has produced enormous scale multimodal neuroimaging information.

Alzheimer’s disease

Alzheimer's disease is basically an irreversible, dynamic mind issue that gradually wrecks memory and thinking aptitudes and, in the end, the capacity to do the very simple tasks. In the vast majority of people with this disease—those with the late-beginning sort—indications initially show up in their mid-60s. Early-beginning Alzheimer's happens between an individual's 30s and mid-60s and is exceptionally uncommon. Alzheimer's illness is the most well-known reason for dementia among more established adults.

The malady got its name Dr. Alois Alzheimer. He saw changes in the brain tissue of a 51 year old woman who had passed on of an uncommon psychological and language and memory sickness. The indications of the lady included language issues, memory misfortune, and erratic conduct. These symptoms coordinated the meaning of what was called dementia then.

This harm at first happens in the area of cerebrum fundamental in shaping recollections also known as hippocampus. As the neurons pass on, more parts of the cerebrum are affected. No later than the last phase of Alzheimer's, the harm is boundless, moreover cerebrum tissue has contracted completely.

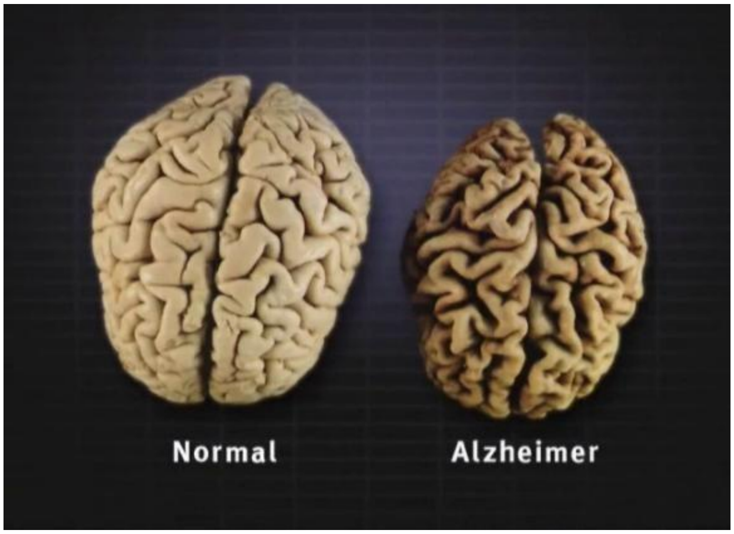


Fig. 1.1 Normal Brain vs. Alzheimer’s Brain

Early Diagnosis of Alzheimer’s disease is important because:

- Symptoms may be reversible

- The Alzheimer’s disease may be treatable at its early stages
- The diagnosis are often more accurate in the early disease process

Problem Statement

Alzheimer’s disease is a neurogenerative issue of cause that is uncertain and pathogenesis that fundamentally influences adults that are older and is the most well-known reason for dementia. The most punctual clinical appearance of AD is particular memory impedance and keeping in mind that medicines are accessible to improve a few indications, there is no fix available at present Brain imaging by means of magnetic resonance imaging that is MRI, is utilized for assessment of patients suspected with AD. MRI discoveries include both generalized and local shrinkage of cerebrum tissue. A few examinations have recommended that MRI highlights may anticipate pace of decay of AD and may direct treatment later on. Anyway so as to arrive at that stage clinicians and scientists should utilize AI systems that can precisely foresee progress of a patient from mild cognitive impairment (mellow intellectual debilitation) to dementia. We propose to build up a sound model that can assist clinicians with doing that and anticipate early Alzheimer's. Also, we will make use of synthetic data because of the restrictions that are placed on the use of real data and also the unavailability of real data.

Objectives

This project proposes to build up a sound model that can assist clinicians with the early symptoms of Alzheimer’s Disease using the synthetic verified data. The research work being done in this field was not upto the extent due to some privacy reasons that is some patients don’t want their medical information to be revealed to world. Another reason was the lack of availability of early detection data due to which the diagnosis was not up to the mark.

So the sole purpose of this project is to generate the data to overcome the above mentioned problems in future.

Methodology

After going through several research papers, inferences can be made that Medical data in the field of Alzheimer’s disease is extremely difficult to get due to several reasons as mentioned above. The initial step towards the success of this project contains pre-processing of the data. So the data is analyzed and the features with maximum impact are selected and rest are dropped for better results. The features with non-numeric values were converted to binary digits for better understanding and also for easy implementation. The missing values in the dataset were exchanged with the mean value of that feature and NaN values were removed. Plotting of the features was done against the number of patients using bar graph and grid. This was done to find the impact of features [Age, MMSE, EDUC, ASF etc] on the labeling: Dementia and Non-Dementia.

Dataset Description:

- Longitudinal MRI data is being used.
- The dataset comprises of a longitudinal MRI information of 375 subjects matured 60 to 96.
- Every subject was scanned no less than once.
- Everybody is right handed.

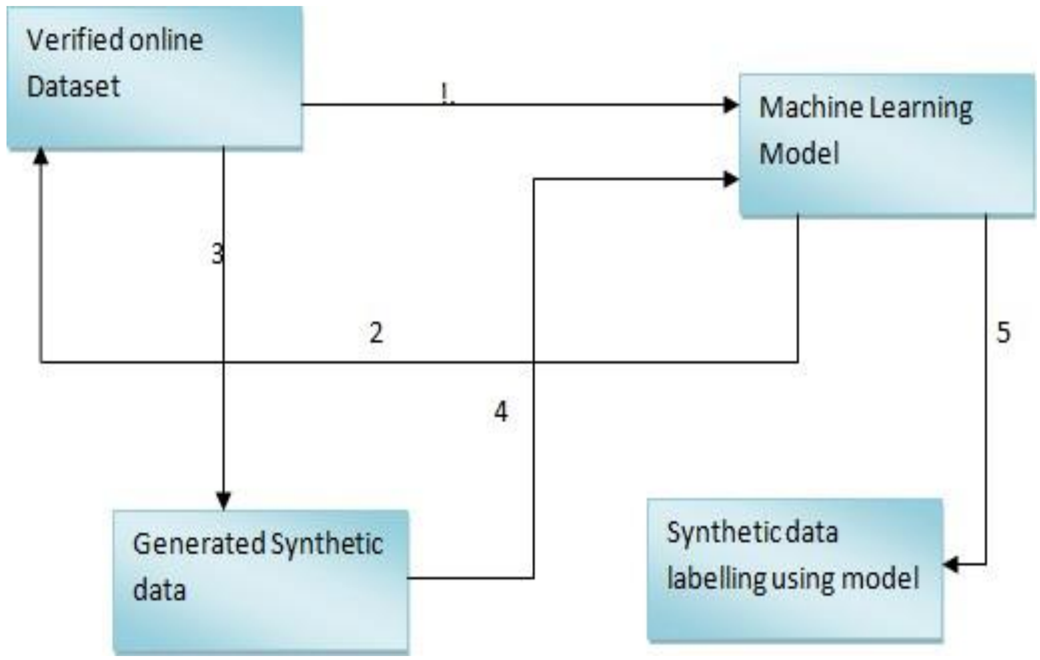


Figure 1.2: Overview of the project

Label 1: The model is trained on the dataset which is collected from online source

Label 2: After the training of model is done, the model is being verified by checking the values from the dataset and new labeling is being predicted.

Label 3: Through some permutation, combinations and using the concepts of probability a raw data set will be generated of a large number of instances.

Label 4: The dataset which is being generated synthetically is tested in the proposed model to have a check on the dataset.

Label 5: The trained model will generate labeling for the synthetic data which we have made using the actual data.

Organization

Python is a high level programming language which is used generally by the programmers. Python can be used for developing desktop GUI applications using Tkinter, websites and applications of web. Also, as a high level programming language, it allows you to focus on deeper and inner functionality of the application by taking proper care of common programming tasks. This programming language is easy because the code you write is readable and easily accessible. Also, python is an object oriented programming language (OOPS). If you know the basics of C and a little bit about data structures and algorithm, you can be the master of basics of Python quite comfortably and easily.

Jupyter Notebook:

“The well known Jupyter Notebook is an open-source application which runs on web that allows you to create articles and share documents that contains live codes, equations, explanatory texts and visualizations. The jupyter notebook has a plenty of uses and some of which are: data cleaning and transformation, numerical simulation, statistical modelling, performing machine learning models and algorithms and much more. The Jupyter Notebook is an extremely powerful tool for inter-actively developing and then presenting data science projects. Firstly, what is a notebook? It is called a notebook because it integrates the code and its output into a single document that combines the visualizations, mathematical equations, narrative text, and other rich defined media.

Chapter – 2

LITERATURE SURVEY

This chapter discusses a survey and a study of insightful sources, (for example journal articles, theses and books) on the topic. It displays an overview of developments, key findings and concepts in relation to the topic.

In 2015, Moradi et al. [1] proposed a MRI based strategy for the identification of Alzheimer's disease transformation in the MCI patients by utilizing propelled ML calculations along with consolidating information with neuro-psychological outcomes. 2 processing steps were performed between the voxel-based morphometry style pre-processing and the learning of LDS classifier. To start with, age-related impacts from MRI information were expelled prior to preparing the classifier to counteract the jumbling between Alzheimer's disease and age related impacts to cerebrum life structures. Already, a comparable method had been utilized for the characterization among NC and AD subjects, however this examination did not consider AD transformation expectation in MCI subjects Furthermore, the effect of age was considered as of late for recognizing AD and also for predicting Alzheimer's disease in MCI patients. Second, feature selection was performed on MRI information autonomously of the classification technique utilizing the auxiliary information from NC and AD subjects. Selection of features is a fundamental part of the consolidated method because the quantity of features (29,852) accessible following the picture preprocessing fundamentally surpasses subject numbers. This thought was implemented by applying RLR that is regularized logistic regression on MRI information of NC as well as AD items for searching the picture voxels bestt separated among the items. Then the chosen voxels were utilized for foreseeing transformation to Alzheimer's in the patients with MCi..

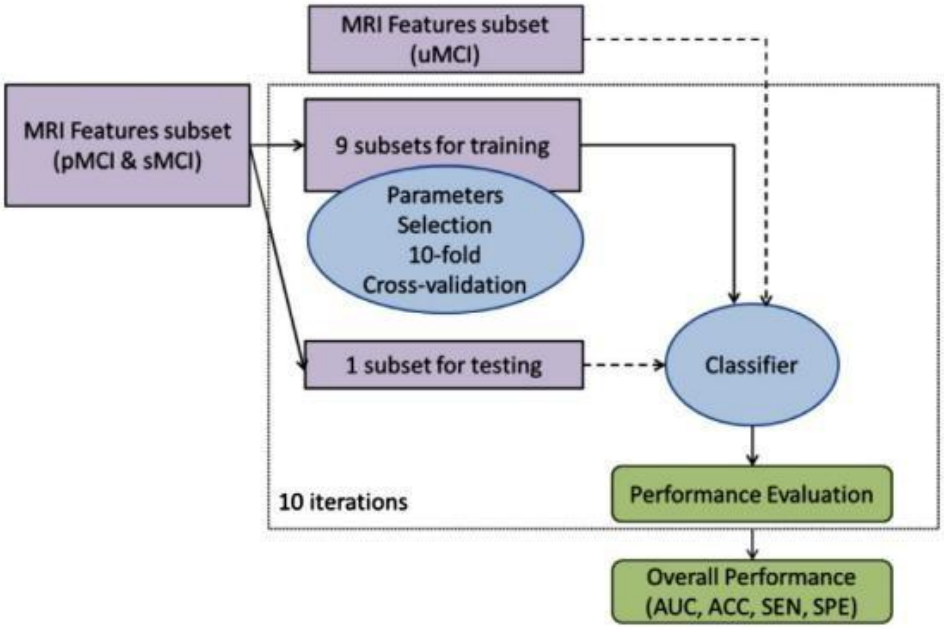


Figure 2.1 Semi Supervised Selection schemes [1]

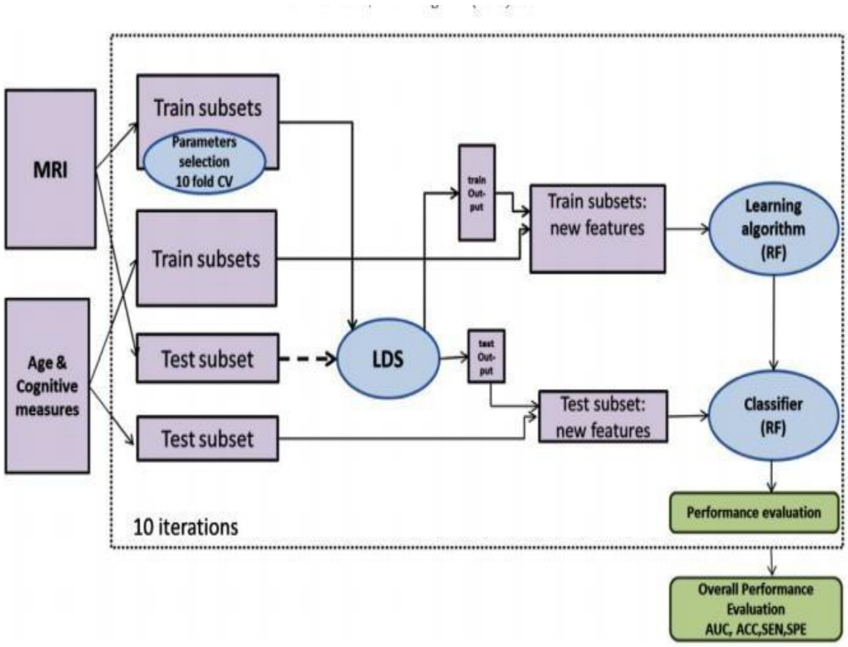


Figure 2.2: Workflow for an Aggregate Biomarker [1]

Already, the blend of MRI determined data and subjective estimations had been taken in account by Ye et al. (2012), prepared a RLR classifier using the psychological estimations and specific region volumes of interests taken as feature also Casanova et al. (2013), consolidated yields of 2 classifiers, 1 prepared which was MRI dependent and the second one was prepared subjective estimations dependent. So as to utilize everything more effectively MRI and essential (cognitive and age) measures, the 2 different classifiers were utilized to create an aggregate biomarker, that is random forest (RF) and LDS, in various phases of the procedure.

In 2015, Zhang et al. [2] proposed a method proposed a method that utilized ICV that chose cuts from the 3-Dimensional volume information. Secondly, they created an eigenbrain group for each and every subject. Thirdly, the MIE was obtained by WTT. At last, utilization of vector mahines with kernel support comprising of a number of kernels was made for precise format of the disease items. MIE coefficients having value greater than or equal to 0.98 quantile were featured to acquire the discriminant locales that recognize Alzheimer's disease from NC.

The examinations demonstrated that the strategy proposed can diagnose Alzheimer's disease subjects with an aggressive execution with existing techniques. The eigenbrain build Computer aided design framework distinguished 30 AD related cerebrum regions.

It was concluded that the eigenbrain technique was viable in Alzheimer's disease subject prediction of subject and also discriminant cerebrum part identification in the MRI examining.

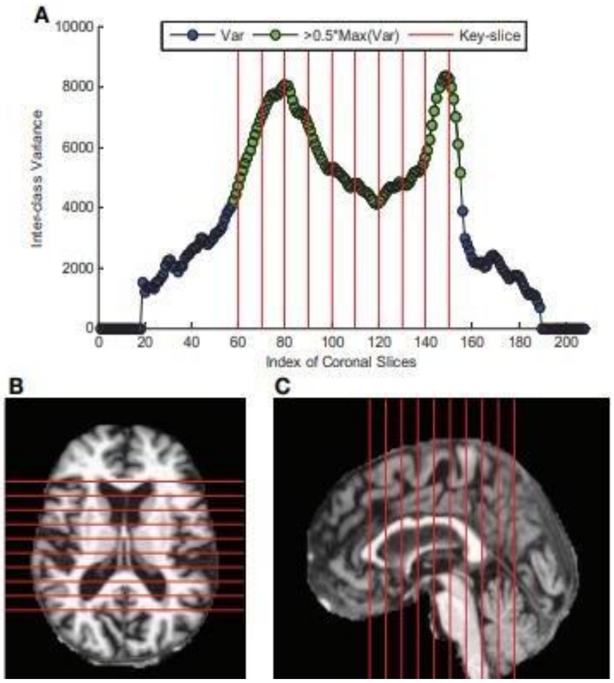


Figure 2.3 Key Slice Selection (with red lines corresponding to key lines) [2]

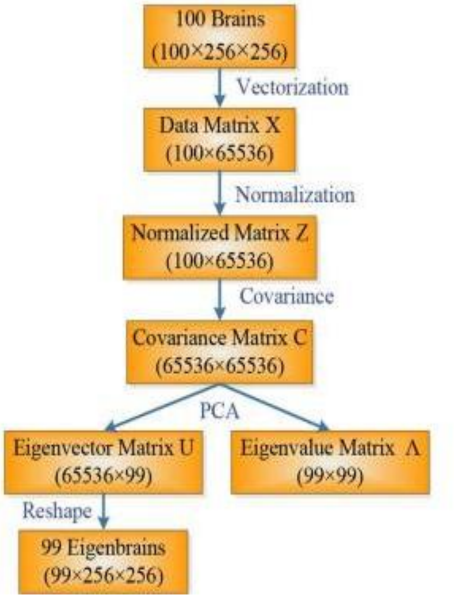


Figure 2.4 Flowchart for calculating the eigenbrain [2]

In 2015, Paul and Davatzikos [3] proposed an image build classifier that they used for classifying of cerebrum MRI checks of people having MCI. The large dimensionality of the picture information was decreased utilizing nonlinear complex learning methods, in this way giving a very low dimension embed. Highlights of embed were utilized in co-occurrence with semi-supervised classifier, using unlabeled and labeled pictures for boosting and also supporting the performance. The technique was used with 237 sweeps of patients so as to diagnose the MCI to AD. Exploratory outcomes exhibited superior prediction accuracy contrasted with a best in class strategy.

VBM is regularly used for estimating the collection contrasts, therefore was not successful in arranging people. So as to get over the constraints, high dimension classifying grouping techniques were put forward to the previously ongoing writings. The high dimension pattern characterization techniques study similarities in numerous mind regions. The blend of estimations from a wide range of areas can conceivably manufacture designs of higher discriminative power, because not even a single mind region had adequate affectability and explicitness due to inter subject inconstancy. But, the strategies had two central restrictions concerning restorative imaging applications. To start with, the dimensionality of restorative pictures is highly comparative with the constrained selection. Secondly, marks for restorative pictures were frequently accessible to a limited extent. Accordingly, traditional supervised methodologies may neglect to find classifications at better granularity levels and also reflect the hidden information conveyance. This paper addressed the high dimensionality problem through non-linear dimensionality decrease pointing picture information and uncertain name problem through semi-supervised learning. Reduction in dimensionality was performed through complex learning. Complex learning strategies install high-dimension information to a much low-dimension area. The subsequent inserting directions could be utilized as low dimension highlights to contrast informational indexes. The paper also addressed the second issue of uncertain marking through semi-supervised learning scheme. Semi-supervised retaining is used to train the classifier onto unlabeled as well as labeled information.

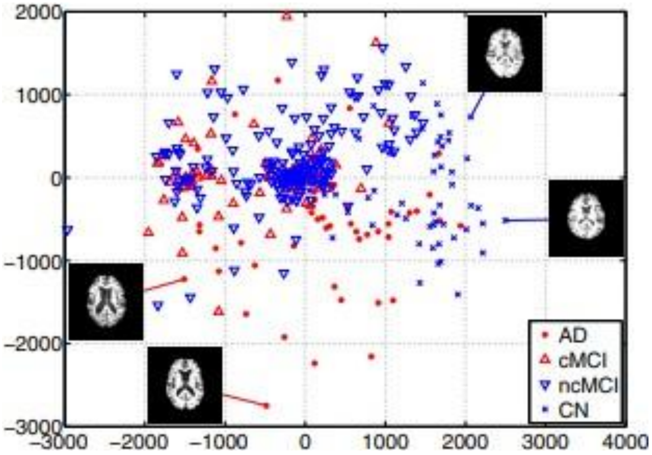


Figure 2.5 Two dimensional embedding of each and every subject [3]

In 2010, Marcus et al. [4], proposed the longitudinal MRI information of the older adults having or not having Alzheimer’s disease. The informational index incorporated longitudinal MRI information from 150 people aged 60-96 yrs, incorporating Sixty-Four people having mild to moderate Alzheimer's disease analyzed through clinical means moreover portrayed utilizing the CDR scale during the first visiting. Another Fourteen people got portrayed as non-demented initially in the first hour afterward clinical resolved of having AD after hour of consequent test. The information was procured using a similar scanner utilizing similar methodology. People were tested to remove the ones having mental and neurological conditions that may add to dementia. In this manner, a significant number of the more established grown-ups had age connected increments in BP what's more, a little part of diabetes. Test qualities were comparable in the people having and without Alzheimer's.

Longitudinal cerebrum imaging had been demonstrated valuable in considering ordinary and infected aging. The volume of brain decay estimated by longitudinally obtained MRI. The volumes of the entire mind what's more, structures related with memory had been appeared to decay at a fundamentally more prominent rate in early AD and MCI.

Longitudinal investigation of measures got from progressed computational strategies, including nonlinear miss happenings and shape examination, had likewise uncovered age and malady related changes in the cerebrum. Longitudinal proportions of cerebrum structure were developing as apparatuses for following movement of ailment and as subordinate result measures in clinical preliminaries. The OASIS longitudinal information set was being made transparently accessible to support proceeded examination concerning maturing and malady forms and to help advancement of improved techniques for studying these forms.

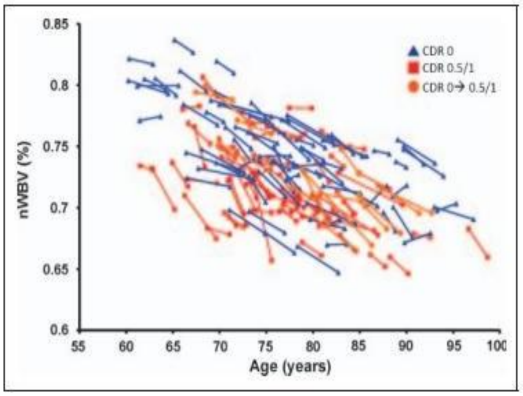


Figure 2.6 Longitudinal Plot of nWBV demonstrated [4]

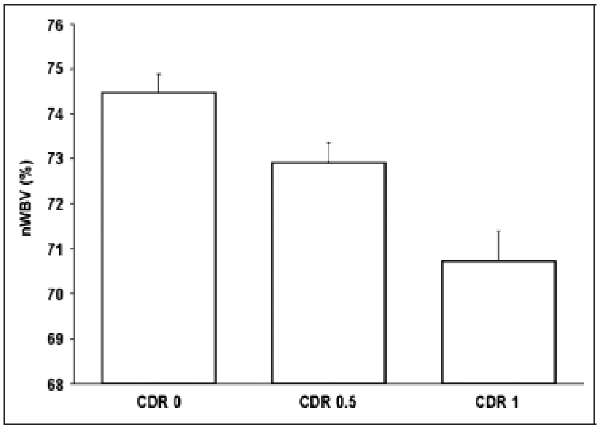


Figure 2.7 Mean cross sectional nWBV for each individual separated by CDR- 0.5 to 1 during their initial visit [4]

In 2008,Magnin et al.[5] proposed to evaluate and present a new computerized strategy in view of SVM that is SVM classifying of entire mind MRI to compare the people having AD and the older people. The method used contemplated 16 patients with Alzheimer's disease and finally 22 older controls. 3D t1 MRI pictures of each person was parcellated into interest areas.

The results demonstrated 94.5% mean rightness classification for control subjects and Alzheimer's disease (mean particularity, 96.6%; mean affectability, 91.5%). Lastly it was concluded that the technique was likely to recognize subjects with Alzheimer’s than the old techniques and in this way it will be successful in early diagnosis of Alzheimer’s Disease.

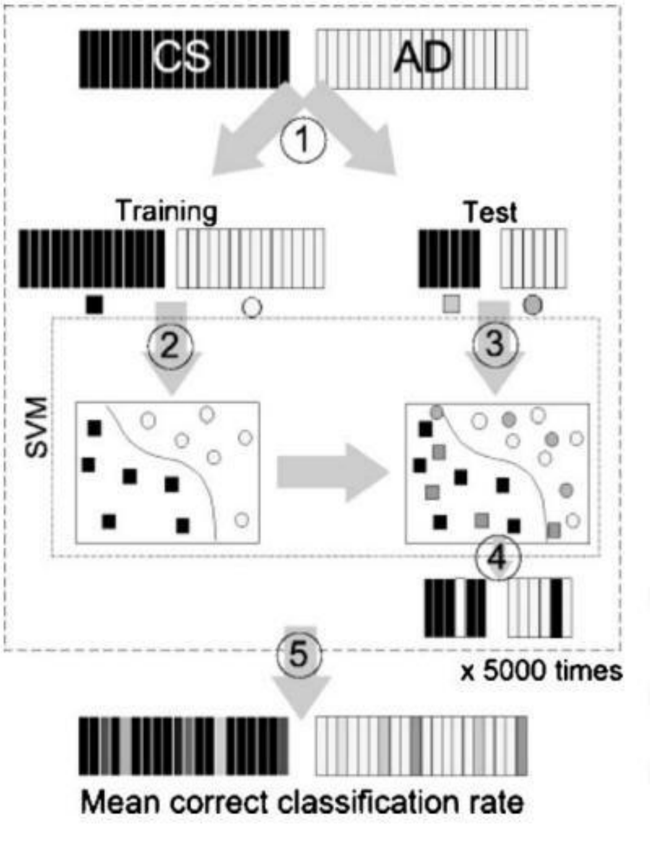


Figure 2.8 Method for the data selection [5]

In 2007, Marcus et al. [6], proposed a model in which chose subjects matured 18 -96 from a large database of people. This was in view of accessibility of not less than three obtained righthand strength, T1 weighted pictures, and a then ongoing clinical assessment for the elderly people.

Various information sharing methodology had been embraced to guarantee the integrity furthermore, the convenience of using the data provided:

a) Quality control: The pictures were deliberately tested for obtaining and handling mistakes. Pictures having extreme relics were barred from the conveyance. Pictures with noticeable antiques regular of MR acquisitions were incorporated. This changeability in quality was proposed to give tool builders with practical objective pictures.
b) Documentation: Point by point obtaining conventions and parameters, socioeconomics, and supporting measures (for instance [MMSE] scores) were accommodated for every single picture.
c) Planning: Additionally to the raw acquisition information, pictures representing basic post processing methodology(e.g bias field correction, atlas registration) were dispersed as a benefit and for correlation with comparable post processing techniques.
d) Anonymization: In consistence with privacy guidelines, distributed information were allocated identification numbers that were random (ID's), and all identifying data were expelled from the picture documents. All connections among original and arbitrary IDs were destroyed permanently before release.
e) Access: The information were accessible on DVD and available for review and downloading by means of a committed Website. Outsiders were urged to readjust the information in manners that cause in general accessibility.

f) Continuous help: The OASIS Web webpage filled in as a gathering for declarations and dialogs in regards to the circulations and as a clearinghouse for outsider inferred informational indexes.

People with main source as dementia other than Alzheimer disease (example vascular dementia), mental sickness (example depression), severe skull damage, past of important stroke, what's more, utilizing psycho active medications have been prohibited. Be that as it may, people with age related cerebrum change(example leleukoaraiosis) have been acknowledged. MRI acquisitions normally have been acquired inside a time of a subject's clinical evaluation (mean equals 105 days, run equals 0–314 days). 3 subjects with Alzheimer's disease were checked after a long time (635, 449 & 443 days). Bits of the clinical, statistic, and picture information acquired from subjects in this discharge had been utilized in past examinations.

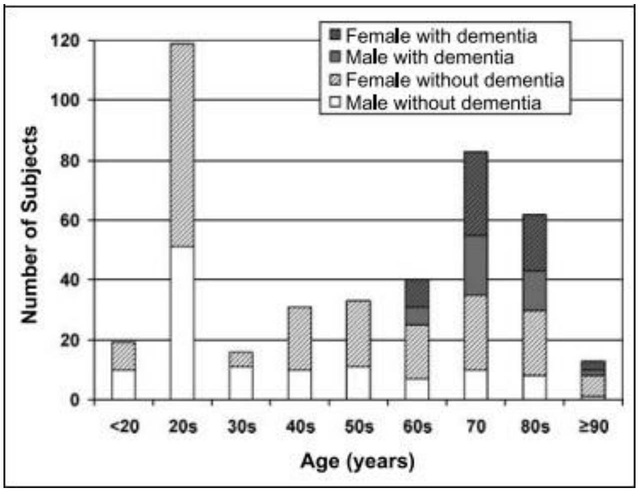


Figure 2.9 Subjects by age, sex and dementia status [6]

The following work was found in literature and has been used to the research work involved in this project:

a) ML structure for the prior MRI based Alzheimer's transformation forecast in Mild Cognitive Impairment patients. [3]:

In this paper the creators were keen on recognizing MCI as a intermediate step between Alzheimer's and age based coginitive decay. The gathering proposes a novel MRI based biomaker that they created utilizing machine learning methods.. The paper asserts that their total biomarker accomplished a 10-overlap cross-approval AUC(area under curve) value of 0.9120 in separating between continuous and non continuous MCI..

Noteworthy Techniques:

- Semi-regulated learning on information accessible from AD patients and typical controls, without utilizing MCI patients to help with sMCI/pMCI characterization. Regularized logistic regression for performed feature selection
- They expelled aging impacts from MRI information before classifier preparing to counteract possible bewildering between changes because of AD and those because of ordinary aging.
- At last developed a total biomarker by first studyinga different MRI, then afterward consolidating age as well as psychological features about mild cognitive impairment patients on applying an irregular forecast classifier.

b) Diagnosis of the patients and cerebrum locales identified with Alzheimer's malady utilizing 3 Dimensional MRI outputs dependent on machine learning and eigenbrain. [4]

The creators of this paper have put forward a novel computer aided diagnosis framework for MRI pictures of minds dependent on eigenbrains

c) Bolster classifying from entire mind anatomical MRI.

In the paper the creators propose another strategy to separate patients with AD from old controls based on SVM(Support Vector Machine) arrangement of entire mind anatomical MRI. The creators utilized three dimensional T1 weighted MRI pictures of 16 patients suffering with Alzheimer's Disease and 22 older controls and divided them into ROIs(Region Of Interests). They at that point utilized a SVM calculation to order subjects based on the gray matter attributes of these ROIs. In light of their outcomes the classifier got 94.5% mean rightness. The potential defeats of their system may be the way that they haven't considered age related changes in the gray matter and they were working with a little informational index.

Chapter – 3

SYSTEM DEVELOPMENT

Libraries used for system approach

a) Numpy**:**

It is a general purpose package used for array accessing. It provides us with a high-performance multi-dimensional array object and various tools for working with these arrays. It is the fundamental and necessary package for scientific computation with Python. Along with its obvious scientific uses, Numpy is also used as an efficient multidimensional container of generic information.

Numpy array is a table of elements mainly numbers, all of the same type, indexed by a tuple of non-negative integers. In Numpy, rank of the array refers to the number of dimensions. A shape of array is referred as the tuple of integers which give the size of the array along the dimensions. An array class in this library is called as **nd-array**. Elements are accessed by using square brackets and can be initialized by using nested Python Lists in Numpy arrays.

Creation of Numpy Array:

There are multiple ways to create an array in Numpy, with various number of ranks, defining the size of the Array. Arrays can also be created with the use of various different data types such as lists, tuples, etc. The type of the resultant array is reduced from the type of the data elements in the sequences.

Note: Type of array can also be defined while creating the array.

```
import numpy as np
ar = np.array([1, 20, 35])
print("Array with rank 1: \n",ar)
```

```
ar = np.array([[1, 20, 35], [4, 55, 64]])
print("Array with rank 2: \n", ar)
ar = np.array((1, 30, 22))
print("\nArray created using "
"passed tuple:\n", ar)
```

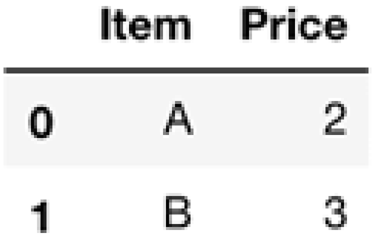
b) Pandas:

This is an open-source library that allows you to perform data manipulation in Python. Pandas library is built with Numpy on the top, meaning Numpy is necessary for Pandas to operate. Pandas provide an easy way to manipulate, create and wrangle the data. This library is also an elegant solution for time series data.

Due to the following advantages its is very commonly used by Data Scientists:

- Missing data is easily handled.
- Data is sliced in very efficient and proper way.
- Merging, concatenation and reshaping is done very swiftly and is very flexible.
- A strong time-series device is also available to work with.

A DataFrame is a 2-dimensional array; with labelled axes i.e. columns and rows. A DataFrame is a way of storing data. Data Frame is represented by:



	Item	Price
0	A	2
1	B	3

Figure 3.1: Data Frame Example

A series is an 1D data structure. A series may have many types such as integer, float, and string. According to its meaning a series cannot have multiple columns. These are some of the common functions which can be performed:

- read_csv : import data
- Series: create series
- DataFrame: create data frame
- Data_range: create data range
- head(): return head
- tail(): return tail
- describe(): describes the data set

c) Matplotlib:

This is a wonderful visualization library in Python for 2-dimensional plots of arrays. Matplotlib is a multiple platform information visualization library on Numpy arrays and designed to work with the large and broad Scipy stack. John Hunter introduced it in the year 2002.

One of the greatest benefits of visualization through matlplotlib is that it allows us visual access to enormous amounts of data in easily digestible visuals. Line, bar, scatter, histogram plots are all different elements of matlplotlib. This process alike to lot of computer graphics processes. First, you have some data, then you draw data to a canvas, but it is only in the computer's memory. Once you've drawn that data, you can then show that data which is called visualization. This is so the computer can draw everything easily, and then perform the more difficult task of showing it on the screen.

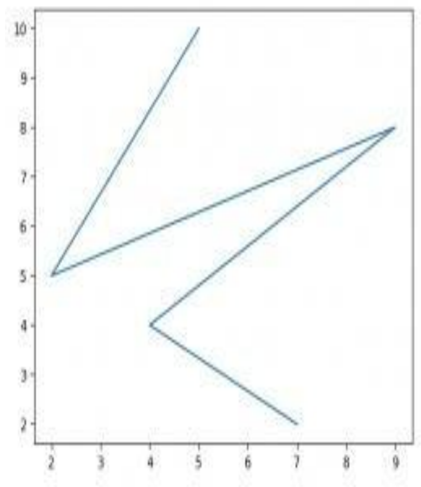


Figure 3.2: Line Graph

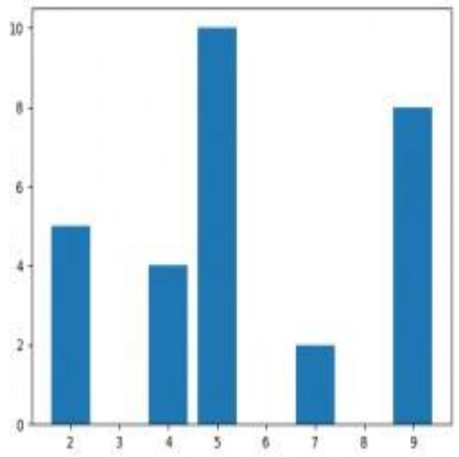


Figure 3.3: Bar Graph

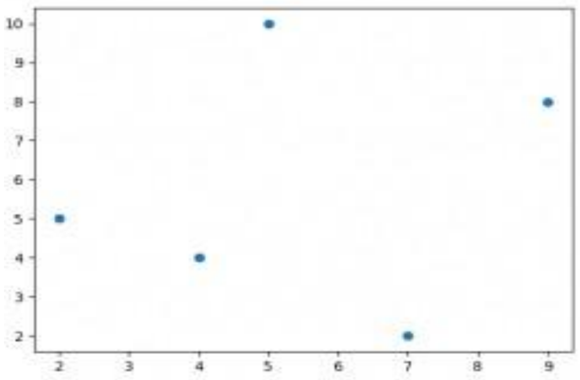


Figure 3.4: Scatter Plot

d) Seaborn:

It is basically a Python information visualization library dependent on matplotlib. It gives a high level interface to drawing appealing and instructive statistical designs.

Seaborn is basically a library for generating statistical designs in Python. It is constructed over matplotlib and firmly incorporated with pandas information structures.

Here is a portion of the usefulness that seaborn offers:

Firstly, a data set oriented API for looking at connections between various factors. Secondly, specific help for utilizing all categorical factors to show aggregate statistics or observations. Also, alternatives for picturing bivariate or univariate distributions and for looking at them between subsets of information. Plotting the linear regression models and automatic estimation for various types subordinate factors. Convenient views on the general structure of complex datasets. Significant level deliberations for organizing multi-plot frameworks that let us effectively construct complex perceptions. Concise control over the matplotlib figure styling with a few inherent subjects. Devices for picking shading palettes that dependably uncover designs in the information

Seaborn makes perception a central point to investigate and get the data. Its data set oriented plot capabilities works on arrays and data frames that contain the whole data sets and internally performs the important statistical aggregation and mappings to create instructive plottings.

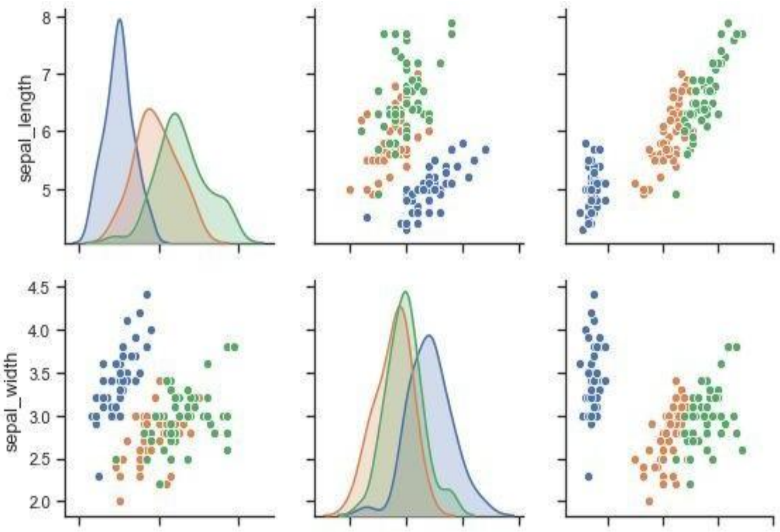


Figure 3.5: Seaborn Example

Measurement Tools

a) Accuracy:

Accuracy is proportion of no. of right predicts to the total number of the predictions made .

Accuracy=

Accuracy functions fine just if there are equivalent no. of tests belonging to each classes

1 .

For instance, think about that there are 98% examples of class X and 2% tests of class Y in our training set. At that point the model can without much of a stretch get 98% training precision by just predicting each training test belonging in Class X.

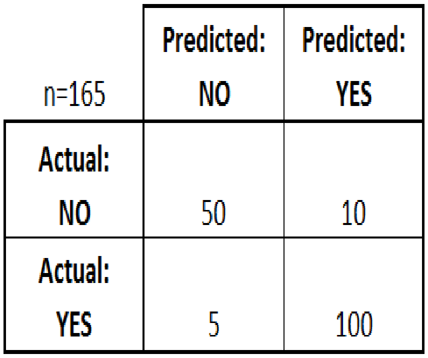
At the point when a similar model is tried on a test-set with 60% examples of class X and 40% examples of class Y, at that point the test precision would drop to 60%.

Characterization Accuracy is extraordinary, however gives us the misguided feeling of accomplishing high exactness.

The genuine issue emerges, when the expense of misclassification of the minor class tests are high. On the off chance that we manage an uncommon however deadly ailment, the expense of failing to analyze the ailment of a debilitated individual is a lot higher than the expense of sending a healthy individual to more tests.

b) Confusion Matrix:

This gives matrix as the output and also describes the full performance of the given model. Let us consider a binary classification issue. Also, we have a few samples belonging to 2 classes i.e. NO or YES. We have a classifier which is used to predict a class for the following input sample. We get the below result on testing the model on around 165 samples.



n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

Figure 3.6: Confusion Matrix

There are 4 significant words:

- True Positives:-It refers to the instances in which the prediction was YES which was also the final yield..

- True Negatives:-It refers to the instances in which the prediction was NO which was also the final yield..

- False Positives:-It refers to the instances in which the prediction was YES while the final yield was NO.

- False Negatives:-It refers to the instances in which the prediction was NO while the final yield was YES. .

Accuracy of matrix can be calculated using the following:

$$\text{Accuracy}= \frac{\text{True positive false negative}}{\text{Total number of samples}}$$

$$\text{Accuracy} = \frac{}{} = 0.91$$

Confusion Matrix is extremely important i.e. it creates the foundation for other forms of metrics.

c) F1 Score

F1 Score calculates the accuracy of the test.

[0, 1] is the scope of F1 score. It basically gives information about the exactness of a classifier, and also its robustness (i.e. it includes the significant no. of cases).

Higher exactness yet low recall points to an amazingly precise, however it at that point misses an enormous no. of occurrences that are hard for classifying. The more prominent the F1 Score, the exhibition of the model is superior. Numerically, it very well may be communicated as:

$$F1 = 2 * \frac{}{}$$

The balance between recall and precision is found by F1 Score

d) Precision:

It refers to the division of the no. of right positive outputs by the positives outputs as the classifier predicted.

$$Precision = \frac{}{}$$

e) Recall :

It refers to the division of the no. of right outputs by the no. of all the applicable inputs.

$$Precision = \frac{}{}$$

f) Mean Squared Error

It is similar to Mean Absolute error- main contrast is that Mean Squares Error considers average of the squares of distinction in the initial qualities and the anticipated qualities. Advantages of MSE is- MSE is simpler for figure the gradient, though the other needs muddled direct program devicesto process the gradient.The impact of bigger blunders become increasingly articulated than littler blunder, consequently the model would now be able to concentrate more on the bigger mistakes.

Mean Squared Error=

g) Sensitivity and Specificity

Specificity and sensitivity are the two more popular metric that people mostly use in biology and medical related fields and define as:

Specificity which equals True Negative Rate= TN/ (TN+FP)

Sensitivity which equals Recall= TP/ (TP+FN)

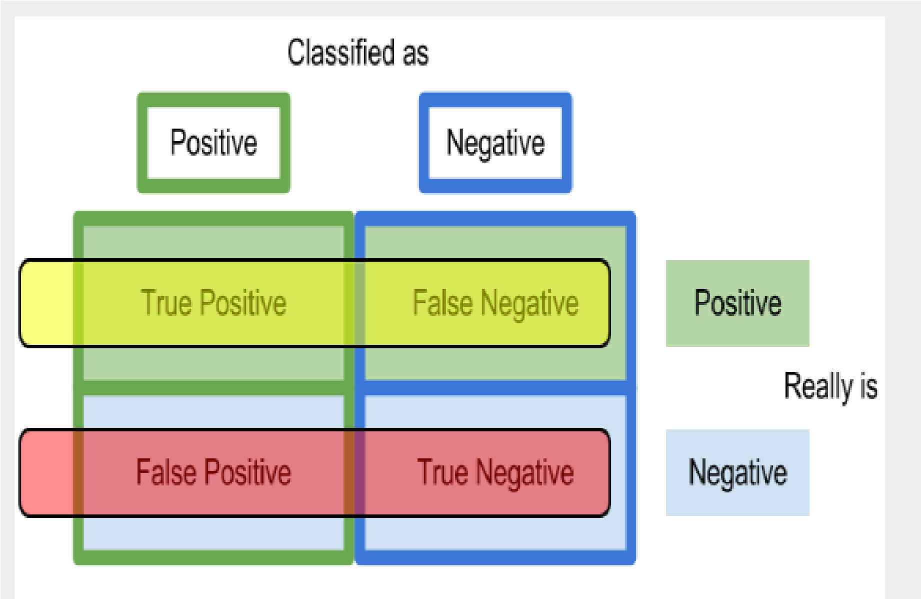


Figure 3.7: Sensitivity and specificity matrix

Model Design For Extractive Approach

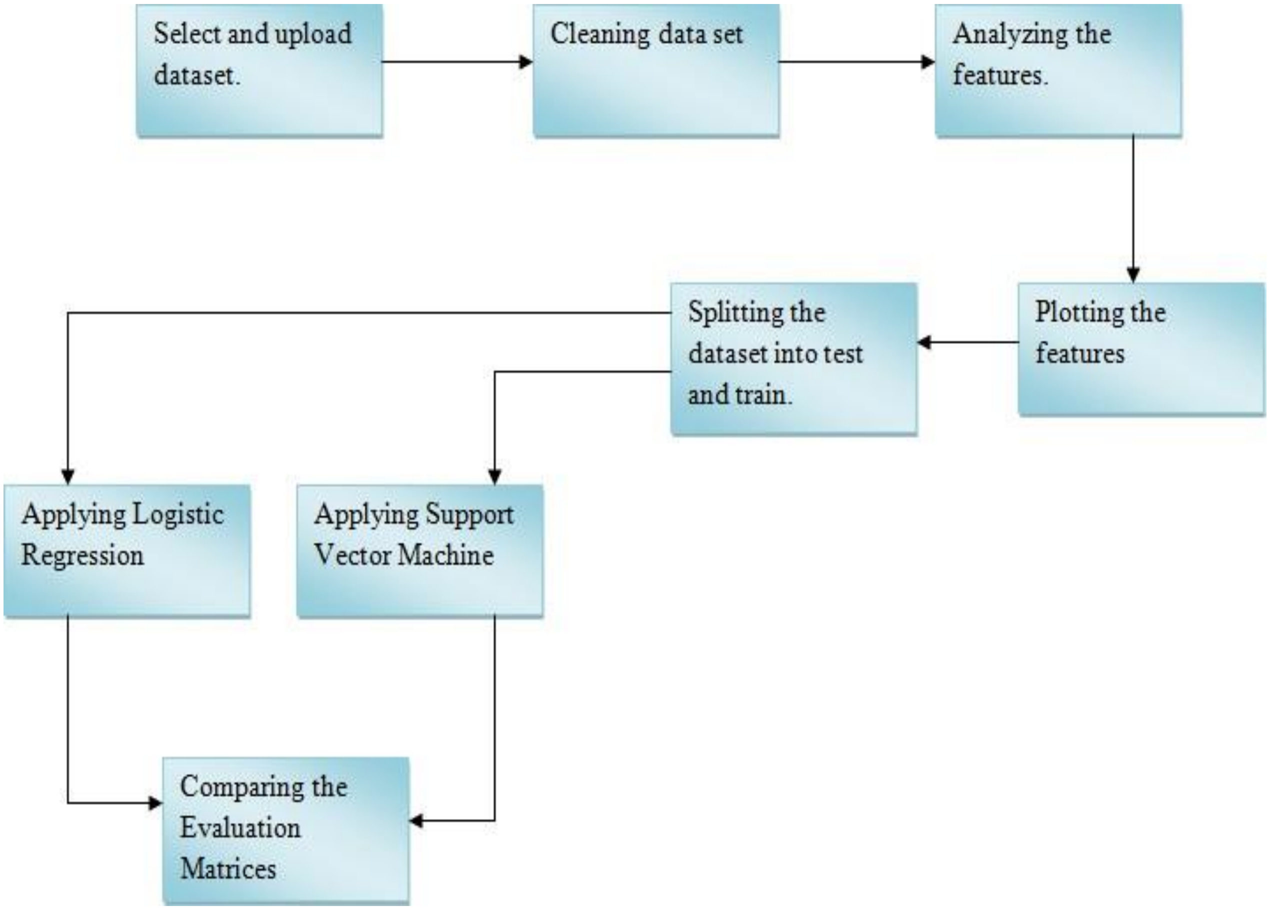

Select and upload dataset.
Cleaning data set
Analyzing the features.
Plotting the features
Splitting the dataset into test and train.
Applying Logistic Regression
Applying Support Vector Machine
Comparing the Evaluation Matrices

Figure 3.8: Model Design

Step 1: Select and upload dataset:

The selection of the dataset is very important part, as the availability of the same is very less. The primary aim of the project is to generate the dataset for a field which has a very less scope of data available online. The synthetic data generator will help generating the data which will further help the physician to do the research work and detect the early symptoms of the same. These days the patients are sometimes not willing to reveal their personal information to the world and that is why it’s the main and urgent reason of the data to be generated. The selection of data is being done on the platform jupyter notebook using the python 3 version of it. The platform uses the library pandas which help in importing the file with the extension CSV. The data is imported and is displayed and the platform by using the head command. This command will display the first five instances of the dataset.

Step 2: Cleaning Data set:

After the process of importing the first and the most important step is to clean the entire data set so that the results come out to be refined. The cleaning of the data set is being done again using some of the inbuilt libraries. First of all the parameters are compared with the target parameter and the influence of each is derived so that the useless parameters can be excluded. In our project we have removed three parameters out of 15 features and that are MRI ID, Visit, Hand because the role of these features seems very negligible towards the result. In the cleaning phase another thing is done that includes changing the non-numeric values to numeric values. Firstly the target variable i.e. Group has the non-numeric values and that were changed to Zero and One. The same was being done with the male and female parameter.

The “Demented” non-numeric value was converted to One and “Non-demented” value was converted to Zero. The data set has the target value of “Converted” which was also converted into Demented because the converted value was for the person who didn’t have dementia at the initial stages and after some time they were listed in demented group and the same was converted to One.

The same procedure was being followed for the M/F feature because the male and female values was not interpreted by the machine learning algorithm. So the Female was replaced by Zero value and Male on the other hand was replaced by One.

The null values were also dropped from the dataset as they also cause some sort of problems.

Step 3: Analyzing and Plotting the Features:

The next step is the analyses of the features meaning what feature has what impact on the target feature because at the end we have to find and predict the labeling for model preparation and synthetic data verification. The inferences can be made when all the features were compared against the number of patients.

First of all the Male and Female feature is being plotted using bar chart feature of matplotlib. This was being compared with Dementia and Non Dementia Vs number of patients. The results came out to be more common for men in the case of dementia. Women on the other hand in more in number in the bar of non-dementia.

The next feature was MMSE i.e. Mini Mental State Examination. This feature was again being compared with the number of patients using the seaborn library which is also under matplotlib. The grid is drawn to get the impact of the feature on the target feature. According to the result of the plot the socre of MMSE is much more than Non-Demented than the demented patients.

The feature ASF, eTIV, nWBV are all categorized under the domain of brain volume. The Alzheimer’s Disease effects the brain and hence effects the brain volume and if we see theoretically the brain volume for a healthy brain should be greater than the brain with disease. The plotting of all these features showcase the same trend i.e. the non-demented brain has the values of these features more than the demented brains.

The people of the age group 70-80 years have more count under the group dementia. The people who are under the age group of 90 years and plus are very less, because the survival rate is very less after the detection of this disease. The maximum number of the people fall under the dementia group which are under the specified age group.

The next feature is EDUC and after plotting this feature against the number of people and the results came out to be more in the case of dementia who are uneducated people.

The last feature is CDR (Clinical Dementia Rating) and the value for this feature varies from minimum value of 0 to maximum value 0f 3. The plot shows that the value of 0.5 is more common than any other. And around 50 patients have a value of 0.5 and this number is maximum also.

Step 4: Splitting the Dataset

After performing the various tasks such as importing, cleaning, plotting and then analyzing the features there comes a task of building up a model. The basic task in a model is training the data set and after that testing the same. So in this project we have applied two different models of Machine Learning i.e. Logistic Regression and Support Vector Machine. So again to train the dataset and then test it we need to convert the entire dataset in two parts i.e. train set and test set. To perform this we need to separate the features from which are to be predicted with the features which are used for prediction. As in our project the parameter "Group" is the target parameter as in the result we have to predict the value for a patient with specific scores of the other features to be demented or non-demented. The remaining features were all clubbed together in another independent variable i.e. X and the dependent feature is Y, which is the target value for our project.

Now the dataset is split into two parts for each variable that means that there are total of 4 data sets namely X_testset1, X_trainset1, Y_trainset,Y_testset. This is done by the help of inbuilt libraries which are imported form sklearn.model_selection. The name of the library is train_test_split. The input parameters in this function are the X and Y variables with the size of the test dataset defined and additionally the random_state defined as ZERO.

Step 5: Applying Logistic Regression.

As Logistic regression is utilized to depict information and to clarify the relationship between a dependent binary variable and 1 or more than 1 nominal, interim, ordinal or ratio level independent factors. We have applied the calculated relapse on the dataset utilizing the inbuilt libraries which were imported from sklearn. A variable is created which will contain the highlights of this capacity and afterward with assistance of this variable the datasets which were the aftereffect of train_test_split are fitted in this model. After the fruitful arrangement of the code another variable is produced which will be utilized to anticipate the qualities. It is prepared on the informational collection X_testset1.

The subsequent stage in strategic relapse is test the exactness, accuracy, f1 score and backing of the model. So to play out a similar disarray metric is being imported from the inbuilt modules. Logistic Regression is a type of regression algorithm that predicts a nonstop result: the likelihood of an occasion. We use it as a binary classifier because of the understanding of the result. Logistic regression is a sort of generalized linear regression model.

Step 6: Applying SVM (Support Vector Machine)

This report shows the implementation of this model over a set of data which has been split into 2 parts previously for its use in logistic regression. Some of the inbuilt libraries are being used for its implementation.

Step 7: Comparing the Evaluation Metrics

Both the algorithms perform well accordingly but there should be an authorized way of evaluating their performances through parameters like accuracy, precision, f1 score, confusion metric, recall, support, mean squared error, specificity and sensitivity.

Front-End Development

a) Python 3.7:

Python is a general-purpose, high-level and interpreted programming language. Its object-oriented approach and language constructs aim in helping the programmers to write clear and logical code for small as well as large scale projects. Python is garbage collected and dynamically typed. It supports a large number of programming paradigms such as functional, object-oriented and structured programming. The design philosophy of Python mainly emphasizes on code readability with notable use of whitespaces.

b) Tkinter :

The Tkinter module or Tk interface is a standard interface of Python to Tk GUI tool-kit from Scriptics(initially developed by the Sun Labs). Both Tkinter and Tk are available on almost all the Unix platforms and additionally on Macintosh and Windows systems. Tkinter compromises of various platforms. The Tk interface is given by a binary-extension module called _tkinter. This module consists the low level interface to Tk, and ought to never be utilized directly by the application programmers. It is generally a shared library or DLL, yet may in few cases be statically connected with Python interpreter.

The public interface is given through various Python modules. Tkinter module containts all the functions,classes and other things that are needed for woking with Tk toolkit. The most significant interface module is itself the Tkinter module. To use Tkinter, we simply import the Tkinter module:

```
import Tkinter
```

or,

```
from Tkinter import *
```

Tkinter supports Fifteen core widgets:

a) Button

b) Canvas

c) Checkbutton

d) Entry

e) Frame

f) Label

g) Listbox

h) Menu

i) Menubutton

j) Message

k) Radiobutton

l) Scale

m) Scrollbar

n) Text

o) Toplevel

Button, Label and frame are used in the project to develop the front end. Button is used to execute an operation or a command. Frame is a container widget. It can have a background or a border and is used ijn grouping the other widgets while creating an application. Label displays an image or text.

c) Python-Database Connectivity:

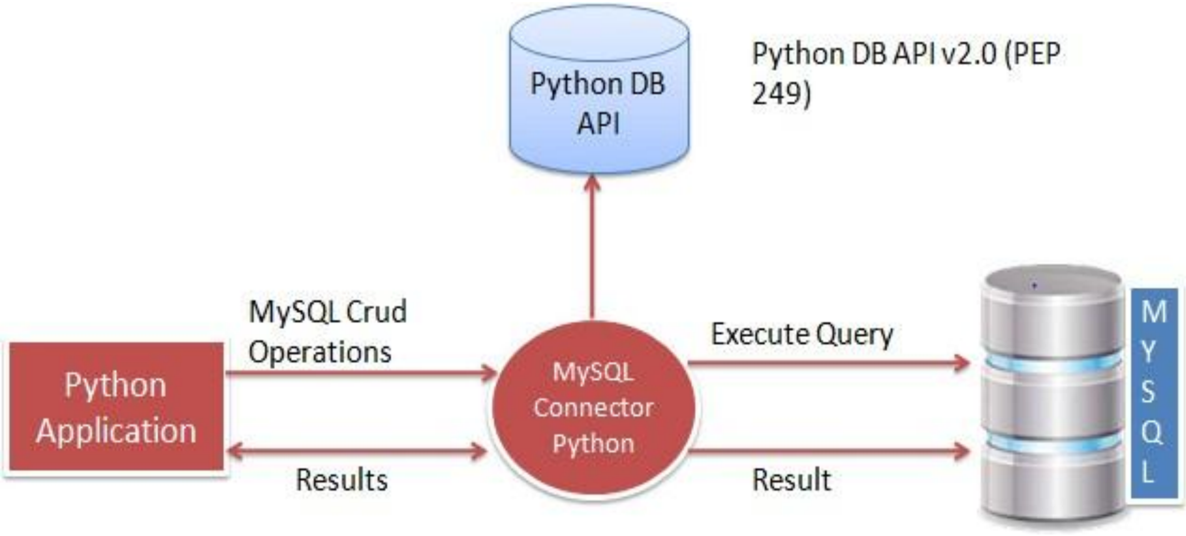


Figure 3.9: Python-database Connectivity

Arguments to connect MySql to Python are Username, Password, Hostname and Database name.

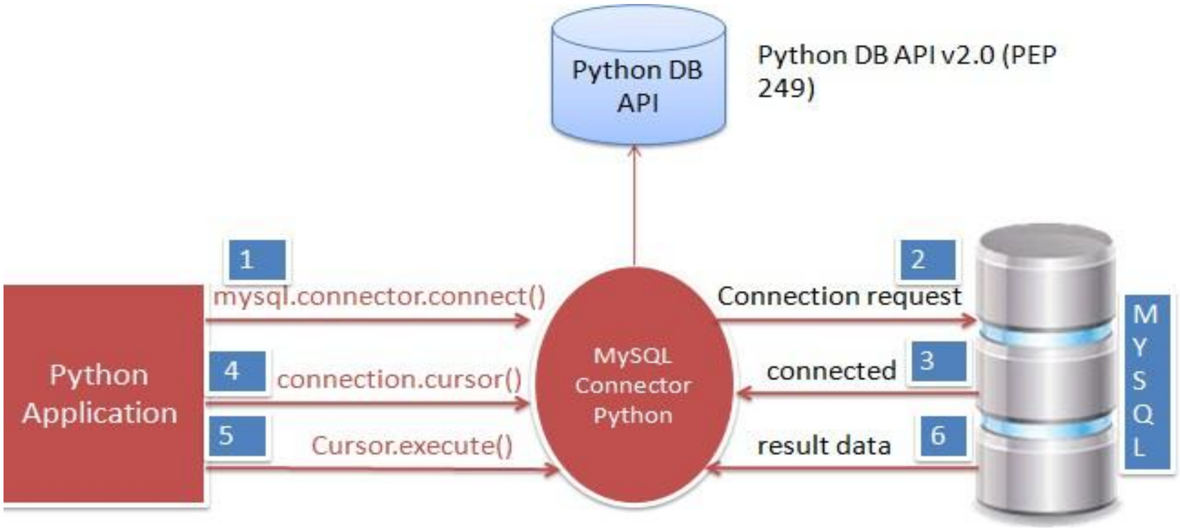


Figure 3.10: Steps to connectivity

Chapter – 4

PERFORMANCE ANALYSIS

Logistic Regression Approach

Logistic Regression used in python is basically a predictive procedure for the analysis. It is likewise utilized in Machine Learning for parallel order issues or binary classification issues. Relapse investigation is a ground-breaking measurable examination procedure. A dependent feature(variable) of advantage is utilized to foresee the estimations of other free factors in a data set.

We meet up with regression in a much intuitive way almost all the time. Such as anticipating the climate utilizing the informational collection of the climate conditions before.

It utilizes numerous techniques to predict and analyze the result, but the accentuation is basically on connection between one or more than one independent variable and dependent variable.

Logistic regression investigation predicts the result on binary variable which has just two potential results.

It is a procedure to examine an informational collection which has a dependent variable and one or more than one independent variable to anticipate the result in a binary variable, which means it will have just two results.

The reliant variable is absolute (categorical) in nature. Subordinate variable is likewise alluded as target variable and the free factors are known as the predictors.

Logistic Regression is a unique and a special instance of linear regression where only the outcome is predicted in categorical nature. It basically predicts the likelihood of the occasion utilizing the log work.

Sigmoid function or curve is used which predicts the categorical value of the problem. The outcome is decided by the threshold value i.e. win or lose

The equation of linear regression: $Y = \beta0 + \beta1x1 + \beta2x2 \dots + \beta nxn$

- We need to predict Y i.e. the dependent variable
- β0=Y-intercept i.e. the point on line touching the y axis.
- Slope of line is β1 (the slope may be positive or negative based on the connection between the independent variable and the dependent variable.)
- The independent variable is represented by x and the resultant dependent value is predicted through x.

Sigmoid function:- $\mathbf{p = 1 / 1 + e^{-y}}$

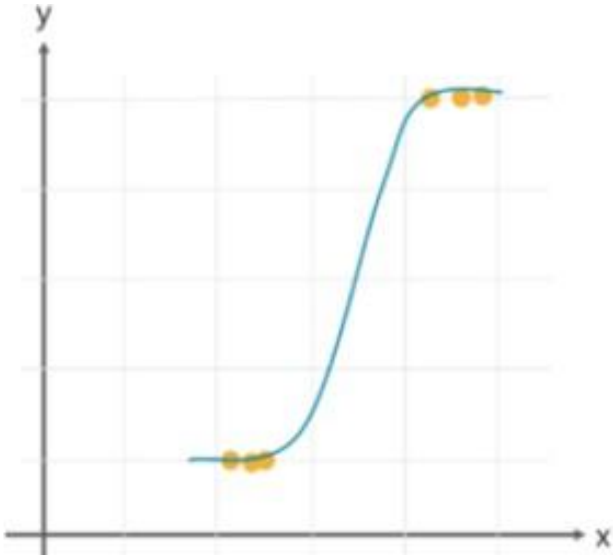


Figure 4.1: Sigmoid Function

Equation of Logistic Regression: $\boldsymbol{p = 1 / 1 + e^{-(\beta0 + \beta1X1 + \beta2X2 \dots + \beta nXn)}}$

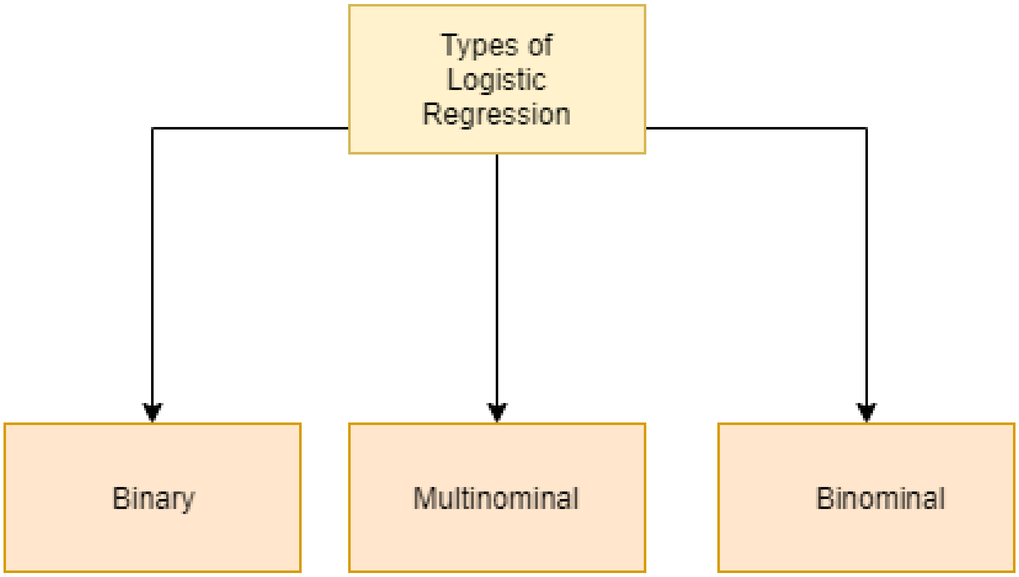


Figure 4.2: Types of Logistic Regression

- Binary logistic regression: It basically has 2 viable outcomes i.e. No or Yes.
- Multinomial logistic regression: It has 3 or even more nominal classifications. Eg- elephant, dog, cat.
- Ordinal logistic regression: It has 3 or even more ordinal classifications, ordinal means that the classifications will be in an order. Eg. user ratings i.e. 1-5.

SVM Approach:

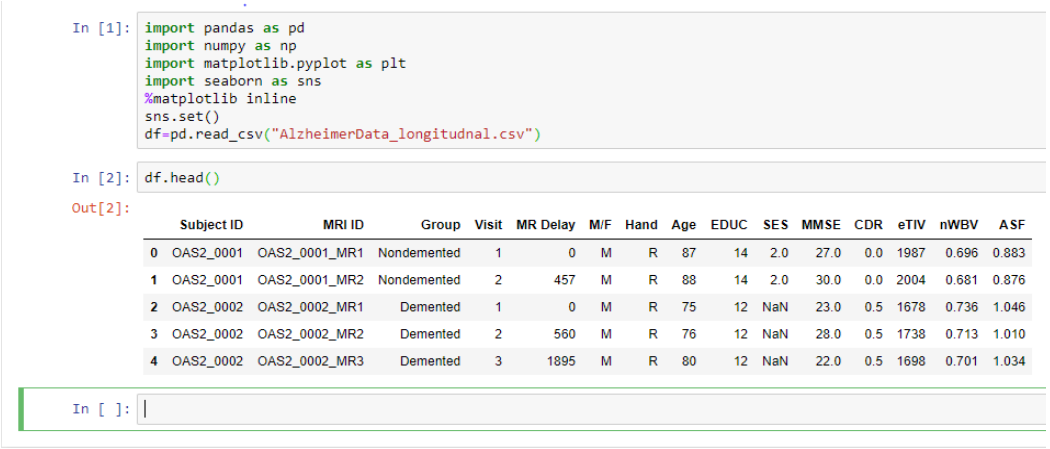
SVM offers high precision contrasted with different classifiers, for example decision trees and logistic regression. SVM is known for kernel trick to deal with nonlinear information spaces. It is utilized in a wide variety of applications, for example, intrusion detection, classification of web pages, emails and news articles.

SVM is exciting calculation and the ideas are generally straightforward. The classifier isolates information points utilizing a hyperplane with the biggest amount of the margin. That is the reason a SVM classifier is otherwise called a discriminative classifier. It finds an ideal hyperplane which helps in ordering new information points.

The target of the SVM is to locate the foremost splitting boundary between information. In 2-D space, it is like the foremost fit line that partitions your dataset. With a SVM, we are dealing in a vector space, in this manner the isolating line is really an isolating hyperplane. The best isolating hyperplane is characterized as the hyperplane that includes the "most extensive" margin between the support vectors. The hyperplane may likewise be alluded to as a choice limit. We noted in the past that the most widely recognized instinct is that you would order another information point dependent on what it is nearest to or the nearness, which is the thing that the K Nearest Neighbors calculation accomplishes for us. The fundamental issue with this goal is that, per datapoint, you need to contrast it with each and every datapoint to get the separations, subsequently the calculation simply doesn't scale well, regardless of being genuinely solid exactness insightful. What the Support Vector Machine means to do is, once, produce the "best fit" line (however really a plane, and much more explicitly a hyperplane!) that best partitions the information. When this hyperplane is found, we allude to it as a choice limit. We do this, since, this is the limit between being some class. When we compute this choice limit, we never need to do it again, except if obviously we are re-preparing the dataset. Accordingly, this calculation is going scale, in contrast to the KNN classifier.

Implementation and Analysis

In implementation of this project we have built and analysed a model and have drawn inferences through it. We have executed Logistic Regression and SVM (Support Vector Machine) on the dataset to find the labelling and further this model will be used for verification of the synthetic data. Different plots and grids are being used for better visualization of the parameters so that better result can be obtained.

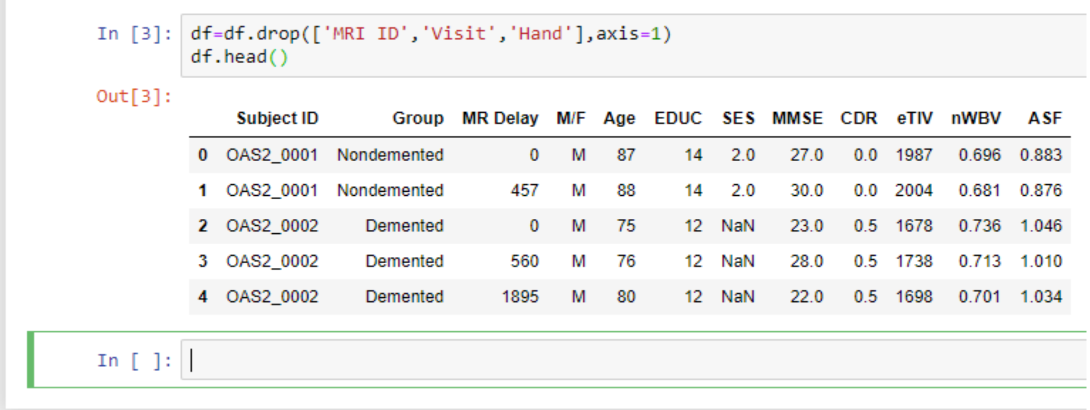


```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        %matplotlib inline
        sns.set()
        df=pd.read_csv("AlzheimerData_longitudnal.csv")

In [2]: df.head()
Out[2]:
```

	Subject ID	MRI ID	Group	Visit	MR Delay	M/F	Hand	Age	EDUC	SES	MMSE	CDR	eTIV	nWBV	ASF
0	OAS2_0001	OAS2_0001_MR1	Nondemented	1	0	M	R	87	14	2.0	27.0	0.0	1987	0.696	0.883
1	OAS2_0001	OAS2_0001_MR2	Nondemented	2	457	M	R	88	14	2.0	30.0	0.0	2004	0.681	0.876
2	OAS2_0002	OAS2_0002_MR1	Demented	1	0	M	R	75	12	NaN	23.0	0.5	1678	0.736	1.046
3	OAS2_0002	OAS2_0002_MR2	Demented	2	560	M	R	76	12	NaN	28.0	0.5	1738	0.713	1.010
4	OAS2_0002	OAS2_0002_MR3	Demented	3	1895	M	R	80	12	NaN	22.0	0.5	1698	0.701	1.034

```
In [ ]: |
```

Figure 4.3: Importing data

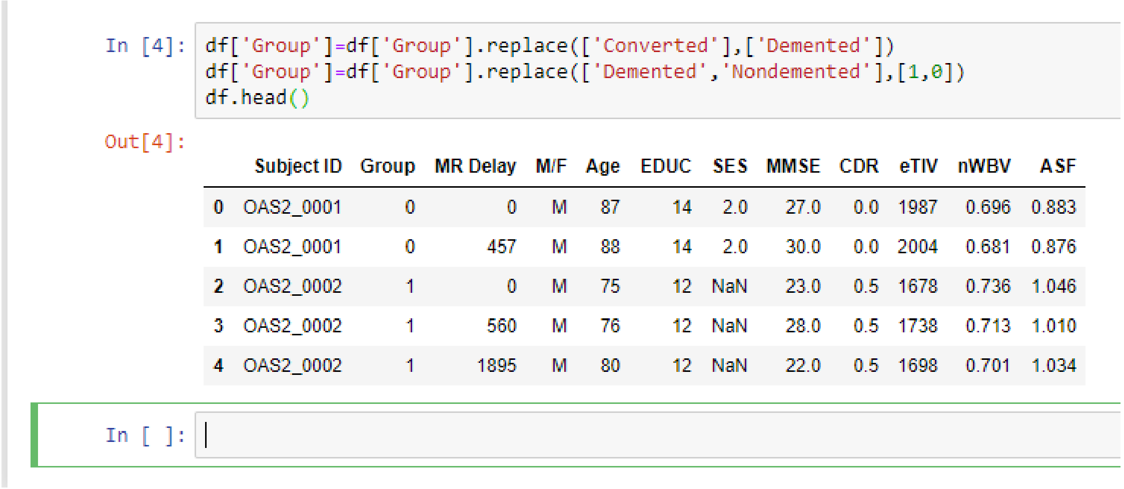


```
In [3]: df=df.drop(['MRI ID','Visit','Hand'],axis=1)
        df.head()
Out[3]:
```

	Subject ID	Group	MR Delay	M/F	Age	EDUC	SES	MMSE	CDR	eTIV	nWBV	ASF
0	OAS2_0001	Nondemented	0	M	87	14	2.0	27.0	0.0	1987	0.696	0.883
1	OAS2_0001	Nondemented	457	M	88	14	2.0	30.0	0.0	2004	0.681	0.876
2	OAS2_0002	Demented	0	M	75	12	NaN	23.0	0.5	1678	0.736	1.046
3	OAS2_0002	Demented	560	M	76	12	NaN	28.0	0.5	1738	0.713	1.010
4	OAS2_0002	Demented	1895	M	80	12	NaN	22.0	0.5	1698	0.701	1.034

```
In [ ]: |
```

Figure 4.4: Dropping the unnecessary columns



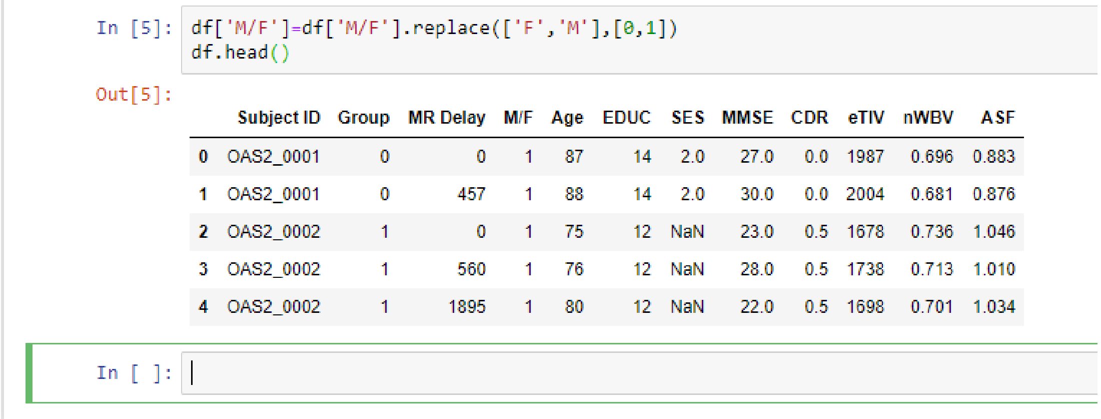
```
In [4]: df['Group']=df['Group'].replace(['Converted'],['Demented'])
        df['Group']=df['Group'].replace(['Demented','Nondemented'],[1,0])
        df.head()
```

Out[4]:

	Subject ID	Group	MR Delay	M/F	Age	EDUC	SES	MMSE	CDR	eTIV	nWBV	ASF
0	OAS2_0001	0	0	M	87	14	2.0	27.0	0.0	1987	0.696	0.883
1	OAS2_0001	0	457	M	88	14	2.0	30.0	0.0	2004	0.681	0.876
2	OAS2_0002	1	0	M	75	12	NaN	23.0	0.5	1678	0.736	1.046
3	OAS2_0002	1	560	M	76	12	NaN	28.0	0.5	1738	0.713	1.010
4	OAS2_0002	1	1895	M	80	12	NaN	22.0	0.5	1698	0.701	1.034

```
In [ ]: |
```

Figure 4.5: Conversion of datatype (1)



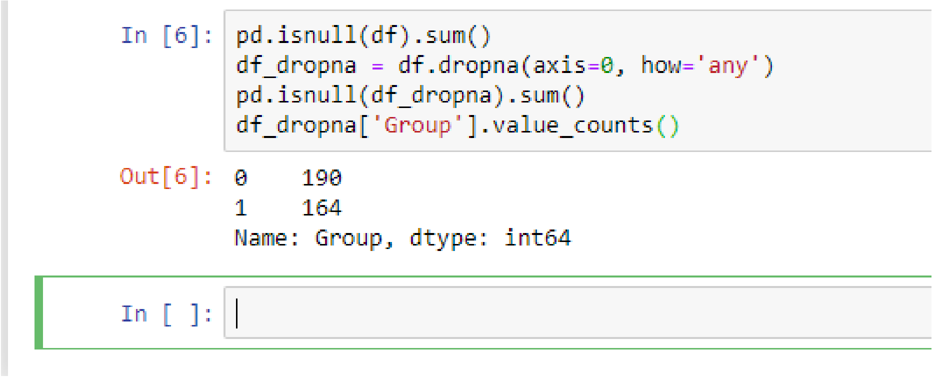
```
In [5]: df['M/F']=df['M/F'].replace(['F','M'],[0,1])
        df.head()
```

Out[5]:

	Subject ID	Group	MR Delay	M/F	Age	EDUC	SES	MMSE	CDR	eTIV	nWBV	ASF
0	OAS2_0001	0	0	1	87	14	2.0	27.0	0.0	1987	0.696	0.883
1	OAS2_0001	0	457	1	88	14	2.0	30.0	0.0	2004	0.681	0.876
2	OAS2_0002	1	0	1	75	12	NaN	23.0	0.5	1678	0.736	1.046
3	OAS2_0002	1	560	1	76	12	NaN	28.0	0.5	1738	0.713	1.010
4	OAS2_0002	1	1895	1	80	12	NaN	22.0	0.5	1698	0.701	1.034

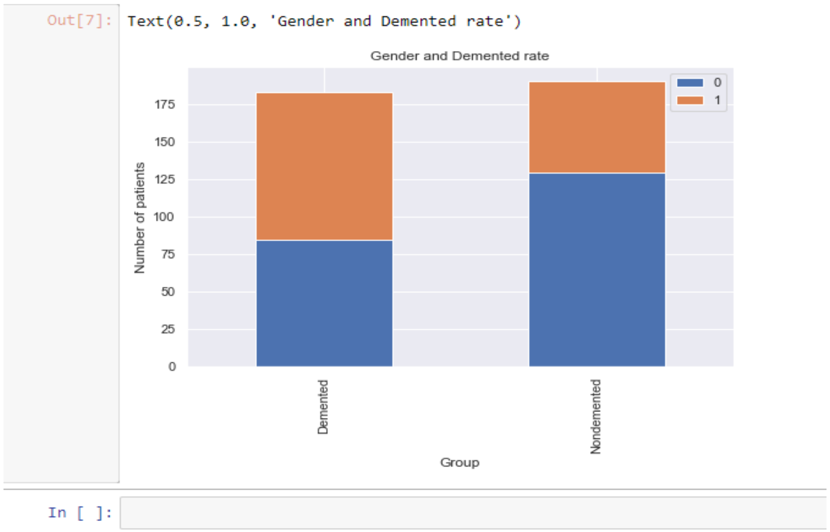
```
In [ ]: |
```

Figure 4.6: Conversion of datatype (2)



```
In [6]: pd.isnull(df).sum()
        df_dropna = df.dropna(axis=0, how='any')
        pd.isnull(df_dropna).sum()
        df_dropna['Group'].value_counts()

Out[6]: 0    190
        1    164
        Name: Group, dtype: int64

In [ ]: |
```

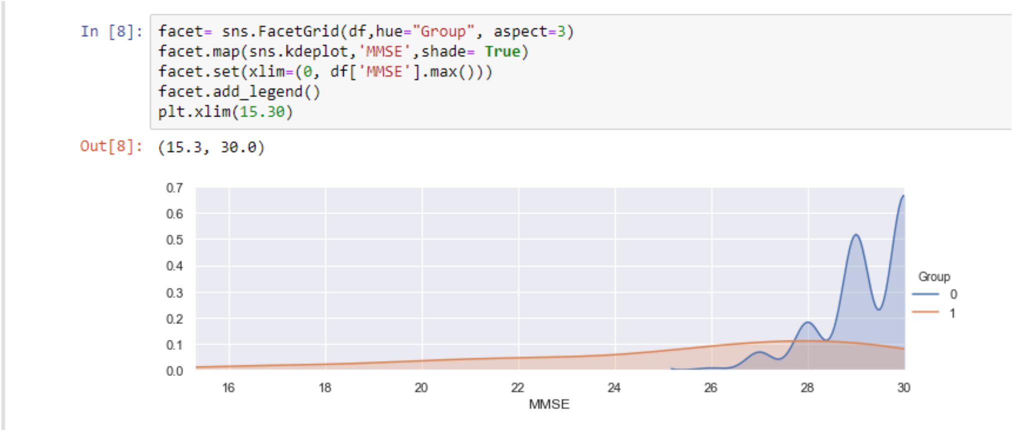
Figure 4.7: Dropping the Null values



```
Out[7]: Text(0.5, 1.0, 'Gender and Demented rate')
```

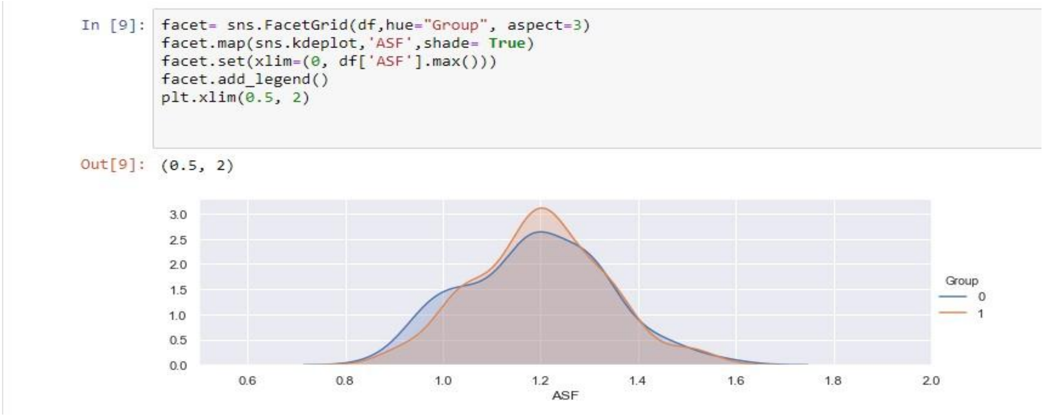
```
In [ ]:
```

Figure 4.8: Bar-Chart showing the gender feature



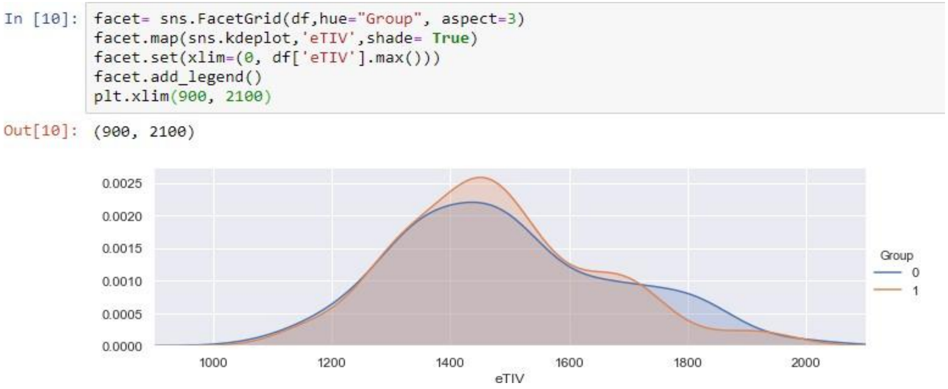
```
In [8]: facet= sns.FacetGrid(df,hue="Group", aspect=3)
        facet.map(sns.kdeplot,'MMSE',shade= True)
        facet.set(xlim=(0, df['MMSE'].max()))
        facet.add_legend()
        plt.xlim(15.30)

Out[8]: (15.3, 30.0)
```

Figure 4.9: Grid plot showing the MMSE feature



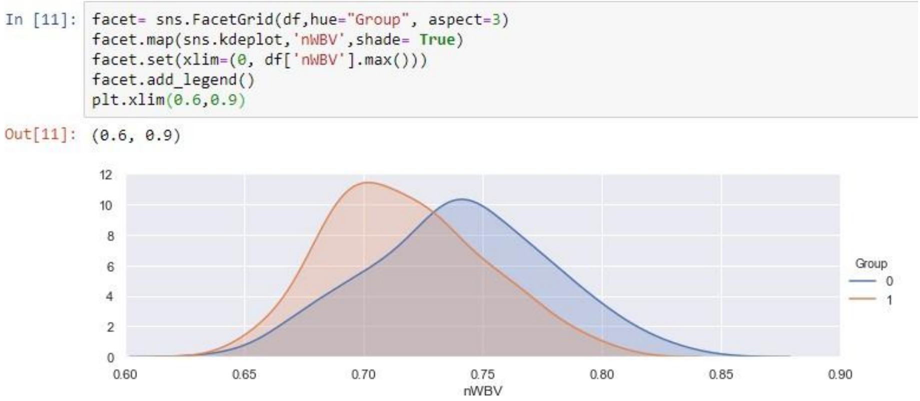
```
In [9]: facet= sns.FacetGrid(df,hue="Group", aspect=3)
        facet.map(sns.kdeplot,'ASF',shade= True)
        facet.set(xlim=(0, df['ASF'].max()))
        facet.add_legend()
        plt.xlim(0.5, 2)

Out[9]: (0.5, 2)
```

Figure 4.10: Grid plot showing ASF feature



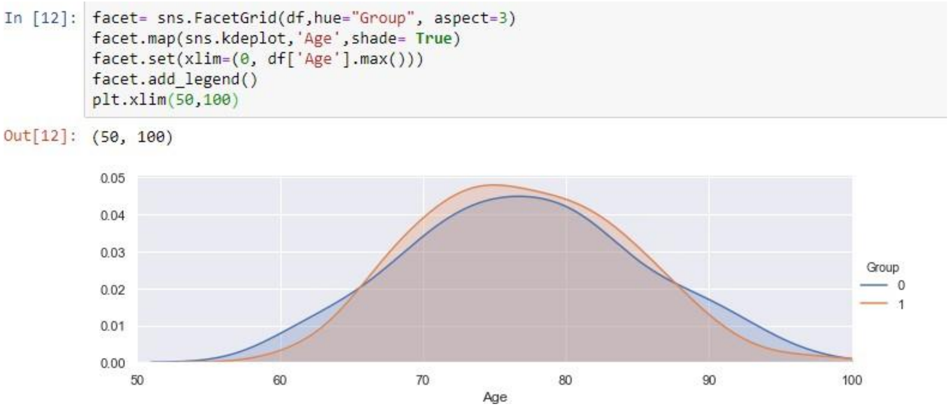
```
In [10]: facet= sns.FacetGrid(df,hue="Group", aspect=3)
         facet.map(sns.kdeplot,'eTIV',shade= True)
         facet.set(xlim=(0, df['eTIV'].max()))
         facet.add_legend()
         plt.xlim(900, 2100)

Out[10]: (900, 2100)
```

Figure 4.11: Grid plot showing eTIV feature



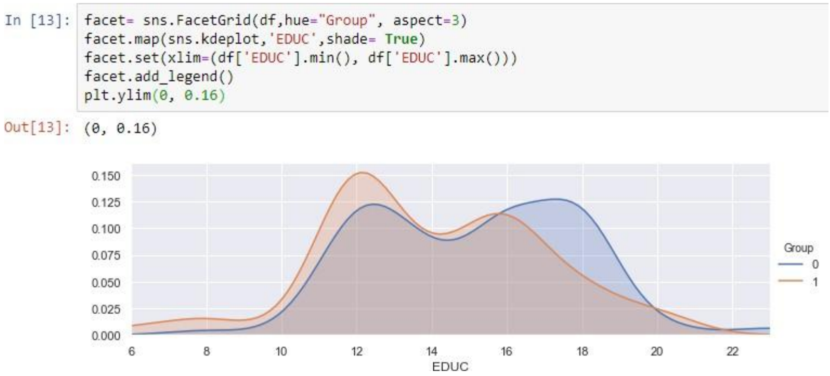
```
In [11]: facet= sns.FacetGrid(df,hue="Group", aspect=3)
         facet.map(sns.kdeplot,'nWBV',shade= True)
         facet.set(xlim=(0, df['nWBV'].max()))
         facet.add_legend()
         plt.xlim(0.6,0.9)

Out[11]: (0.6, 0.9)
```

Figure 4.12: Grid plot showing nWBV



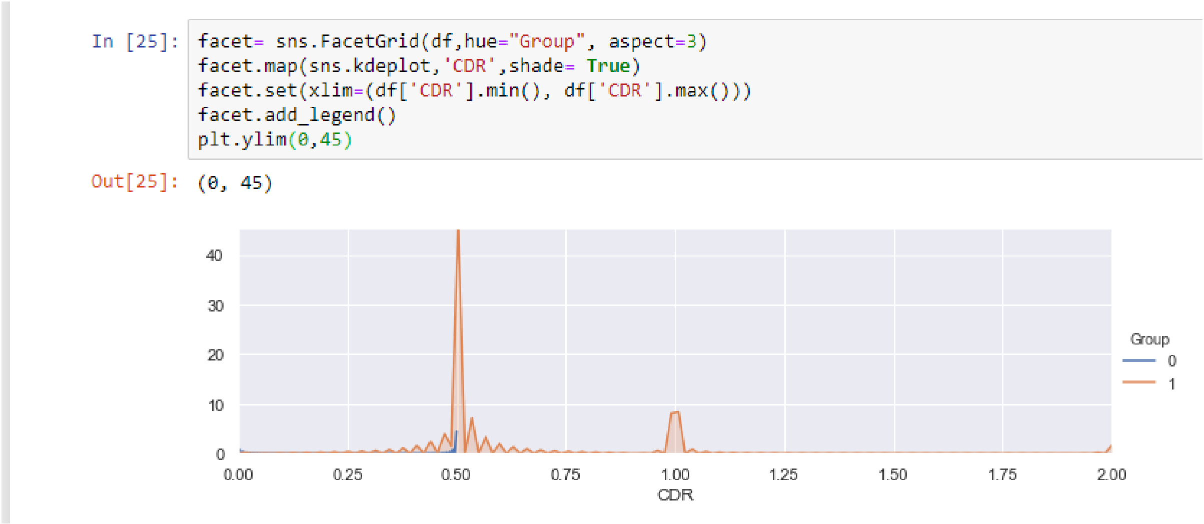
```
In [12]: facet= sns.FacetGrid(df,hue="Group", aspect=3)
         facet.map(sns.kdeplot,'Age',shade= True)
         facet.set(xlim=(0, df['Age'].max()))
         facet.add_legend()
         plt.xlim(50,100)

Out[12]: (50, 100)
```

Figure 4.13: Grid plot showing age feature



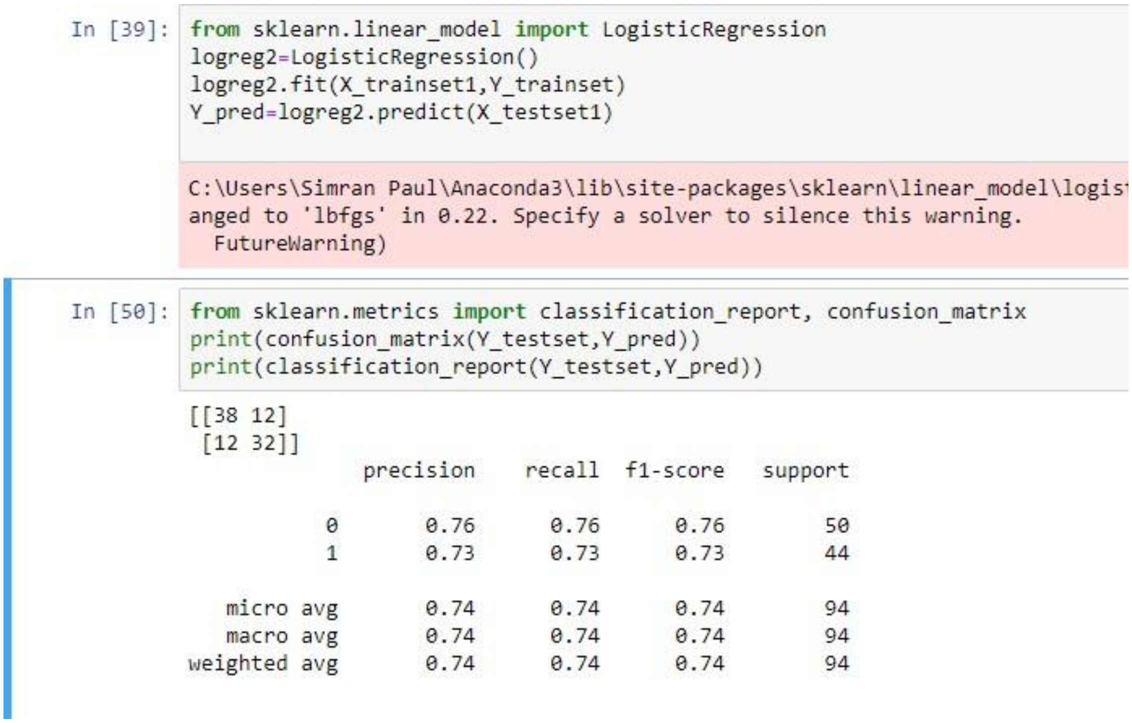
```
In [13]: facet= sns.FacetGrid(df,hue="Group", aspect=3)
         facet.map(sns.kdeplot,'EDUC',shade= True)
         facet.set(xlim=(df['EDUC'].min(), df['EDUC'].max()))
         facet.add_legend()
         plt.ylim(0, 0.16)
Out[13]: (0, 0.16)
```

Figure 4.14: Grid plot showing EDUC feature



```
In [25]: facet= sns.FacetGrid(df,hue="Group", aspect=3)
         facet.map(sns.kdeplot,'CDR',shade= True)
         facet.set(xlim=(df['CDR'].min(), df['CDR'].max()))
         facet.add_legend()
         plt.ylim(0,45)
Out[25]: (0, 45)
```

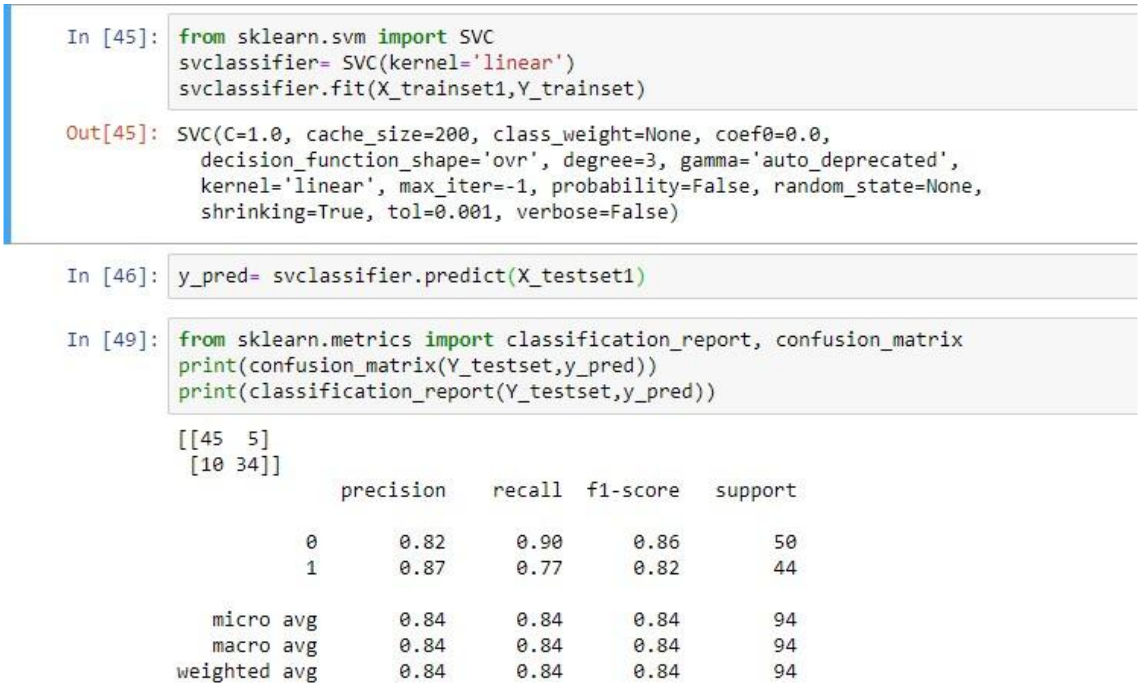
Figure 4.15: Grid plot showing the CDR feature



```
In [39]: from sklearn.linear_model import LogisticRegression
         logreg2=LogisticRegression()
         logreg2.fit(X_trainset1,Y_trainset)
         Y_pred=logreg2.predict(X_testset1)

         C:\Users\Simran Paul\Anaconda3\lib\site-packages\sklearn\linear_model\logis
         anged to 'lbfgs' in 0.22. Specify a solver to silence this warning.
           FutureWarning)
```

```
In [50]: from sklearn.metrics import classification_report, confusion_matrix
         print(confusion_matrix(Y_testset,Y_pred))
         print(classification_report(Y_testset,Y_pred))

         [[38 12]
          [12 32]]
                       precision    recall  f1-score   support

                    0       0.76      0.76      0.76        50
                    1       0.73      0.73      0.73        44

            micro avg       0.74      0.74      0.74        94
            macro avg       0.74      0.74      0.74        94
         weighted avg       0.74      0.74      0.74        94
```

Figure 4.16: Evaluation metric for Logistic Regression



```
In [45]: from sklearn.svm import SVC
         svclassifier= SVC(kernel='linear')
         svclassifier.fit(X_trainset1,Y_trainset)

Out[45]: SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
           decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
           kernel='linear', max_iter=-1, probability=False, random_state=None,
           shrinking=True, tol=0.001, verbose=False)
```

```
In [46]: y_pred= svclassifier.predict(X_testset1)
```

```
In [49]: from sklearn.metrics import classification_report, confusion_matrix
         print(confusion_matrix(Y_testset,y_pred))
         print(classification_report(Y_testset,y_pred))

         [[45  5]
          [10 34]]
                       precision    recall  f1-score   support

                    0       0.82      0.90      0.86        50
                    1       0.87      0.77      0.82        44

            micro avg       0.84      0.84      0.84        94
            macro avg       0.84      0.84      0.84        94
         weighted avg       0.84      0.84      0.84        94
```

Figure 4.17: Evaluation metric for SVM

Explanations for the above mentioned figures:

In figure 4.3: Basic libraries are imported such as pandas, numpy, matplotlib and seaborn which will be used further in our project. With the help of pandas library we are importing the dataset from the folder in .csv extension. The first five instances of data are shown using the head() command.

In figure 4.4: After analysis, three parameters seemed useless which were MRI ID, Visit(number of visits of patients to hospital), Hand(left, right). So they all were removed using the drop() command and then again the data is displayed.

In figure 4.5: To implement logistic regression we have to change the target values from words to numbers i.e zero and one. The targeted variable whose value is "Converted" is being converted into Demented as the end result is demented. The parameters "Demented" is referred as 1 and "Non-Demented" is referred as 0.

In figure 4.6: The same technique is used for the column M/F. "Male" is referred as 1 and "Female" is referred as 0.

In figure 4.7: The null values are being deleted.

In figure 4.8: Bar chart is made using Matplotlib library and It tells us the measurement of male and female who has dementia. As per the graph, first bar shows that men have more common dementia problem then women as the number of women is less in non-dementia.

In figure 4.9: Figure tells about the value of MMSE (Mini Mental State Examination) for both dementia and non-dementia, it comes out to be higher for non-demented group i.e. 0

In figure 4.10, 4.11, 4.12: Figure tells about the value of ASF (Atlas Scaling Factor), eTIV (Estimated Total Intracranial Volume), nWBV (Normalize Whole Brain Volume). These are the brain volume ratios and in all the three cases the values are higher for non-demented group.

In figure 4.13: Figure tells us that the groups of people who have dementia are of the age group 70-80 years. It can be guessed that the patients who have suffered this kind of disease has low survival rate so there are less 90 year old candidate.

In figure 4.14: Figure compares the number of EDUC (Education Years) for both demented and non-demented people and results came out to be more for demented people.

In figure 4.15: Figure shows the CDR rating (Clinical Dementia Rating) and the value for CDR is more in case non-dementia. According to this feature CDR value of 0.5 means that the state is questionable so around 50 patients have CDR value 0.5 and that is also the maximum value.

In figure 4.16: Logistic Regression module is being imported and a model is trained named as logreg2.
logreg2 is then fitted with the datasets that were the result of train_test_split i.e. X_trainset1, Y_trainset. Then a variable Y_pred is assigned to a function predict() on the value of X_testset1.
We have applied the inbuilt function for precision recall f1-score and support and the values are shown in the figure respectively

In figure 4.17: SVC module is being imported and a model is trained named as svclassifier. This is then fitted with the dataset that were previously generated. Then a variable y_pred is assigned to a function predict() on the value of X_testset1. We have applied the inbuilt function for precision recall f1-score and support and the values are shown in the figure respectively

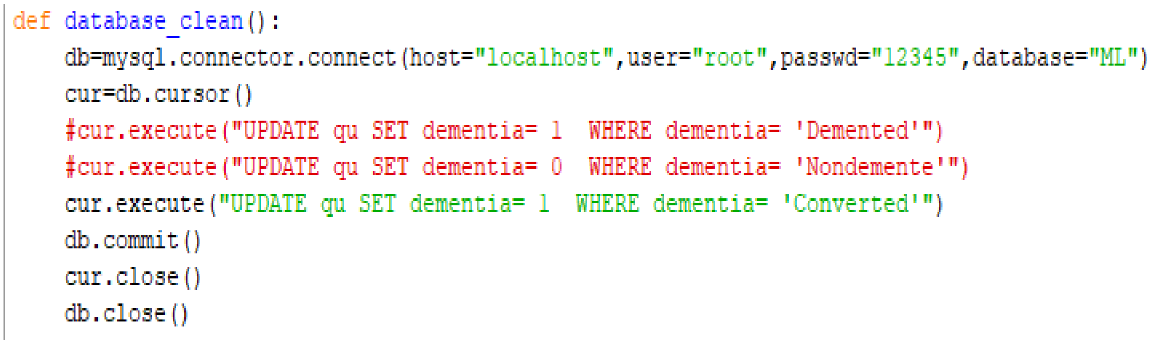
4.3.1 Front End Implementation

Before doing all the work we will import some libraries to solve our purpose. We have imported tkinter for the frontend construction, CSV to import the file, pandas and numpy for our machine learning work, matplotlib and seaborn for the plotting of the features and at last mysql.connector for the connection of the database to our project .First of all connection of our database to your python project is made. We will import our CSV file to the database for further use. This will be done using table CREATE queries. A table is made with the same number of instances as in the dataset. We will use the load data local infile command to import the file. This command uses the local path of the file in your computer to fetch the details. The database has to be closed so that is why we use the commit and close command.

For cleaning the dataset we will further make another function following the same steps for accessing the database to get the dataset. In this piece of code we have converted the non-numerical values into numeric so that we can run our model properly. So the “Demented” and “Converted” value has been converted to “1” and “Non-demented” to “0”. Again the database has been closed using the previous procedure. The same thing has been done in the male/female column, the “M” is converted to “1” and “F” is converted to “0”.

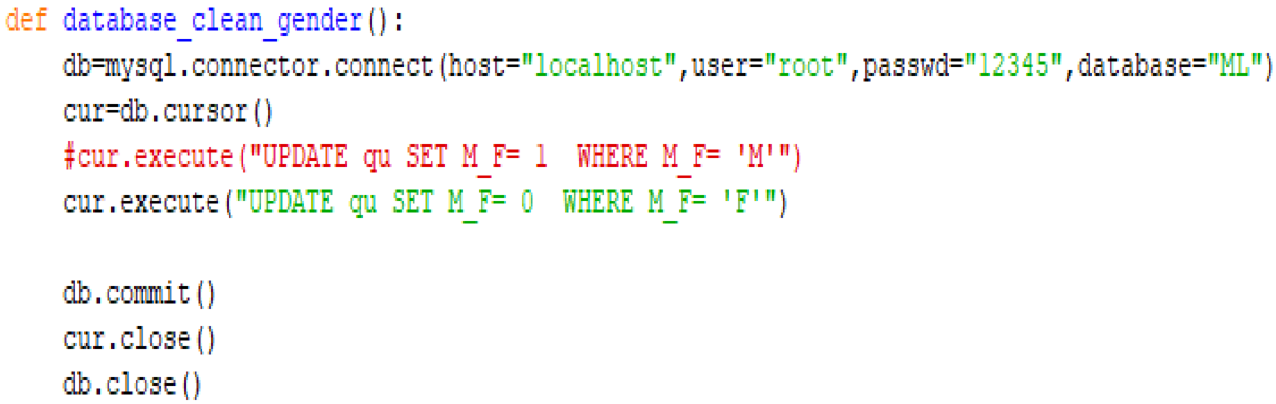
For further cleaning procedure, some of the columns are being dropped after proper analysis. We get to know about three of these columns so we have eliminated them one by one using the database and queries. Again we have to make a function and follow the steps as above and then we will use table queries to do the same and we will end this up using the commit and close command.

Now further the task is to do all the above work by just a single click of a button. So we will use the tkinter. This is done with the help of tkinter library and we will start by initializing the a variable ti Tk() which will generate a separate window for display



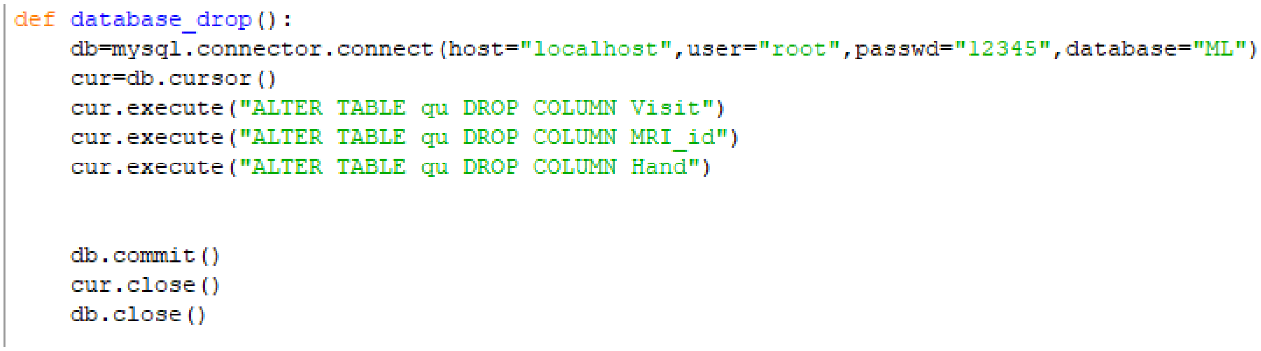
```
def database_clean():
    db=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="ML")
    cur=db.cursor()
    #cur.execute("UPDATE qu SET dementia= 1  WHERE dementia= 'Demented'")
    #cur.execute("UPDATE qu SET dementia= 0  WHERE dementia= 'Nondemente'")
    cur.execute("UPDATE qu SET dementia= 1  WHERE dementia= 'Converted'")
    db.commit()
    cur.close()
    db.close()
```

Figure 4.18: Creating a connection



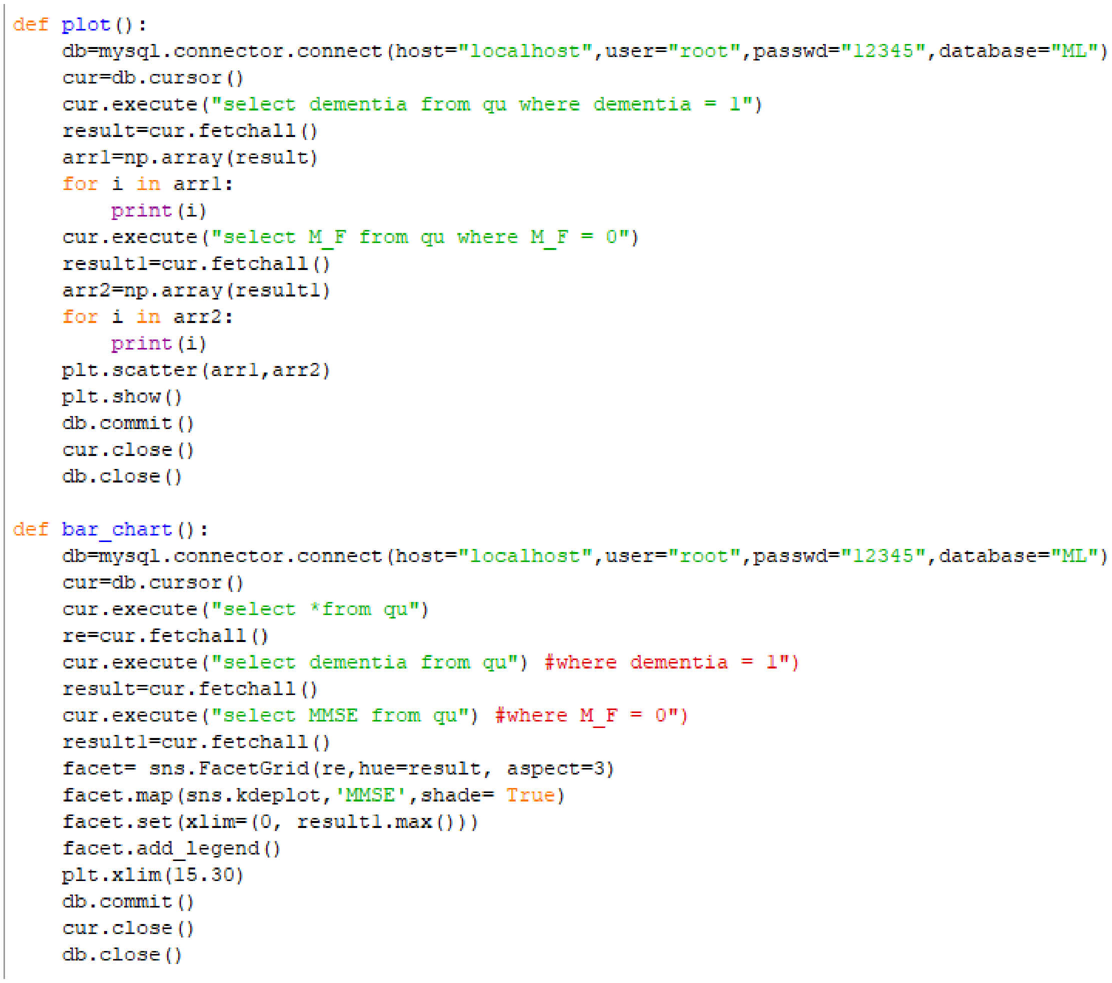
```
def database_clean_gender():
    db=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="ML")
    cur=db.cursor()
    #cur.execute("UPDATE qu SET M_F= 1  WHERE M_F= 'M'")
    cur.execute("UPDATE qu SET M_F= 0  WHERE M_F= 'F'")

    db.commit()
    cur.close()
    db.close()
```

Figure 4.19: Cleaning the dataset



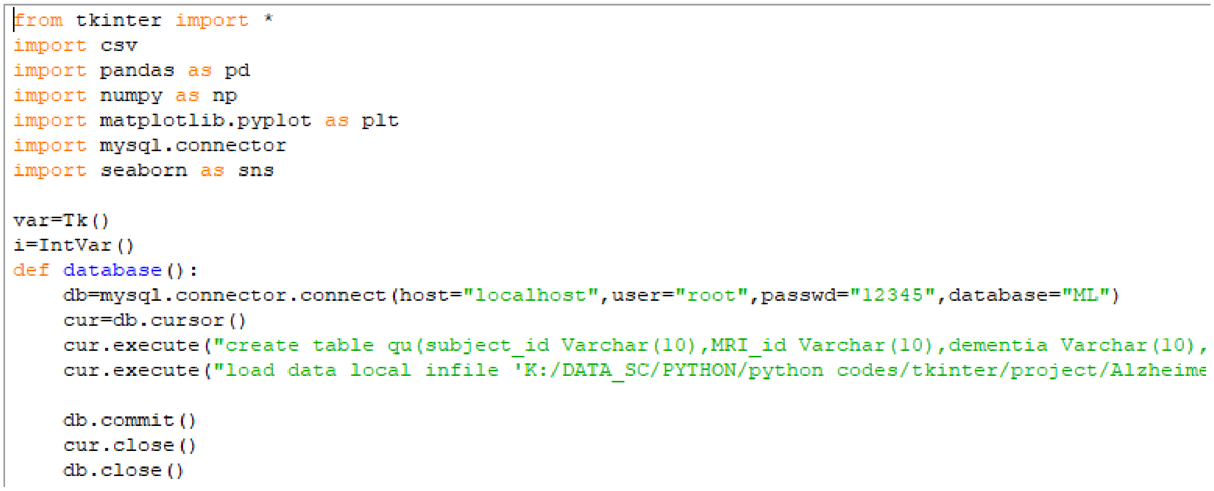
```
def database_drop():
    db=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="ML")
    cur=db.cursor()
    cur.execute("ALTER TABLE qu DROP COLUMN Visit")
    cur.execute("ALTER TABLE qu DROP COLUMN MRI_id")
    cur.execute("ALTER TABLE qu DROP COLUMN Hand")


    db.commit()
    cur.close()
    db.close()
```

Figure 4.20: Dropping Columns



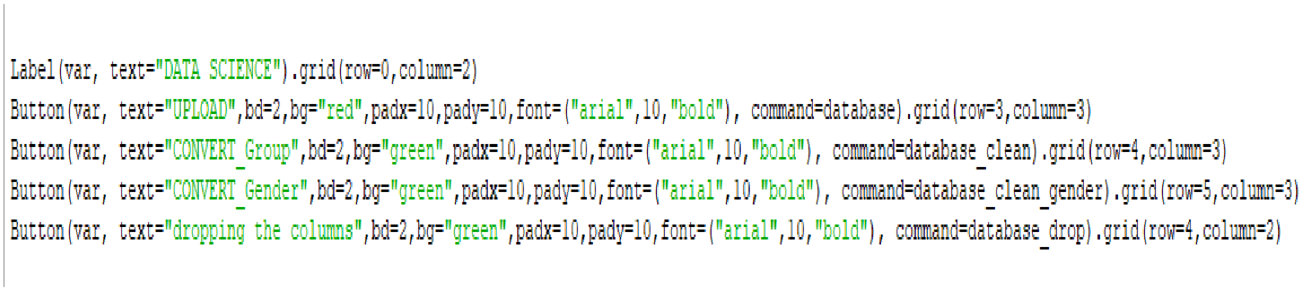
```
def plot():
    db=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="ML")
    cur=db.cursor()
    cur.execute("select dementia from qu where dementia = 1")
    result=cur.fetchall()
    arr1=np.array(result)
    for i in arr1:
        print(i)
    cur.execute("select M_F from qu where M_F = 0")
    result1=cur.fetchall()
    arr2=np.array(result1)
    for i in arr2:
        print(i)
    plt.scatter(arr1,arr2)
    plt.show()
    db.commit()
    cur.close()
    db.close()

def bar_chart():
    db=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="ML")
    cur=db.cursor()
    cur.execute("select *from qu")
    re=cur.fetchall()
    cur.execute("select dementia from qu") #where dementia = 1")
    result=cur.fetchall()
    cur.execute("select MMSE from qu") #where M_F = 0")
    result1=cur.fetchall()
    facet= sns.FacetGrid(re,hue=result, aspect=3)
    facet.map(sns.kdeplot,'MMSE',shade= True)
    facet.set(xlim=(0, result1.max()))
    facet.add_legend()
    plt.xlim(15.30)
    db.commit()
    cur.close()
    db.close()
```

Figure 4.21:Analyzing the data i.e. Plotting the features



```
from tkinter import *
import csv
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import mysql.connector
import seaborn as sns

var=Tk()
i=IntVar()
def database():
    db=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="ML")
    cur=db.cursor()
    cur.execute("create table qu(subject_id Varchar(10),MRI_id Varchar(10),dementia Varchar(10),
    cur.execute("load data local infile 'K:/DATA_SC/PYTHON/python codes/tkinter/project/Alzheime

    db.commit()
    cur.close()
    db.close()
```

Figure 4.22:Initialzing the database and creation of table

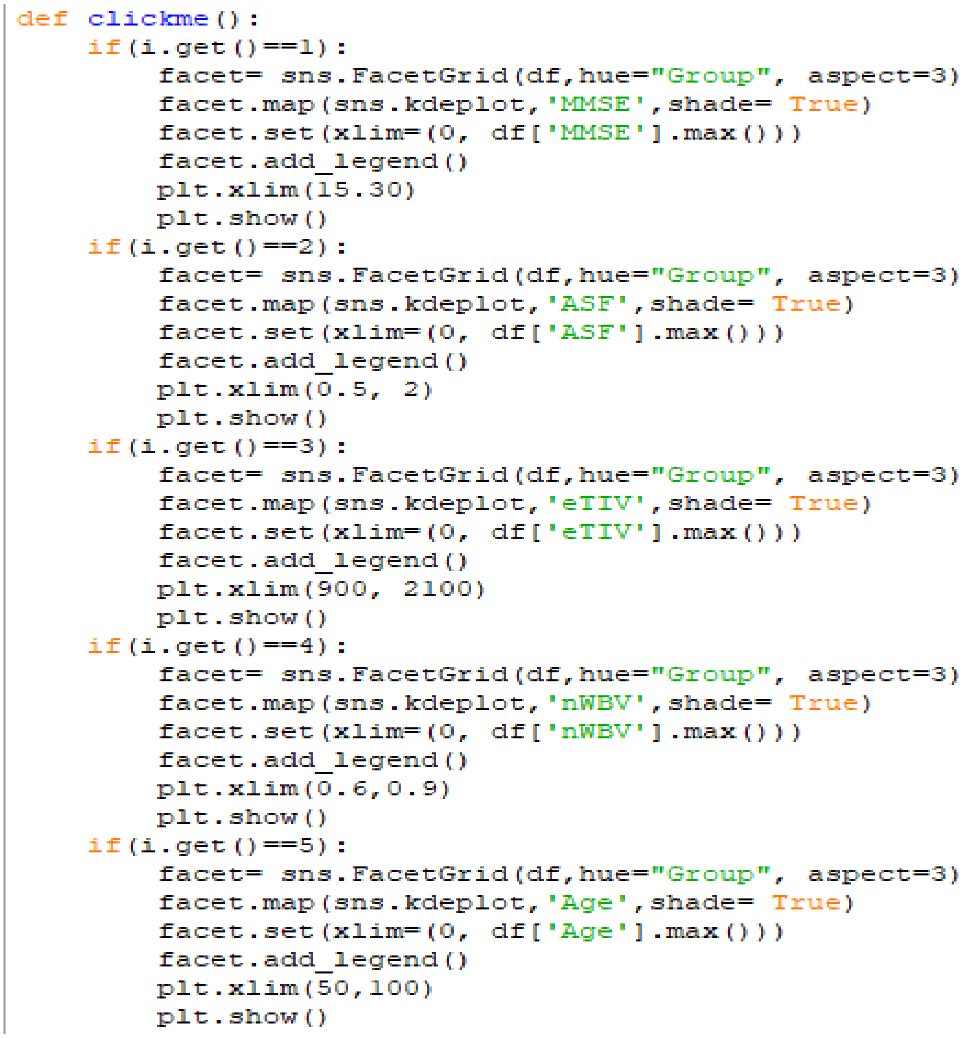


```
Label(var, text="DATA SCIENCE").grid(row=0,column=2)
Button(var, text="UPLOAD",bd=2,bg="red",padx=10,pady=10,font=("arial",10,"bold"), command=database).grid(row=3,column=3)
Button(var, text="CONVERT_Group",bd=2,bg="green",padx=10,pady=10,font=("arial",10,"bold"), command=database_clean).grid(row=4,column=3)
Button(var, text="CONVERT_Gender",bd=2,bg="green",padx=10,pady=10,font=("arial",10,"bold"), command=database_clean_gender).grid(row=5,column=3)
Button(var, text="dropping the columns",bd=2,bg="green",padx=10,pady=10,font=("arial",10,"bold"), command=database_drop).grid(row=4,column=2)
```

Figure 4.23: Buttons on Tkinter with commands

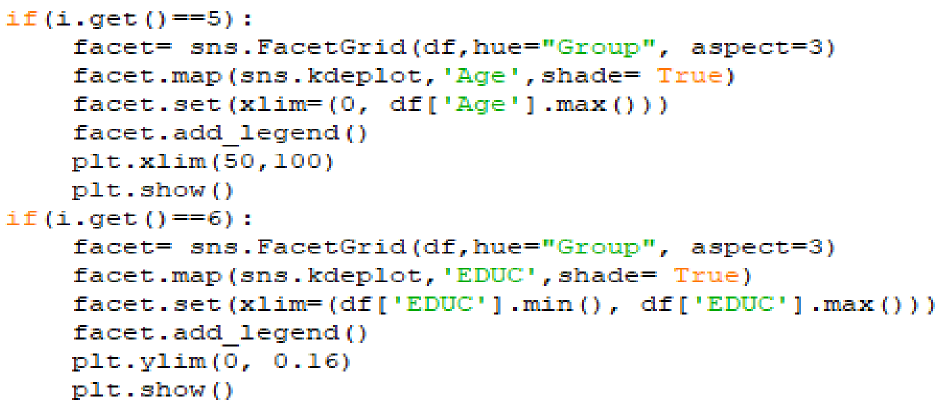
So now to print the graphs we have used Radio buttons and if loop. In the radio button we use “value” which is the value, that is assigned to the selection made by the user. Like we have assigned “1” to MMSE selection and so on. A variable is generated “I” with the data-type IntVar() so that we can use the input to perform actions. Similarly we have created 6 different radio buttons within different values so that the user can get the options to select. A button is made to perform the command when we select the radio button value.

A new function is made named clickme(), this function is for the button click on the selection of radiobuttons. Different codes are written according to different features using the simple if loop. The value for the radio button is being extracted with get() function. And accordingly the functions are performed. The facet variable is made and we use seaborn library to get the graphs.



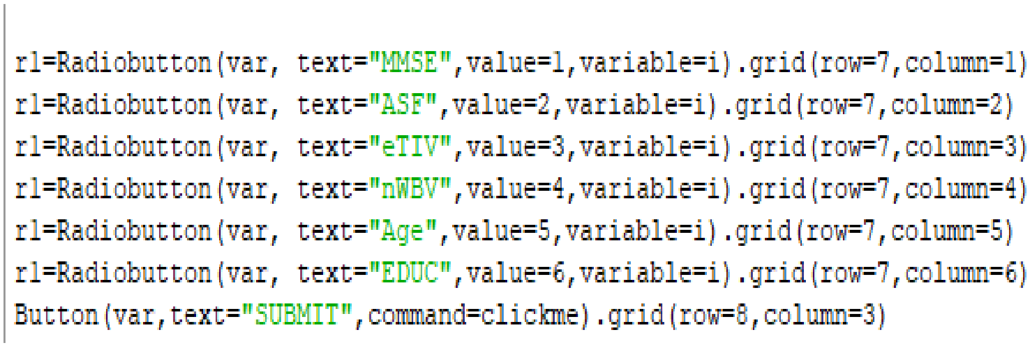
```
def clickme():
    if(i.get()==1):
        facet= sns.FacetGrid(df,hue="Group", aspect=3)
        facet.map(sns.kdeplot,'MMSE',shade= True)
        facet.set(xlim=(0, df['MMSE'].max()))
        facet.add_legend()
        plt.xlim(15.30)
        plt.show()
    if(i.get()==2):
        facet= sns.FacetGrid(df,hue="Group", aspect=3)
        facet.map(sns.kdeplot,'ASF',shade= True)
        facet.set(xlim=(0, df['ASF'].max()))
        facet.add_legend()
        plt.xlim(0.5, 2)
        plt.show()
    if(i.get()==3):
        facet= sns.FacetGrid(df,hue="Group", aspect=3)
        facet.map(sns.kdeplot,'eTIV',shade= True)
        facet.set(xlim=(0, df['eTIV'].max()))
        facet.add_legend()
        plt.xlim(900, 2100)
        plt.show()
    if(i.get()==4):
        facet= sns.FacetGrid(df,hue="Group", aspect=3)
        facet.map(sns.kdeplot,'nWBV',shade= True)
        facet.set(xlim=(0, df['nWBV'].max()))
        facet.add_legend()
        plt.xlim(0.6,0.9)
        plt.show()
    if(i.get()==5):
        facet= sns.FacetGrid(df,hue="Group", aspect=3)
        facet.map(sns.kdeplot,'Age',shade= True)
        facet.set(xlim=(0, df['Age'].max()))
        facet.add_legend()
        plt.xlim(50,100)
        plt.show()
```

Figure 4.24: Function Definition(1)



```
if(i.get()==5):
    facet= sns.FacetGrid(df,hue="Group", aspect=3)
    facet.map(sns.kdeplot,'Age',shade= True)
    facet.set(xlim=(0, df['Age'].max()))
    facet.add_legend()
    plt.xlim(50,100)
    plt.show()
if(i.get()==6):
    facet= sns.FacetGrid(df,hue="Group", aspect=3)
    facet.map(sns.kdeplot,'EDUC',shade= True)
    facet.set(xlim=(df['EDUC'].min(), df['EDUC'].max()))
    facet.add_legend()
    plt.ylim(0, 0.16)
    plt.show()
```

Fig 4.25: Function Definition(2)



```
r1=Radiobutton(var, text="MMSE",value=1,variable=i).grid(row=7,column=1)
r1=Radiobutton(var, text="ASF",value=2,variable=i).grid(row=7,column=2)
r1=Radiobutton(var, text="eTIV",value=3,variable=i).grid(row=7,column=3)
r1=Radiobutton(var, text="nWBV",value=4,variable=i).grid(row=7,column=4)
r1=Radiobutton(var, text="Age",value=5,variable=i).grid(row=7,column=5)
r1=Radiobutton(var, text="EDUC",value=6,variable=i).grid(row=7,column=6)
Button(var,text="SUBMIT",command=clickme).grid(row=8,column=3)
```

Figure 4.26: Radio Buttons initialization

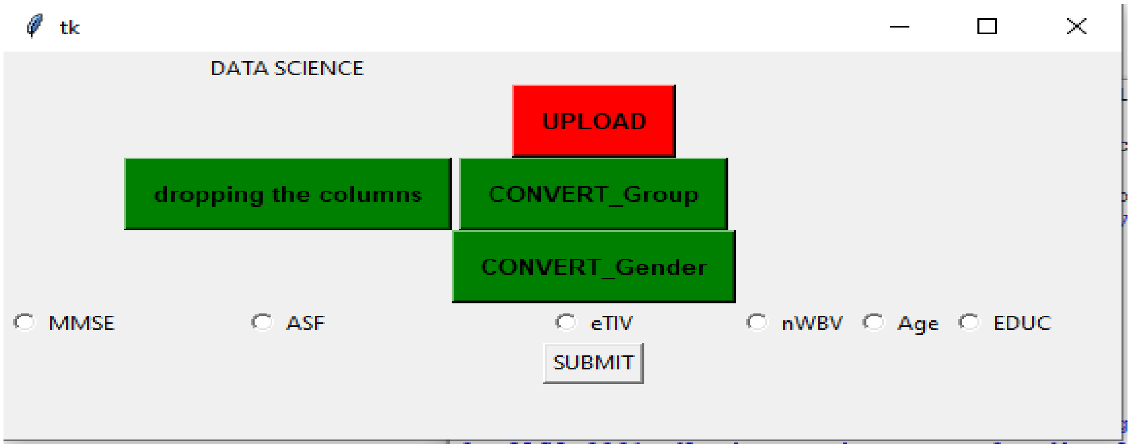


Figure 4.27: Front End

Chapter – 5

CONCLUSION

5.1 Conclusion and Future Scope

In this project, we used two machine learning algorithms i.e. Logistic Regression and Support Vector Machine on the Alzhiemer's disease dataset to build a model and further compare the accuracy obtained by each of the model. The project aimed at analyzing the data through graphs to predict which of the features are more likely to cause dementia in a person. Through this project, given the features, we can easily predict whether a person will have dementia or not.

Furthermore, a front end is including radiobuttons and labels is developed using tkinter to perform functions like dropping the columns, converting gender and group to integer values and upload using the database and queries. All this can be done by a single click of button hence improving the usability of project.

Proposals for Future Works

Before deploying the model, it can be tested in systems and further used in a real time scenario. Also this project can be used for generation of synthetic data for the early detetion of Alzheimer's disease by using the train and test model.

REFERENCES

[1] Elaheh Moradi, Antonietta Pepe, Christian Gaser, Heikki Huttunen, Jussi Tohka, Machine learning framework for early MRI-based Alzheimer's conversion prediction in MCI subjects, In NeuroImage, Volume 104, 2015, Pages 398-412, ISSN 1053-8119, doi.org/10.1016/j.neuroimage.2014.10.002.

[2] Zhang Y, Dong Z, Phillips P, et al. Detection of subjects and brain regions related to Alzheimer’s disease using 3D MRI scans based on eigenbrain and machine learning. Frontiers in Computational Neuroscience. 2015;9:66. doi:10.3389/fncom.2015.00066.

[3] Ye, D.H., Pohl, K.M., Davatzikos, C., 2011. Semi-supervised pattern classification: application to structural MRI of Alzheimer's disease. Pattern Recognition in NeuroImaging(PRNI), 2011 International Workshop on. IEEE, pp. 1–4. http://doi:10.1109/PRNI.2011.12.

[4] Marcus DS, Fotenos AF, Csernansky JG, Morris JC, Buckner RL. Open Access Series of Imaging Studies (OASIS): Longitudinal MRI Data in Nondemented and Demented Older Adults. Journal of cognitive neuroscience. 2010;22(12):2677-2684. doi:10.1162/jocn.2009.21407.

[5] Magnin, B., Mesrob, L., Kinkingnéhun, S. et al. Support vector machine-based classification of Alzheimer’s disease from whole-brain anatomical MRI. Neuroradiology (2009) 51: 73. doi.org/10.1007/s00234-008-0463-x

[6] Marcus, DS, Wang, TH, Parker, J, Csernansky, JG, Morris, JC, Buckner, RL. Open Access Series of Imaging Studies (OASIS): Cross-Sectional MRI Data in Young, Middle Aged, Nondemented, and Demented Older Adults. Journal of Cognitive Neuroscience, 19, 1498-1507. doi:10.1162/jocn.2007.19.9.1498

[7] https://en.wikipedia.org/wiki/Precision_and_recall

[8] Batmanghelich, K.N., Ye, D.H., Pohl, K.M., Taskar, B., Davatzikos, C., 2011. Disease classification and prediction via semi-supervised dimensionality reduction. Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on. IEEE, pp. 1086–1090. http://10.1109/ISBI.2011.5872590

[9] Batmanghelich, K.N., Ye, D.H., Pohl, K.M., Taskar, B., Davatzikos, C., 2011. Disease classification and prediction via semi-supervised dimensionality reduction. Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on. IEEE, pp. 1086–1090. http://10.1109/ISBI.2011.5872590

[10] Ardekani,B.A.,Bachman,A.H.,Figarsky,K.,andSidtis,J.J.(2014).Corpus callosum shape changes in early Alzheimer’s disease: an MRI study using the OASISbraindatabase. BrainStruct.Funct. 219,343–352.doi:10.1007/s00429-013-0503-0

Major_simaran_upasana

ORIGINALITY REPORT

13%	7%	2%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.edureka.co Internet Source	2%
2	www.geeksforgeeks.org Internet Source	2%
3	Submitted to Amity University Student Paper	1%
4	Submitted to Netaji Subhas Institute of Technology Student Paper	1%
5	www.omicsonline.org Internet Source	1%
6	Submitted to University of Balamand Student Paper	1%
7	Submitted to Texas A & M University, Kingville Student Paper	1%
8	www.guru99.com Internet Source	<1%
9	Submitted to Chichester College of Arts,	

	Science and Technology, West Sussex Student Paper	<1%
10	Submitted to Higher Education Commission Pakistan Student Paper	<1%
11	Submitted to Babes-Bolyai University Student Paper	<1%
12	Submitted to Government College of Engineering Aurangabad, Maharashtra State, India Student Paper	<1%
13	worldwidescience.org Internet Source	<1%
14	Submitted to VIT University Student Paper	<1%
15	"Data Engineering and Communication Technology", Springer Science and Business Media LLC, 2020 Publication	<1%
16	digitalcommons.wustl.edu Internet Source	<1%
17	Submitted to Times2 Academy Student Paper	<1%
18	link.springer.com Internet Source	<1%

19	Manohar Swamynathan. "Mastering Machine Learning with Python in Six Steps", Springer Science and Business Media LLC, 2019 Publication	<1%
20	amitrajsantosh.com Internet Source	<1%
21	S. Soundarya, M.S. Sruthi, S. Sathya Bama, S. Kiruthika, J. Dhiyaneswaran. "Early detection of Alzheimer disease using Gadolinium material", Materials Today: Proceedings, 2020 Publication	<1%
22	Submitted to Baltimore City Community College Student Paper	<1%
23	Submitted to Bristow High School Student Paper	<1%
24	Submitted to SVKM International School Student Paper	<1%
25	Submitted to Glasgow Caledonian University Student Paper	<1%
26	blog.sudobits.com Internet Source	<1%
27	Submitted to University of Western Australia Student Paper	<1%
28	Submitted to Taylor’s Education Group Student Paper	<1%

29	www.pssru.ac.uk Internet Source	<1%
30	Submitted to Heriot-Watt University Student Paper	<1%
31	Submitted to University of KwaZulu-Natal Student Paper	<1%
32	Mohamed M. Dessouky, Mohamed A. Elrashidy, Taha E. Taha, Hatem M. Abdelkader. "Computer Aided Diagnosis System Feature Extraction of Alzheimer Disease Using MFCC", International Journal of Intelligent Computing in Medical Sciences & Image Processing, 2015 Publication	<1%
33	jamia.bmj.com Internet Source	<1%
34	www.mitpressjournals.org Internet Source	<1%
35	www.frontiersin.org Internet Source	<1%
36	www.cotocus.com Internet Source	<1%
37	Submitted to University of Central Lancashire Student Paper	<1%
38	Submitted to St. Michael-Albertville High School	

	Student Paper	<1%
39	Submitted to Indiana University Student Paper	<1%
40	Submitted to Yeditepe University Student Paper	<1%
41	Jesia Mathew, Lasitha Mekkayil, Hariharan Ramasangu, B. Ramaswamy Karthikeyan, Aditya G Manjunath. "Robust algorithm for early detection of Alzheimer's disease using multiple feature extractions", 2016 IEEE Annual India Conference (INDICON), 2016 Publication	<1%
42	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
43	Submitted to Eiffel Corporation Student Paper	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: …14/07/2020………………………

Type of Document (Tick): **PhD Thesis** **M.Tech Dissertation/ Report** **B.Tech Project Report√** **Paper**

Name: _Simran Paul,Upasana Bhasin___**Department:** CSE_______ **Enrolment No** ____161352,161347__

Contact No. 977954773,8076229679_E-mail: simran.7.paul@gmail.com,upasanabhasin31@gmail.com

Name of the Supervisor: _____Dr.Rajinder Sandhu____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters) ____EARLY DETECTION OF ALZHEIMER'S DISEASE USING MEDICAL DATA____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

- Total No. of Pages =73
- Total No. of Preliminary pages =9
- Total No. of pages accommodate bibliography/references =2

Simranpaul Bhasin

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at …..13%..(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Abstract & Chapters Details**	
	• All Preliminary Pages • Bibliography/ Images/Quotes • 14 Words String		Word Counts	
			Character Counts	
Report Generated on		**Submission ID**	Page counts	
			File Size	

Checked by
Name & Signature **Librarian**

………

Please send your complete Thesis/Report in (PDF) & DOC (Word File) through your Supervisor/Guide at plagcheck.juit@gmail.com