A PRELIMINARY REPORT ON

Part A: AN APPLICATION TO MONITOR DEMENTIA PATIENTS
Part B: COMPARATIVE ANALYSIS OF MACHINE LEARNING AND DEEP LEARNING ALGORITHMS FOR CLASSIFICATION OF PATIENTS AS DEMENTED OR NON-DEMENTED

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF COMPUTER ENGINEERING

SUBMITTED BY

RUTUJA KAJAVE	Exam No: C22017221303
SHRAVNI SANGAMNERKAR	Exam No: C22017221305
SUSHMA BHOJANE	Exam No: C22016221257
POOJA AMALGIRI	Exam No: C22018222324

DEPARTMENT OF COMPUTER ENGINEERING

2020 -2021

Maharshi Karve Stree Shikshan Samstha's
Cummins College of Engineering for Women, Pune
(An autonomous Institute affiliated to Savitribai Phule Pune University)

CERTIFICATE

This is to certify that the project report entitled

"AN APPLICATION TO MONITOR DEMENTIA PATIENTS"

Submitted by

RUTUJA KAJAVE	Exam No: C22017221303
SHRAVNI SANGAMNERKAR	Exam No: C22017221305
SUSHMA BHOJANE	Exam No: C22016221257
POOJA AMALGIRI	Exam No: C22018222324

are bonafide students of this institute and the work has been carried out by them under the supervision of Prof. **Prakash Gulab Date** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Computer Engineering.**

(Prof. Prakash Date)
Guide,
Department of Computer Engineering

(Dr. Supriya Kelkar)
Head,
Department of Computer Engineering

(Dr. M. B. Khambete)
Principal,
Cummins College of Engineering for Women Pune – 52

Place: Pune
Date: 25th June 2021

ACKNOWLEDGEMENT

Before getting into the thickest of things and digging deep into the project, we would like to thank the personalities who were a part of our projects in numerous ways. I would also like to thank them for providing me a concrete and outstanding support from the birth of our project.

We are extremely thankful for our Principal Mrs. Madhuri Khambete and the Head of Department of Computer Science Mrs. Supriya Kelkar for providing necessary motivation, infrastructure and resources for the accomplishment of our project here at MKSSS's Cummins College of Engineering, Pune. We are highly indebted for their support during the tenure of our project.

We hereby wish to express our deep sense of gratitude to Prof. Mr. Prakash Gulab Date, Department of Computer Science and Engineering, for his esteemed guidance, moral support and invaluable advice for the success of the project. We also wish to thank Prof. Mrs. Madhuri Tasgaonkar who evaluated us throughout our project journey and gave honest and valuable feedbacks on our work and progress.

We are also thankful to the staff members of Computer Science and Engineering Department who have co-operated in making our project a success. We would like to thank all our parents and friends who extended their help, encouragement and moral support either directly or indirectly through the tenure of our project.

Thank you for your valuable guidance and kind support.

Rutuja Kajave
Shravni Sangamnerkar
Pooja Amalgiri
Sushma Bhojane

ABSTRACT

Based on a survey conducted, more than 5 million people are suffering from dementia in India today and many more are expected with a high risk of not getting cured. Dementia robs a person's memories and personality and dignity. Stigmatization, lack of adequate services, and barriers to access available services all worsen the families' physical, psychological and financial hardships. Undoubtedly, taking care of someone suffering from dementia is one of the toughest jobs and families often do it with no meaningful support from any other sources. Most of the user data in such medical organizations are highly unstructured and handling such complex records of data from multiple sources can be difficult to process and interpret into insights. Machine Learning and Deep Learning allows the acquisition and transformation of data into insightful information for analysis and validation purposes and getting a better understanding by visualizing the data that provides classification of historical, current, and predictive views of the operations performed. By interactively changing what data you see and how it's processed, we wish to take the concept a step further by using technology to drill down into charts and graphs for more detail.

Keywords: Deep Learning, Machine Learning, Android, Firebase, Flutter, Dementia.

TABLE OF CONTENTS

LIST OF ABBREVATIONS

ABBREVIATIONS	ILLUSTRATION
GUI	Graphic User Interface
DBMS	Database Management System
App	Application
EMR	Electronic Medical Record
UML	Unified Modelling Language
ERD	Entity Relationship Diagram
EHR	Electronic Health Record
PHN	Patient Health Number
SRS	System Requirement Specification
CRUD	Create Read Update Delete
SDK	Software Development Kit
RAM	Random Access Memory
IDE	Integrated Development Environment
API	Application Program Interface
DFD	Data Flow Diagram
ML	Machine Learning
PHM	Population Health Management
CM	Case Management

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

About Dementia

Dementia is a prevalent term to represent a group of symptoms that occurs owing to the damage and death of brain cells. The familiar symptoms observed are impaired memory, judgment, thinking, decision making, orientation, language, impairment in cognitive functions, memory, judgment, thinking, decision making, orientation, language that is occasionally preceded by deterioration in emotional control, social behavior, or motivation.

According to a survey conducted, illiterate people are three times more likely to suffer from dementia later in their lives as compared to literate people since the latter can engage in activities that stimulate the brain, which help build up cognitive reserve, which is the mind's resistance to the brain getting damaged.

Dementia is generally not identified as a health condition and even after identification the caretakers, in maximum cases being the family members do not receive long-term support. Due to the fragmented nature of institutional care, even today the predominance of patients is treated at home. It is strenuous to influence practice, introduce reforms or provide training in the private health care center. Even today on the whole it is still considered to be culturally unacceptable.

About Alzheimer

Alzheimer's is the most common type of dementia that causes problems with memory, thinking and behavior. Symptoms tend to grow slowly and worsen over time, becoming severe enough to disrupt daily activities. Those with Alzheimer’s live an average of eight years after their symptoms appear in others, but survival can be between four and 20 years, depending on age and other health conditions.

About 200,000 Americans under the age of 65 have Alzheimer's early set and is the sixth leading cause of death in United States as per reports.

Alzheimer's does not have a current treatment, but symptomatic treatment is available and research is ongoing. While current treatment for Alzheimer's will not stop Alzheimer's from continuing, it can temporarily reduce the increase in symptoms and improve the quality of life of those with Alzheimer's and their caregivers. Today, there is an ongoing worldwide effort to find better ways to treat the disease, to delay its onset, and to prevent its spread.

Dementia and Alzheimer's both mainly affects older people but it isn't a normal part of aging. Though rarely, dementia could affect young people. Also, Alzheimer's is not just a disease of aging. There are people under the age of 65 suffering from the latter which is also known as Alzheimer's early set.

In a continuous care operation, treating and caring for patients with Dementia has always been an arduous task, for both the hospital staff as well as the patient's caretakers. Therefore, it is critical to document each patient's condition and history of care to avoid any negative repercussions.

1. Motivation

The use of manual mode of managing medical records in the healthcare institution at Satyak: Assisted Living Centre, Baner, Pune has been a dominant challenge. This dismally affects the accuracy, relevance, timeliness, confidentiality of the data, easy management of records and timeous access during healthcare service delivery.

As per the records, even today many patients suffering from Dementia or Alzheimer has no access to the correct resources, care centers and guidance for correct line of treatment.

To detect these diseases at early stages on the basis symptoms, there's a need to validate the data and classify them as demented and non-demented.

2. Problem Definition

Part A

Treating and caring for patients with Dementia has always been an arduous task, for both the hospital staff as well as the patients' caretakers. The proposed system provides a convenient and quick view to help facilitate the patient onboarding, maintaining patient health history, monitor the medication as well as the health of the patients and maintain the inhouse activities and records.

Part B

To validate the data by using Machine Learning and Deep Learning Algorithms and classify the patients under the category, demented and not demented using open-source database.

CHAPTER 2: LITERATURE SURVEY

2.1 Background of Domain

1. Android
2. Firebase: Database Manager and Cloud Technology
3. Flutter and Dart
4. Machine Learning
5. Deep Learning

2.2 Comparisons: Research Papers and Current applications in the market

Research Paper	Keywords	Supported Platforms	Limitations of Current health system	Advantages over Current health system
ECM System implementation readiness to improve medical record management in Limpopo Province, South Africa	ECM, ERM, Healthcare Service	Windows, MAC iOS	1.Don't have a collaborative system to integrate the records management responsibility. 2.Incapable of capturing medical history of the patients, tracking paper-based records movement or creation but capture only personal details and billing data.	1.One record accessed by many people at the same time. 2.Provision of timely, accurate, trustworthy and complete records, effective records security throughout the life span. 3.Compliance with legislative framework. 4.Creation of reliable knowledge at all stages of the life span.

Utilizing Open ERP for creating medical record management system in Smart Hospital	ERP, EHR, Healthcare Service	Desktop Application	1.Financial issues. 2.Temporary loss of productivity associated with EHR adoption , privacy and security concerns and several unintended consequences.	1.Quality of care. 2.Reduction in medical errors, and other improvements in patient-level measures that describe the appropriateness of care.
Design and development of a web-based hospital Information System [1998]	Computer-based Patient Record (CPR)	Desktop Application	1.Reduction in access 2.Reduction in the rate of hospital admissi ons 3. Increasing employees' workload and dissatisfaction.	1.Improved Medical Processes. 2. Digital medical records. 3.Increase in Staff Interaction & Facility management. 4.Market strategy

Table 2.1 Research Papers Comparison

Dr Pad Application	Medical Records Application (Medclin)	Medical Records Application (Yerokhin Vldimir)
Create and keep track of patient's appointments, visit history, and medical records.	Manages appointments, Tracks Medical history reports and information history, etc.	Record-keeping of all visits to doctors, appointments, recommendations, referrals to lab tests.
Easy to use the mobile app, elegantly designed for phone and tablet.	Supports multiple screens, phones, small and large size tablets. Works on Chromebook system.	User-friendly and intuitive interface supports all mobile devices.

Attach Medical documents (audio, video, and image) in PDF format only. (Number depends on free, paid, cloud plan)	Attach medical documents of any type (pdf, word, etc.) or capture them using the camera or video recording. It also exports medical data to Excel sheets, pdf and graphs.	Enclosing of separate medical documents (files of all formats) in the form of attachment to a medical visit/analysis.
Autocomplete medication once saved initially.	Most of the data are stored using the auto-complete technique.	Data isn't stored using the auto-complete technique.
No Search technique is used in this app.	Multiple Search Technique by name, phone no, visit date, appointment date, etc.	No Search technique is used in this app.
Records are maintained through texts.	Records video or image capture for medical activities, with or instead of writing text.	Records are maintained through texts.
No Such feature is a part of this application	The availability to launch Google Maps navigation to drive to the (patient & doctor & hospital) stored address starting from your current location.	No Such feature is a part of this application.
Store clinical data in unlimited secured cloud storage. No separate cloud account or setting is required. Cross-user and cross device syncing available.	According to needs, choose to store your medical data either in your device memory or on secure cloud storage where data sync is enabled.	Synchronization of data, safe data storage on the Google servers.
A feature of switching to the visiting clinic and accessing the clinical data is added if the doctor is visiting many clinics, and access to the clinical data is provided with an invitation.	Could be used for private practice management as a clinical information system, clinic management system, Doctor Patient Medical Record, Healthcare management mobile application, Patient Medical Records to track Patient history, EMR, EHR.	No such feature is a part of this application.

No such feature is a part of this application.	The capability of adding patient information via the device contact list; if the patient info is on the device contacts list.	No such feature is a part of this application.
No such feature is a part of this application.	Username and password authentication.	No such feature is a part of this application.

Table 2.2 Current Applications Comparison

Name of Application	Supported Platforms	End Users	User Support	Subscription	Categories
Dr Chrono (2009)	SaaS, iOS	Medical practices (all types and sizes)	Business Hours, Online	NA	EMR, E-Prescription Billing, Patient Portal Claim Processing. Chiropractic, Medical Spa, Optometry, Dermatology, Occupational & Physical Therapy, Plastic Surgery, Telemedicine.
iCare (2012)	SaaS, iOS, Android	Health Systems, Hospitals, Clinics	Online, 24/7 Live Support	Free	E-Prescription, EMR, Billing, Appointment Scheduling, Nursing Home &Patient Portal. Pharmacy, PHM, Telemedicine.
Simple Practise (2012)	SaaS, iOS, Android	Solo & group private practitioners	Business Hours, Online	Paid, Free Version available with limited features	EMR, Billing, Appointment Scheduling, Patient Portal. Case Management, Physical & Speech Therapy

Thera Nest (2012)	SaaS, iOS, Android	Individual practitioners, family counseling centers, mental health agencies, social service organizations, faith-based groups, psychology practices, schools, universities & large therapy practices.	Business Hours, Online	Paid Only	EMR, HIPAA Compliance, Billing, Appointment Scheduling, CM, Patient Portal. Human Services, Telemedicine

Table 2.3: Current Applications Comparison

CHAPTER 3: PROJECT REQUIREMENTS

3.1. Problem Statement/ Objective

Part A

1. To curate the registration form digitally which till now was filled manually. Since the searching process is too tedious and long when it comes to manual records. Turning a large number of pages including patient's personal details, medical history, progress, medication, etc. is not practically feasible. The chances of losing or misusing the information are also a possibility.
2. To auto summarize the data based on scales and scoring system used to understand the patient's activities of everyday life. Therefore, providing a clear classification of the three levels of Dementia Patients.
3. To analyze the progress of patients' assessments using dashboard module.
4. To activate the camera of the android devices to record the investigations and other details.
5. To highlight notes placed by all the doctors working on the patients for e.g., physicians, physiotherapists, cardiologists, psychiatrists, psychologists etc. At times a progressive note is not placed, verbal orders are passed, operative notes on the charts are missing the instant after the procedure is performed, date and time aren't entered. As a result, when the hospital is called for the patient's status or the next doctor visits for follow-up, there is no way to find out unless there is a written post-checkup note. Also, it is time-consuming and difficult to take the previous updates made by the other doctors on call every time.

Part B

1. To validate the data using an opensource database of dementia/ Alzheimer patients. Machine learning and deep learning algorithms are used to classify the patients as demented and non-demented based on the symptoms, occurrences and predictive analysis.

3.2. SRS (System Requirement Specification)

3.2.1. Features

Part A

1. Login Authentication: There are three logins in the system. The data is encapsulated based on the inheritance.
    a) Admin consists of the psychiatrists and the medical officer. They are given access to CRUD operations performed on the complete patient data.
    b) Nursing care and the Specialists consists of all the nursing staff and all the other doctors working on a patient. They are given access to their respective domain assessments and can perform CRUD operations on those only. For e.g., Nursing Assessment, Physiotherapy Assessment, Social Workers, Psychology Assessment etc.
    c) Legal Guardian consists of the registered relatives of the patient admitted. They are given access to view the data only. They can view the monthly report of the patient and the photo gallery updated on a regular interval.

2. Dashboard: The Assessments performed on the patients are broadly classified under two categories namely, Nursing Assessments and Doctor's Assessments.
    a) Nursing Assessments consists chart, dietary chart, activities of daily life etc.
    b) Doctors Assessments consists of physical and cognitive assessments.

These assessments are evaluated on daily, weekly and monthly basis and the analysis of the progress is represented in the form of real-time dashboards.

Part B

1. Classification Feature: Using deep learning and machine learning algorithms the data is classified as demented and non-demented on the basis of symptoms, occurrences etc. These algorithms are also compared to analyze the accuracy for each.

3.2.2. Functional Requirements

1. Registration:
    - **a)** System allows the use cases i.e., the legal guardian, nursing care and the specialists to add the details. After this step, all the documents required for record purpose should be uploaded. Once this is done the use case can successfully login into the system.
    - **b)** When the patient is successfully added into the system, he/she is provided with a unique PHN. This will be used as the primary key throughout his stay at the dementia care center by all the use cases.
    - **c)** At the time of discharge, the PHN is deleted from the system but the data associated with it isn't lost. Therefore, in case of readmission in a restricted time frame the data could be retrieved for any medical correlation. This will eliminate the need to conduct all the investigations, history, medical highlights again from scratch. The time and cost factor will also reduce.

2. Medical Record Management: The system feeds in data that includes medical assessments, investigations, calculations and evaluation of progress, report and prescriptions etc. The CRUD operations are performed by Admins, Nurses and Specialist. The data is encapsulated allowing only restricted use cases to access and edit data.

3. Database:
    a) The database will contain the data entries of the uses cases. Mandatory details must be filled by all the uses cases without which login into the system isn't possible.

b) Primary key for each entity is pre-defined is used to search the use case. The patient records will be searched using PHN.
c) The system will allow only restricted use cases to edit and update the medical records.

3.2.3. Non-Functional Requirements

1. Performance:
    a) Response time: The system will generate response in 1 second after verifying the patient's details.
    b) Capacity: The system must support 1000 people at a time.
    c) User Interface: The user-interface screen will generate a response within 5 seconds.
    d) Conformity: The System ought to conform Microsoft Accessibility guidelines.
    e) Availability: The system shall be available all time.
    f) Backup: The system will be capable of backing the data on cloud.
    g) Errors: The system shall keep a log of all the errors.
    h) Password Management: The system shall generate strong passwords.
    i) Regular Database archiving.

2. Safety: The negative effects of common errors on the system performed by human must be limited. For instance, if the medical data/ record is deleted by any other use cases other than the admin, the intent should be confirmed and an option to undo the unnecessary changes made should be provided.

3. Security:
    a) Patient Record identification will be conducted only via PHN.
    b) Login Authentication
    c) Any change performed on the medical record of a patient (CRUD operations) should be encapsulated and only use case domain specific data should be made visible.

d) The medical records in the database are protected using multiple layers of protection.

4. Software quality Attributes: The software used in the proposed system is user friendly and the following attributes are taken into account considering the quality aspect:
    a) Accuracy
    b) Reliability
    c) Security
    d) Efficiency
    e) Compatibility

3.2.4. External Interface Requirements

1. User Interfaces: The software provides user friendly and readable GUI. Use Cases are able to perform the required task.
2. Software Interfaces:
    a) Google Colab
    b) Dart
3. Communication Interfaces:
    a) Android Application
    b) Python Application

3.2.5. System Requirements

1. Database: Firebase
2. Software:
    a) Android Studio via SDK to develop GUI
    b) Flutter and Dart to create aesthetic user interface.
    c) Git for version control
    d) VsCode Editor
    e) Google Colab

f) Firebase

3. Hardware: All the components should be able to execute on personal mobile devices with android OS platform. For developing the application on laptop/PC with following specifications are required:
    a) Operating System: Windows, Linux.
    b) Hard Disk: 40GB
    c) RAM: 256 GB
    d) Processor: Intel core i3 7th Generation
4. Technology:
    a) Platform: Android OS/ iOS
    b) Technology: Flutter Dart SDK
    c) Language: Dart, Python
    d) IDE: VS Code
    e) Framework: Flutter Framework

3.3. Use Cases

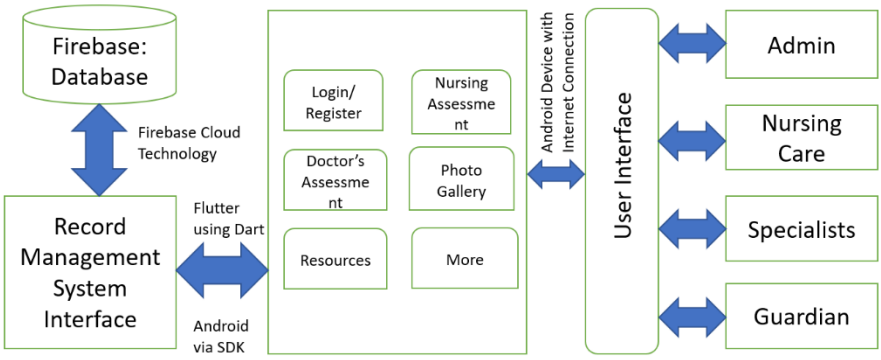
1. Admin: This consists of the Psychiatrist and the Medical Officer. They have access to the entire data. Also, they can perform CRUD operation on the complete data.

2. Nursing Care and Specialists: This consists of the entire nursing staff and all the specialists who are the visiting doctors for the patients. Specialists includes Physicians, Psychologists, Physiotherapist's etc. The nursing care has access and can perform CRUD operations on Nursing Assessments only. The specialists have an access to their domain specific assessments and the nursing data.

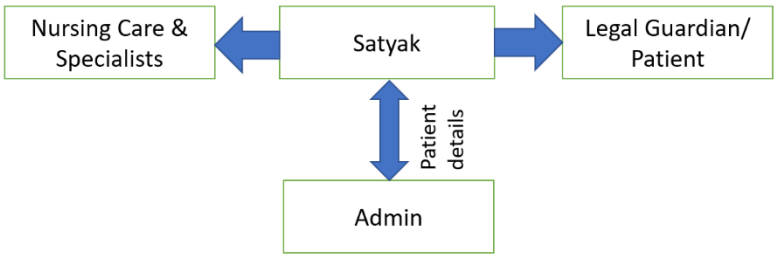
3. Legal Guardian: The registered legal guardian of the patient can only view their patient's data which consists of their monthly medical reports and progress. Along with that they can also view the photo gallery that consists of medical investigations, if any and the photos of all the activities, festivals, daily routine conducted at the center.

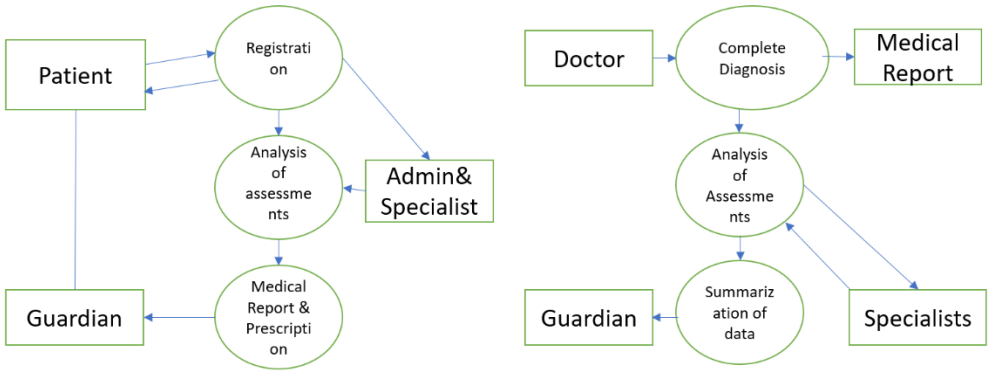
CHAPTER 4: SYSTEM DESIGN

4.1. System Architecture Diagram

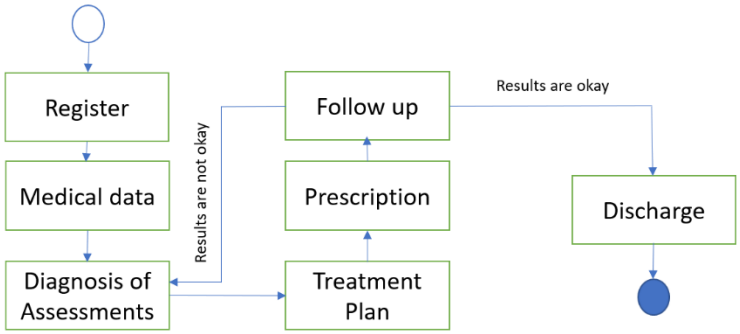


4.2. Data Flow Diagram

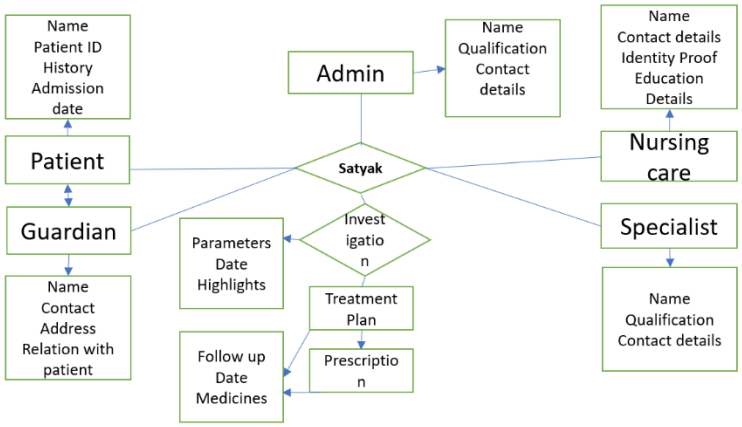




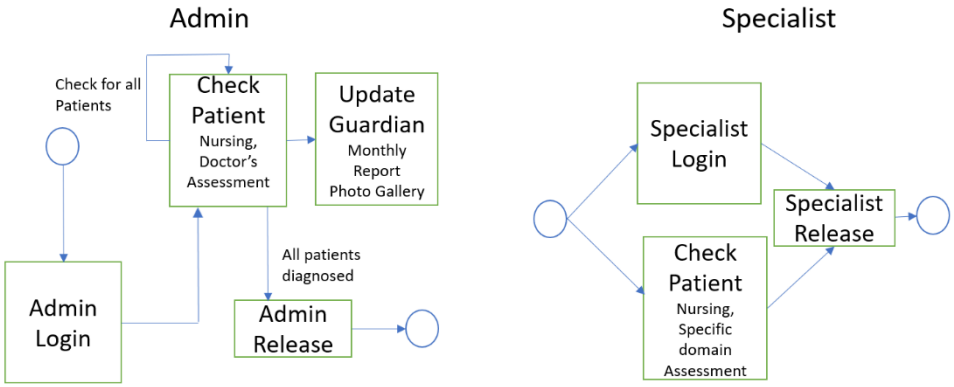
4.3. State Diagram

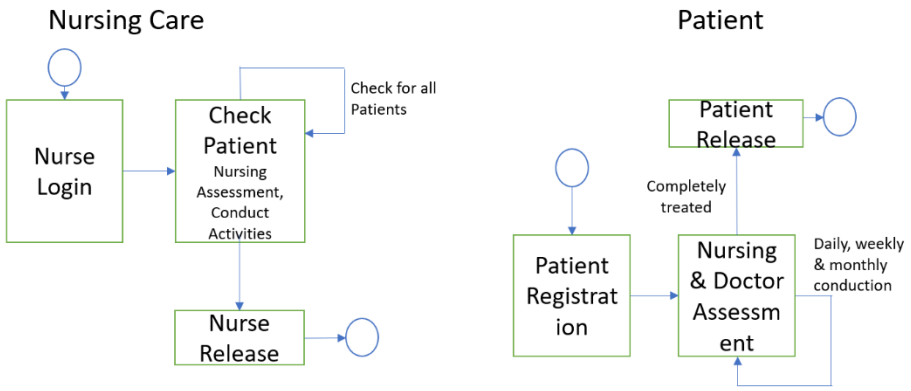


4.4. Entity Relationship Diagram

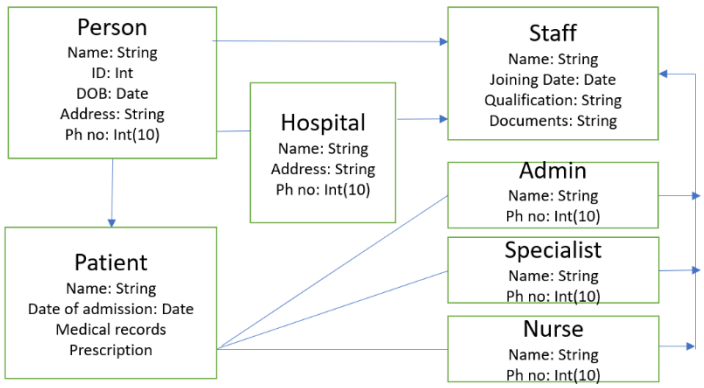


4.5. UML Diagram

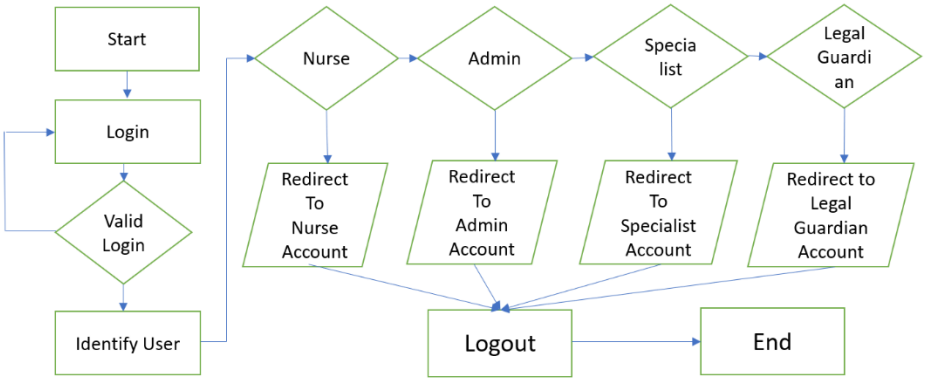




4.6. Class Organization Diagram



4.7. Workflow of the Application



CHAPTER 5: TECHNOLOGIES

5.1. Tools and Technologies

5.1.1. Android Studio:

a) Android Studio is the official IDE for Android application development. It is based on the IntelliJ IDEA, a Java IDE for software, and incorporates its code editing and developer tools.

b) To support application development within the Android operating system, Android Studio uses a Gradle-based build system, emulator, code templates, and GitHub integration. Every project in Android Studio has one or more modalities with source code and resource files. These modalities include Android app modules, Library modules, and Google App Engine modules.

c) Android Studio uses an Instant Push feature to push code and resource changes to a running application. A code editor assists the developer with writing code and offering code completion, refraction, and analysis. Applications built in Android Studio are then compiled into the APK format for submission to the Google Play Store.

d) The software was first announced at Google I/O in May 2013, and the first stable build was released in December 2014. Android Studio is available for Mac, Windows, and Linux desktop platforms. It replaced Eclipse Android Development Tools (ADT) as the primary IDE for Android application development. Android Studio and the Software Development Kit can be downloaded directly from Google.

5.1.2. VS Code:

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other

languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

5.1.3. Flutter:

a) Flutter is a mobile app SDK for building high-performance, high-fidelity, apps for iOS and Android, from a single codebase.
b) The goal is to enable developers to deliver high-performance apps that feel natural on different platforms. We embrace differences in scrolling behaviors, typography, icons, and more.
c) No mobile development experience is required to get started. Apps are written in Dart, which looks familiar if you've used a language like Java or JavaScript. Experience with object-oriented languages is definitely helpful, but even non-programmers have made Flutter apps!

5.1.4. Dart:

a) Dart is an object-oriented language with C-style syntax which can optionally trans compile into JavaScript. It supports a varied range of programming aids like interfaces, classes, collections, generics, and optional typing.
b) Dart can be extensively used to create single-page applications. Single-page applications apply only to websites and web applications. Single-page applications enable navigation between different screens of the website without loading a different webpage in the browser. A classic example is Gmail ─ when you click on a message in your inbox, browser stays on the same webpage, but JavaScript code hides the inbox and brings the message body on screen.

5.1.5. Firebase: Database

a) The Firebase Realtime Database is a cloud-hosted NoSQL database allows one to store and sync data between your users in real time.

b) Realtime syncing makes it easy for the users to access their data from any device i.e., web or mobile, and it helps the users to collaborate across devices with one another in ease.

c) Realtime Database ships with mobile and web SDKs so the user can build apps without the need of servers. One can also execute backend code that responds to events triggered by your database using Cloud Functions for Firebase.

d) It optimizes the use when the users go offline. The Realtime Database SDKs use local cache on the device to serve and store changes. When the device comes online, the local data is automatically synchronized.

e) The Realtime Database integrates with Firebase Authentication to provide simple and intuitive authentication for developers. Declarative security model allows access based on user identity or with pattern matching on your data. Therefore, providing a strong user-based security.

f) Cloud Functions are single-purpose JavaScript functions that are executed in a secure, managed Node.js environment. They are only executed when a specific event being watched is emitted. One can run the mobile backend code without managing servers.

g) Deploying the code to firebase servers requires just one command. After that, Cloud Functions automatically scales up computing resources to match the usage patterns of our app. Therefore, keeping the maintenance low without worrying about SSH credentials, server configuration, provisioning new servers, or decommissioning old ones.

h) Application logic is best controlled on the server to avoid tampering on the client side. Cloud Functions is fully insulated from the client. Therefore, keeping the functions and logic private and secure that can’t be reverse engineered.

5.1.6. Google Colab

a) Colab is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

b) Colab notebooks are stored in Google Drive or can be loaded from GitHub. Colab notebooks can be shared just as Google Docs or Sheets.

5.1.7. Python:

a) It is an extensible, portable, interpreted, integrated, developer friendly and high-level programming language.

b) Python is very easy to learn and code in python language as compared to other languages like C, C#, JavaScript, Java, etc.

c) It is an Object-Oriented programming that supports object-oriented language and concepts of classes, objects encapsulation, etc.

d) One can write python code into C or C++ language and also compile that code in C/C++ language.

e) Python has a large standard library which provides a rich set of module and functions and a dynamically typed language.

CHAPTER 6: TESTING

The purpose of testing is to discover errors in our application. Testing is the process of trying to discover every conceivable fault or weakness in our final product before deploying it in the market. It provides a way to check the functionality of components, sub-assemblies, assemblies or a finished product. It is a process of exercising software with an intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests done before the actual deployment of a product. Each test type addresses a specific testing requirement.

6.1 Types of Testing Implemented in our System

1. Unit Testing:
    a) It involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs.
    b) All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration.
    c) This is a structural testing, that relies on knowledge of its construction and is invasive.
    d) Unit tests perform basic tests at component level and test a specific business process, application or system configuration.
    e) Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected outputs.
2. Functional Testing:
    a) It provides systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation and user manuals.

b) Functional testing is centered on following items:

1. Valid Input: Identified classes of valid input must be accepted.
2. Invalid Input: Identified classes of invalid input must be rejected.
3. Functions: Identified functions must be exercised.
4. Output: Identified classes of application output must be exercised.
5. Systems/Procedures: Interfacing systems or procedures must be invoked.

3. Regression Testing:
   a) Regression Testing is purely a repetitive testing process in which previously executed test scenarios are re-executed when code changes have been implemented.
   b) The purpose is to verify if code change introduces issues/defects into the existing functionality. There are so many kinds of possible changes that can impact the existing functionality in an application system. It aims at validating how newly modified code meets the specified requirements and to ensure that existing code has not been affected by the changes made.
   c) Even the simplest change to the code could impact previously tested functionality.
   d) Sometimes it is quite difficult for a developer to figure out how a change in one part of the software application will affect other parts of the software program. Retesting the changes to the application will ensure data integrity and proper validation after the fixes have been made. This way, newly implemented code will be validated along with the existing code, ensuring the system's functionality is properly verified.

Test Strategy and approach: Field testing must be performed manually and functional tests must be written in detail.

6.2. Test Objectives:

a) All field entries must work properly.

b) The entry screen, messages and responses must not be delayed.

6.3. Features to be tested:

a) Verify that all the entries are of correct format.

b) No duplicate entries should be allowed.

c) All transition buttons must take the user to the correct page.

6.4. **Test Cases:**

Table 6.4.1: Test cases: Login Page

Functional Test Cases	Type of Test
Verify if a user will be able to login with a valid username and valid password	Positive
Verify if a user cannot login with a valid username and an invalid password.	Negative
Verify the login page for both, when the field is blank and Submit button is clicked	Negative
Verify the 'Forgot Password' functionality.	Positive
Verify the messages for invalid login.	Positive
Verify if the data in password field is either visible as asterisk or bullet signs.	Positive
Verify if a user is able to login with a new password only after he/she has changed the password	Positive

Verify if the ‘Enter’ key of the keyboard is working correctly on the login page.	Positive
Verify the time taken to log in with a valid username and password.	Performance and Positive Testing
Verify if the font, text color, and color coding of the Login page is as per the standard.	UI Testing and Positive Testing
Verify if there is a ‘Cancel’ button available to erase the entered text.	Usability Testing

Table 6.4.2: Test cases: Registration Page

Test Cases	Feature	Description	Steps to Execute	Test Data / Input	Expected Results
TC -001	User Interface	Check all: 1.Text Boxes 2.Radio Buttons	Click on: 1.Buttons, Radio Buttons 2.Dropdowns	N/A	UI should work perfect.
TC -002	Required Fields	Check the required fields by not filling any data	1.Don’t enter any value in the field. 2.Click on: Register button	N/A	It must show a mandatory symbol “*” on Mandatory fields.

TC -003	Required Fields	Check that user registers by filling all required fields	1.Enter valid values in the required field. 2. Click on registration button.	N/A	1.User should be registered successfully. 2. A message should be shown for successful registration 3.Mail should be sent to the user.
TC -004	Optional Fields	Check all the optional fields when data isn't filled.	1. Do not enter any detail in optional fields 2. Enter valid data in required fields 3. Click on the Signup button	N/A	1. It should not ask to fill the optional fields 2. User should be registered successfully 3. A successful registration message should show 4. Mail should send to the user
TC -005	Optional Fields	Check all the optional fields by filling all the data	1. Enter valid data in optional fields 2. Enter valid data in required fields 3. Click on the Register button	N/A	1. User should be registered successfully 2. A successful registration message should show 3. Mail should send to the user
TC -007	Email Validation	Check all valid fields.	1. Enter valid Emails 2. Click on the Register Button.	1.test.22@gmail.com 2.test@gmail.com	It should not show any validation message

TC -008	Phone No Validation	Check the phone number when passing alphanumeric data	1. Enter alphanumeric data in phone field 2. Click on Register button	dada5$7567#7	It should show the validation message 8 for Phone Number
TC -009	Phone No Validation	Check the phone number when not pass country code	1. Enter valid phone number without country code 2.Click on Register button	9012078654	It should show the validation message for country code is required
TC -010	Phone No Validation	Check the phone number when passing country code	1. Enter valid phone number with country code 2.Click on Register button	+9190112244	It shouldn't show any validation message
TC -011	Password Validation	Check the password limit when enter value less than min	1. Enter value which is alphanumeric but less than 8. 2.Click on Register button	Password	It should show validation message
TC -012	Password Validation	Check the password limit when enter value greater than max	1. Enter alphanumeric value but more than 32. 2.Click on Register button	Any Random string with numbers	It should show validation message

TC -013	Password Validation	Check the password when passing only numbers	1. Enter a value in numbers which is in between 8-32 2.Click on Register button	12345678	It should show validation message
TC -014	Password Validation	Check the password when passing valid data	1. Enter value in alphanumeric which is in between 8-32 2.Click on Register button	Pass123456	It should not show any validation message
TC -015	Required Fields	Verify if blank spaces are passed in required fields.	1. Go to the Site 2. Passed blank spaces in required fields. 3. Click on the Register button	N/A	Those Blank spaces should trim and Validation error message for required fields should visible.
TC -016	Required Fields	Verify user can verify its Email ID	1. Go to the Email 2. Click on the verification link	test22@gmail.com	User should get a verification link and able to verify his/her Email ID
TC -017	Phone No Validation	Verify if the length of the phone number is incorrect i.e., less than 10.	1. Enter phone number less than 10 digits. 2.Enter all required fields. 3.Click on Register Button	91901122	It should show the validation error message for phone number length.

TC -018	Phone No Validation	Verify if the length of the phone number is incorrect i.e., more than 10	1.Enter phone number less than 10 digits. 2.Enter all required fields 3.Click on Register Button	19011E+13	It should show the validation error message for phone number length.
TC -019	Password Validation	Verify if the password required rules are not satisfied in the password	1. Enter the password which not satisfies the required rule 2. Click on Register button	passw	It should display error with required rules for password value like it

CHAPTER 7: IMPLEMENATATION ASPECTS

Part B:

Classification of Patients as Demented and Non-Demented using Machine Learning and Deep Learning Algorithms

7.1. Dataset Description:

a) We will be using the longitudinal MRI data of 150 subjects aged 60 to 96.
b) Each subject was scanned at least once.
c) Everyone is right-handed.
d) 72 of the subjects were grouped as ‘Non-demented’ and 64 of the subjects were grouped as 'Demented' at the time of their initial visits and remained so throughout the study.
e) 4 subjects were grouped as ‘Non-demented' at the time of their initial visit and were subsequently characterized as 'Demented' at a later visit. These fall under the 'Converted' category.

Table 7.1 Data Description

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

7.1.1. Mini–Mental State Examination (MMSE):

a) The Mini–Mental State Examination (MMSE) or Folstein test is a 30-point questionnaire commonly used in medicine and allied health to screen for dementia.

b) Any score greater than or equal to 24 points (out of 30) indicates a normal cognition. Below this, scores can indicate severe (≤9 points), moderate (10– 18 points) or mild (19–23 points) cognitive impairment.

c) The raw score may also need to be corrected for educational attainment and age. That is, a maximal score of 30 points can never rule out dementia.

Table 7.1.1.1MMSE Score Description

Method	Score	Interpretation
Single Cutoff	<24	Abnormal Range
	<21	Increased Odds of Dementia
	<25	Decreased Odds of Dementia
Education	21	Abnormal for 8th Grade Education
	<23	Abnormal for High School Education
	<24	Abnormal for College Education
Severity	24-30	No Cognitive Impairment
	18-23	Mild Cognitive Impairment
	0-17	Severe Cognitive Impairment

7.1.2. Clinical Dementia Rating (CDR):

a) The CDR is a 5-point scale used to characterize six domains of cognitive and functional performance applicable to Alzheimer disease and related dementias: Memory, Orientation, Judgment & Problem Solving, Community Affairs, Home & Hobbies, and Personal Care.

b) This score is useful for characterizing and tracking a patient's level of impairment/ dementia:

Table 7.1.2 CDR Score Description

Score	Description
0	Normal
0.5	Very Mild Dementia
1	Mild Dementia
2	Moderate Dementia
3	Severe Dementia

7.1.3. Estimated Total Intracranial Volume(eTIV):

a) Total intracranial volume (TIV / ICV) is an important covariate for volumetric analysis of brain and brain regions, especially in the study of non-neurodegenerative diseases, where it can provide a representative of premortem brain mass.

b) Unlike brain atrophy in patients with AD, TIV did not differ over time. TIV means that it is not very different between any topic groups. There was no association between TIV and age or age at the onset of the disease. The only important prediction for TIV was sex. Men showed up to ~12% greater eTIV than women.

c) We measured TIV through a separate isolation procedure on T1- and T2-weighted MR images in 55 controls, 10 AD patients, and two people at risk for family AD. Whole brain values were also measured and adjusted for TIVs.

d) TIV's prevalence of short-term brain rates significantly reduced individual variability; the coefficient of variation (CV) was reduced from 10.0% to 6.0% in controls (P <.001). TIVs measured in T1-weighted images had a lower variability (CV, 0.16%) and were not significantly different from those measured in T2-weighted images (P = .16). TIV normalization of brain volume measurement reduced contact variability caused by voxel-scaling differences (CV was reduced from 1.3% to 0.5%, P = .002) in 10 controls and five AD patients. 1.3% to 0.5%, P =.002) in 10 controls and five AD patients.

7.1.4. Atlas Scaling Factor (ASF):

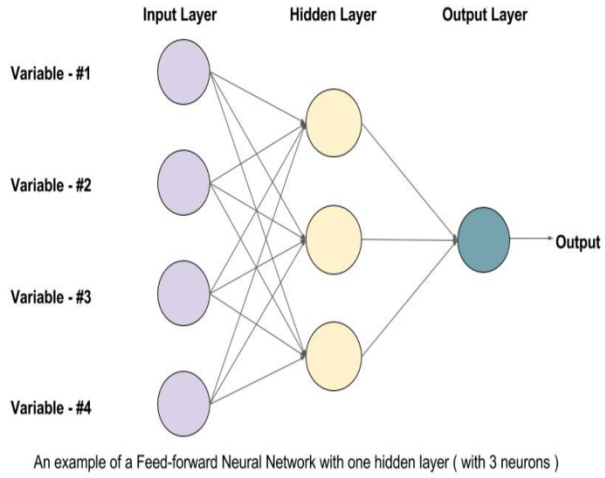
a) A unified approach for morphometric and functional data analysis in young, old, and demented adults using automated atlas-based head size normalization: reliability and validation against manual measurement of total intracranial volume.

7.2. Deep Learning and Machine Learning Algorithms:

7.2.1.. Simple Feed-Forward Neural network:

a) They are also called deep networks, multi-layer perceptron (MLP), or simply neural networks. As data travels through the network's artificial mesh, each layer processes an aspect of the data, filters outliers, spots familiar entities and produces the final output.
b) Feedforward neural networks are made up of the following:

Fig. 7.2.1.1. Simple Feed Forward Neural Network



c) Neural network computes the data in three simple steps:

1. Multiplication of weights and inputs: The input is multiplied by the assigned weight values.

2. Adding the biases: In the next step, the product found in the previous step is added to their respective biases.

3. Activation: An activation function is the mapping of summed weighted input to the output of the neuron. It is called an activation/transfer function because it governs the inception at which the neuron is activated and the strength of the output signal.

4. Output signal: Finally, the weighted sum obtained is turned into an output signal by feeding the weighted sum into an activation function (also called transfer function). Since the weighted sum in our example is greater than 20, the perceptron predicts it to be a rainy day. There are several activation functions for different use cases. The most commonly used activation functions are relu, tanh and SoftMax.

d) Calculating the Loss:

1. A loss function quantifies how good or bad a given model is in classifying the input data. In most learning networks, the loss is calculated as the difference between the actual output and the predicted output.

2. The function that is used to compute this error is known as loss function J(.). Different loss functions will return different errors for the same prediction, having a considerable effect on the performance of the model.

e) Gradient Descent:
   1. It is the most popular optimization technique for feedforward neural networks. The term "gradient" refers to the quantity change of output obtained from a neural network when the inputs change a little.
   2. It measures the updated weights concerning the change in error. The gradient can also be defined as the slope of a function. The higher the angle, the steeper the slope and the faster a model can learn.

f) Backpropagation:
   1. The predicted value of the network is compared to the expected output, and an error is calculated using a function.
   2. This error is then propagated back within the whole network, one layer at a time, and the weights are updated according to the value that they contributed to the error. This clever bit of math is called a backpropagation algorithm.
   3. The process is repeated for all of the examples in the training data. One round of updating the network for the entire training dataset is called an epoch. A network may be trained for tens, hundreds or many thousands of epochs.

Fig. 7.2.1.2. Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

7.2.2. SVM

a) 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.

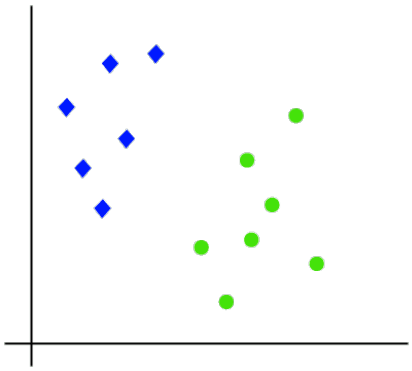
b) The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

c) SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the 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:

d) SVM Working:

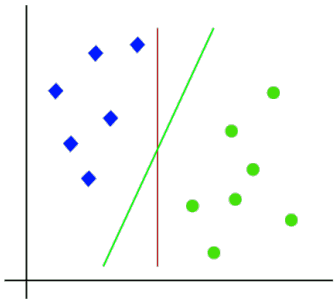
1. Linear SVM: The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair (x1, x2) of coordinates in either green or blue. Consider the below image:

Fig. 7.2.2.1. Linear SVM - I



So, as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:

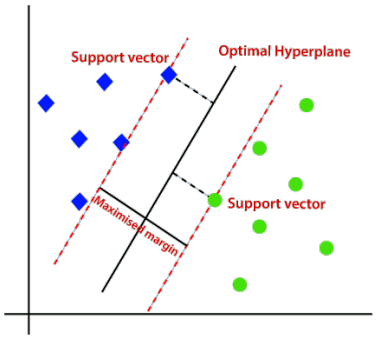
Fig. 7.2.2.2. Linear SVM - II



Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as margin. And the goal of

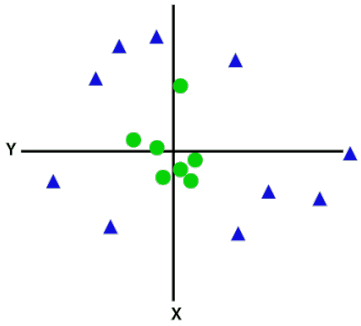
SVM is to maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.

Fig. 7.2.2.3. Linear SVM - III



2. Non-Linear SVM: If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below:

Fig. 7.2.2.4. Non-Linear SVM - I



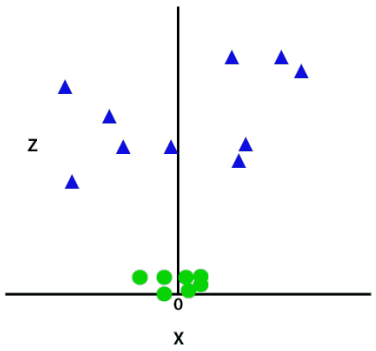
So, to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third-dimension z. It can be calculated as:

$z=x^2 +y^2$

By adding the third dimension, the sample space will become as below image:

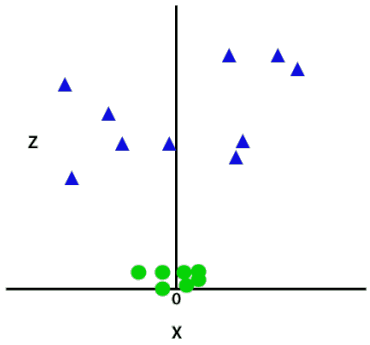
Fig. 7.2.2.4. Non-Linear SVM - I

Fig. 7.2.2.4. Non-Linear SVM - II



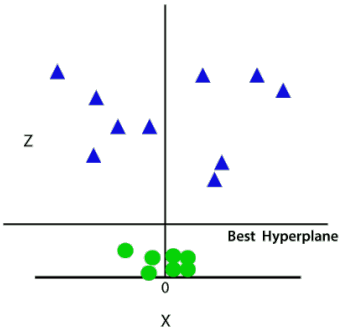
By adding the third dimension, the sample space will become as below image:

Fig. 7.2.2.4 Non-Linear SVM - III



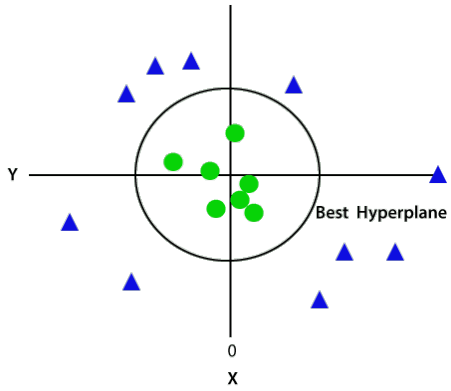
So now, SVM will divide the datasets into classes in the following way. Consider the below image:

Fig. 7.2.2.4. Non-Linear SVM - IV



Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:

Fig. 7.2.2.4. Non-Linear SVM - V



Hence, we get a circumference of radius 1 in case of non-linear data.

3. SVM Advantages:

    1. SVM's are very good when we have no idea on the data.
    2. Works well with even unstructured and semi structured data like text, Images and trees.
    3. The kernel trick is the real strength of SVM. With an appropriate kernel function, we can solve any complex problem.
    4. Unlike in neural networks, SVM is not solved for local optima.

5. It scales relatively well to high dimensional data.
6. SVM models have generalization in practice, the risk of overfitting is less in SVM.
7. SVM is always compared with ANN. When compared to ANN models, SVMs give better results.

4. SVM Disadvantages:
    1. Choosing a good kernel function is not easy.
    2. Long training time for large datasets.
    3. Difficult to understand and interpret the final model, variable weights and individual impact.
    4. Since the final model is not so easy to see, we cannot do small calibrations to the model hence it's tough to incorporate our business logic.
    5. The SVM hyperparameters are Cost -C and gamma. It is not that easy to fine-tune these hyper-parameters. It is hard to visualize their impact.

5. SVM Applications:

    1. Protein Structure Prediction
    2. Intrusion Detection
    3. Handwriting Recognition
    4. Detecting Steganography in digital images
    5. Breast Cancer Diagnosis
    6. Almost all the applications where ANN is used

7.2.3. Decision Tree Classifier Algorithm

a) 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.

b) 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.

c) The decisions or the test are performed on the basis of features of the given dataset.

d) It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

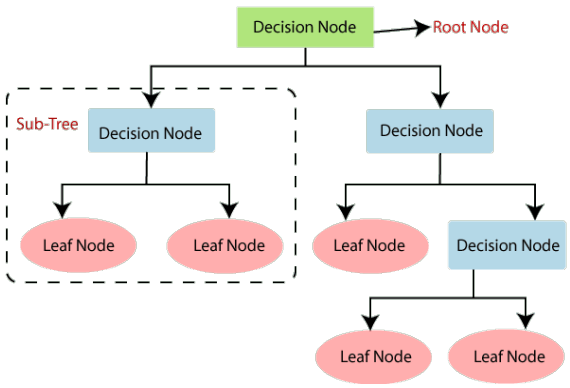
e) It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

f) In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.

g) A decision tree simply asks a question, and based on the answer (Yes/No), it further splits the tree into subtrees.

h) Below diagram explains the general structure of a decision tree:

Fig. 7.2.3.1. Decision Tree General Structure



i) Decision Tree Terminologies:

1. Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
2. Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
3. Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
4. Branch/Subtree: A tree formed by splitting the tree.
5. Pruning: Pruning is the process of removing the unwanted branches from the tree.
6. Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.

j) How does the Decision Tree algorithm Work?

1. In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of the root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.
2. For the next node, the algorithm again compares the attribute value with the other sub-nodes and moves further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

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

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

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

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

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

k) Advantages of Decision Tree:

1. It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
2. It can be very useful for solving decision-related problems.
3. It helps to think about all the possible outcomes for a problem.
4. There is less requirement of data cleaning compared to other algorithms.

l) Disadvantages of Decision Tree:

1. The decision tree contains lots of layers, which makes it complex.

2. It may have an overfitting issue, which can be resolved using the Random Forest algorithm.

3. For more class labels, the computational complexity of the decision tree may increase.

7.2.4. Random Forest Classification

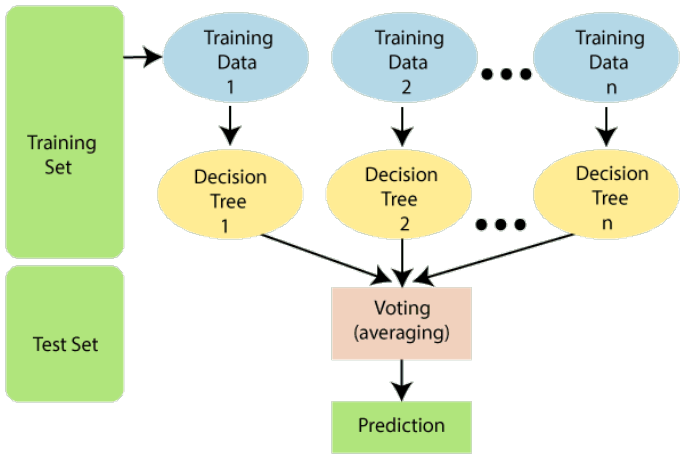
a) Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

b) As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

c) The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

d) The below diagram explains the working of the Random Forest algorithm:

Fig. 7.2.4.1. Random Forest Algorithm Working



e) How does Random Forest Algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision trees, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

f) Applications of Random Forest:

1. Banking: Banking sector mostly uses this algorithm for the identification of loan risk.

2. Medicine: With the help of this algorithm, disease trends and risks of the disease can be identified.

3. Land Use: We can identify the areas of similar land use by this algorithm.

4. Marketing: Marketing trends can be identified using this algorithm.

g) Advantages of Random Forest:

1. Random Forest is capable of performing both Classification and Regression tasks.

2. It is capable of handling large datasets with high dimensionality.

3. It enhances the accuracy of the model and prevents the overfitting issue.

h) Disadvantages of Random Forest:

1. Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

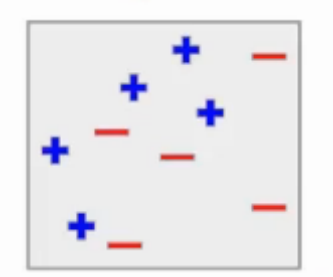
7.2.5. AdaBoost Classification

a) AdaBoost algorithm, short for Adaptive Boosting, is a Boosting technique that is used as an Ensemble Method in Machine Learning. It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights to incorrectly classified instances. Boosting is used to reduce bias as well as the variance for supervised learning. It works on the principle where learners are grown sequentially. Except for the first, each subsequent learner is grown from previously grown learners. In simple words, weak learners are converted into strong ones. Adaboost algorithm also works on the same principle as boosting, but there is a slight difference in working. Let's discuss the difference in detail.

b) Working: Let's understand how this is done using an example.

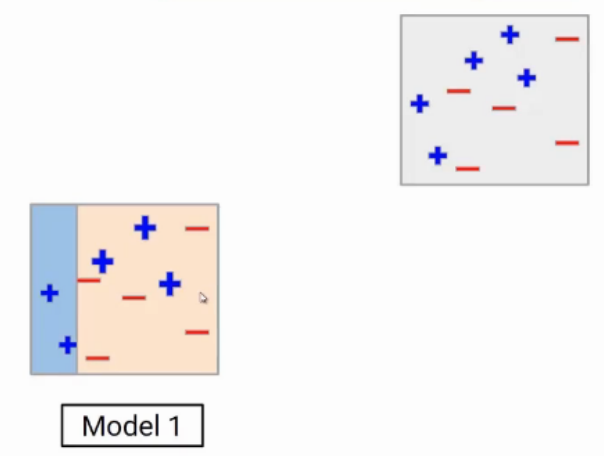
The first step is to build a model to classify data.

Fig. 7.2.5.1. AdaBoost - I



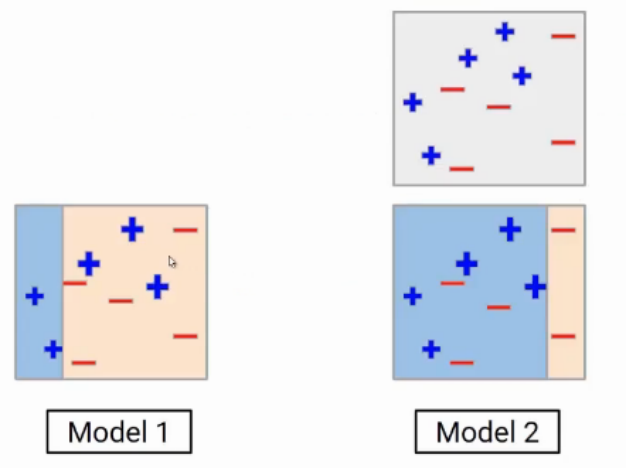
1. Model 1: First Model gives the following result, where it is able to classify two blue points on the left side and all red points correctly. But the model also miss-classify the three blue points here.

Fig. 7.2.5.2. AdaBoost - II



2. Model 2: Now, these miss- classified data points will be given higher weight. So, these three blue positive points will be given higher weights in the next iteration. For representation, the points with higher weight are bigger than the others in the image. Giving higher weights to these points means my model is going to focus more on these values. Now we will build a new model.

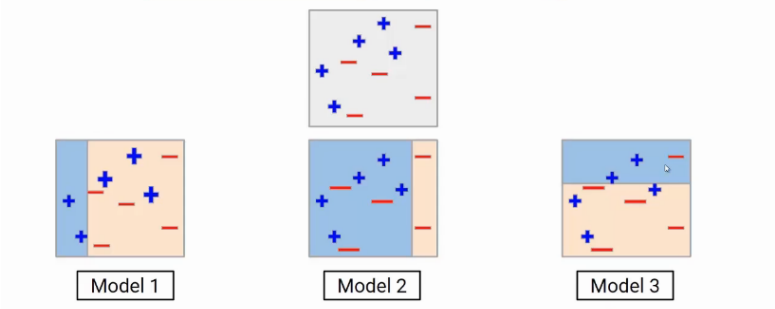
Fig. 7.2.5.3. AdaBoost - III



In the second model you will see, the model boundary has been shifted to the right side in order to correctly classify the higher weighted points. Still, it's not a perfect model. You will notice three red negatives are miss-classified by model 2.

3. Model 3: Now, these miss-classified red points will get a higher weight. Again, we will build another model and do the predictions. The task of the third model is two focuses on these three red negative points. So, the decision boundary will be something as shown here.

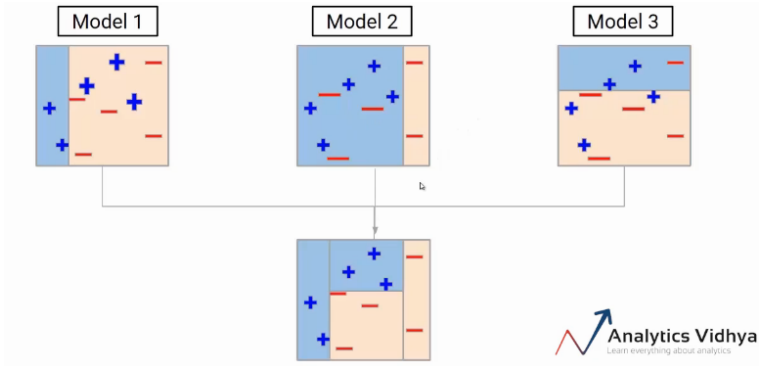
Fig. 7.2.5.4. AdaBoost - IV



This new model again incorrectly predicted some data points. At this point, we can say all these individual models are not strong enough to classify the points correctly and are often called weak learners.

4. Ensemble: Now we have to aggregate these models. One of the ways could be taking the weighted average of the individual weak learners. So our final model will be the weighted mean of individual models.

Fig. 7.2.5.5. AdaBoost - V



After multiple iterations, we will be able to create the right decision boundary with the help of all the previous weak learners. As you can see the final model is able to classify all the points correctly. This final model is known as a strong learner.

c) Advantages of AdaBoost Classification:

1. It is easier to use with less need for tweaking parameters unlike algorithms like SVM.
2. One can also use AdaBoost with SVM.
3. AdaBoost is not prone to overfitting because of the reason that parameters are not jointly optimized — stage-wise estimation slows down the learning process.
4. AdaBoost can be used to improve the accuracy of your weak classifiers hence making it flexible. It has now been extended beyond binary classification and has found use cases in

text and image classification as well.

d) Disadvantages of AdaBoost Classification:

1. Boosting technique learns progressively, it is important to ensure that you have quality data.

2. AdaBoost is also extremely sensitive to Noisy data and it is highly recommended to eliminate them.

e) AdaBoost has also been proven to be slower than XGBoost.

7.2.6. Logistic Regression

a) Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

b) Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either yes or no, 0 or 1, True or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

c) Logistic Regression is much similar to Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

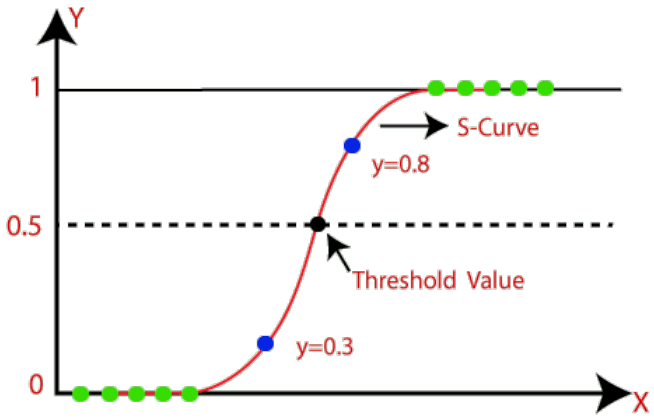
d) In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

e) The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

f) Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

g) Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

Fig. 7.2.6.1. Logistic Regression



h) Advantages of Logistic Regression:

1. Logistic regression is easier to implement, interpret, and very efficient to train.
2. It makes no assumptions about distributions of classes in feature space.
3. It can easily extend to multiple classes (multinomial regression) and a natural probabilistic view of class predictions.

4. It not only provides a measure of how appropriate a predictor (coefficient size) is, but also its direction of association (positive or negative).
5. It is very fast at classifying unknown records.
6. Good accuracy for many simple data sets and it performs well when the dataset is linearly separable.
7. It can interpret model coefficients as indicators of feature importance.
8. Logistic regression is less inclined to over-fitting but it can overfit in high dimensional datasets. One may consider Regularization (L1 and L2) techniques to avoid over-fitting in these scenarios.

i) Disadvantages of Logistic Regression:

1. If the number of observations is lesser than the number of features, Logistic Regression should not be used, otherwise, it may lead to overfitting.
2. It constructs linear boundaries.
3. The major limitation of Logistic Regression is the assumption of linearity between the dependent variable and the independent variables.
4. It can only be used to predict discrete functions. Hence, the dependent variable of Logistic Regression is bound to the discrete number set.
5. Non-linear problems can't be solved with logistic regression because it has a linear decision surface. Linearly separable data is rarely found in real-world scenarios.
6. Logistic Regression requires average or no multicollinearity between independent variables.
7. It is tough to obtain complex relationships using logistic regression. More powerful and compact algorithms such as Neural Networks can easily outperform this algorithm.

7.3. GUI Snippets

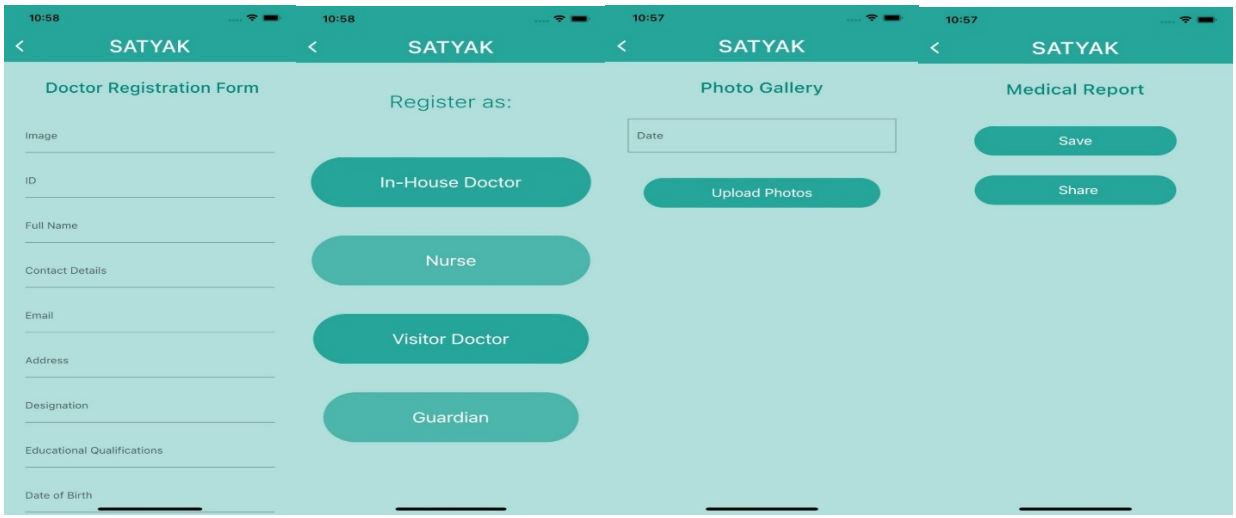


Fig. 7.3.1. GUI - I

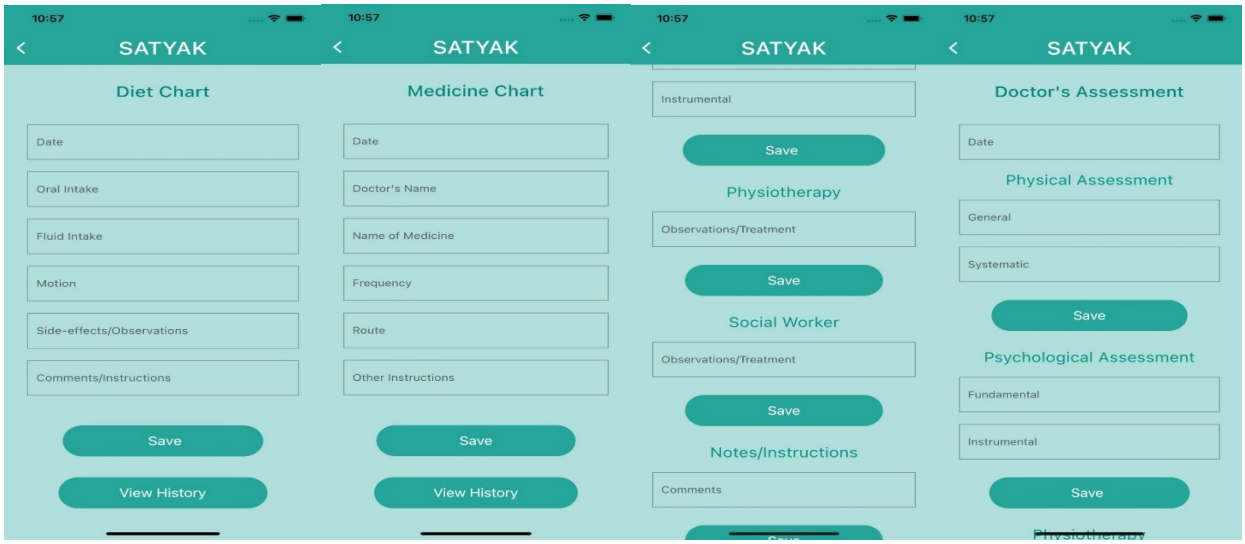


Fig. 7.3.2. GUI - II

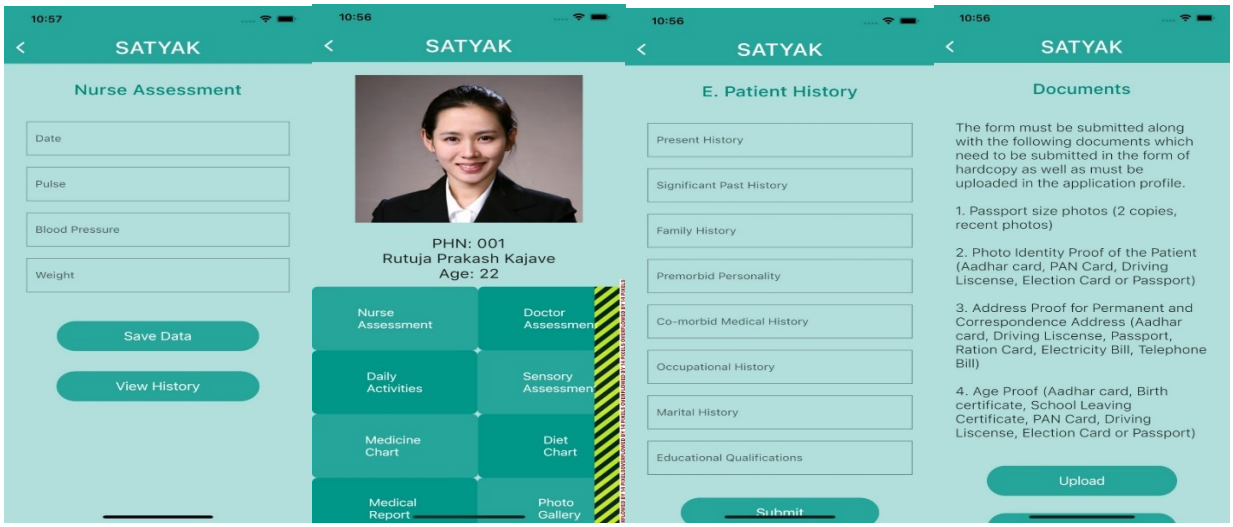


Fig. 6.2.1GUI - III

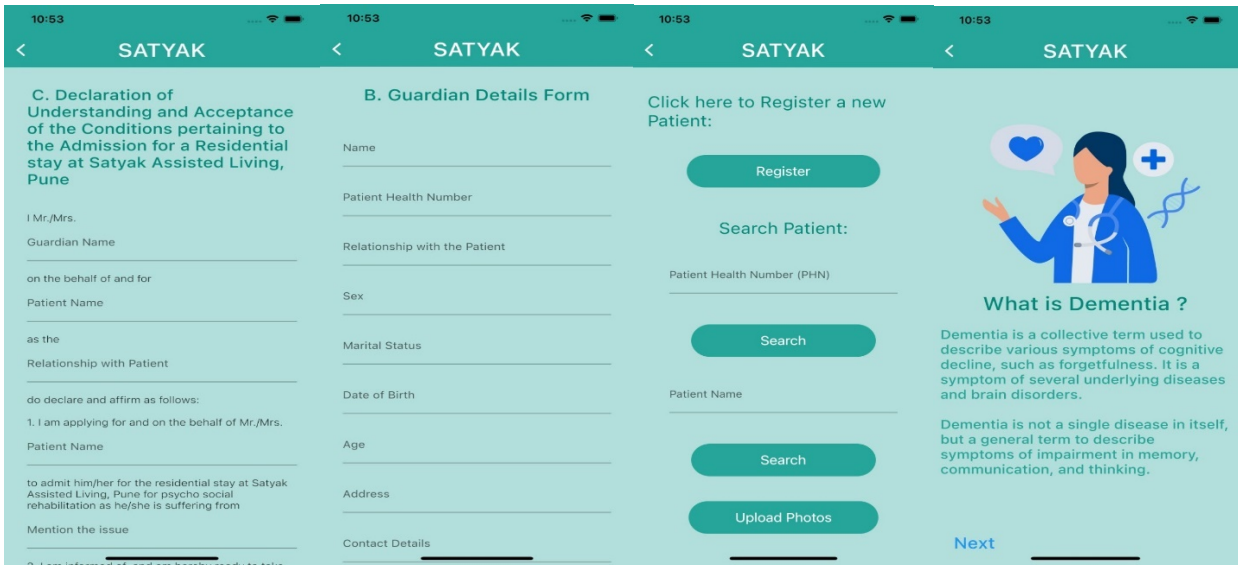


Fig. 6.2.1GUI - IV

7.4 Sample Codes

Part A:

main.dart

```
import 'package:flutter/material.dart';
import 'package:satyak_app/pages/WelcomePage.dart';

void main() {
 runApp(MyApp());
}

class MyApp extends StatelessWidget {
 // This widget is the root of your application.
 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     title: 'Flutter Demo',
     debugShowCheckedModeBanner: false,
     theme: ThemeData(
       primarySwatch: Colors.blue,
     ),
     home: WelcomePage(),
        );
 }
}
```

ContactUs.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:lottie/lottie.dart';
import 'package:satyak_app/pages/loginPage.dart';

class ContactUs extends StatefulWidget {
 @override
 _ContactUsState createState() => _ContactUsState();
}

class _ContactUsState extends State<ContactUs> {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
     resizeToAvoidBottomInset: false,
     backgroundColor: Colors.teal[100],
     appBar: AppBar(
       backgroundColor: Colors.teal[400],
       elevation: 0.0,
       title: Text(
         "SATYAK",
         style: TextStyle(fontSize: 30.0),
       ),
       centerTitle: true,
     ),
     body: SingleChildScrollView(
       child: Column(
```

```
          children: [
            SizedBox(height: 10.0),
            Container(
                child: Padding(
              padding: const EdgeInsets.fromLTRB(20.0, 60.0, 0.0, 20.0),
              child: Text(
                "Satyak: Assisted Living for Dementia Patients",
                style: TextStyle(
                    fontSize: 30,
                    color: Colors.teal[600],
                    fontWeight: FontWeight.w500),
              ),
            )),
            Container(
              height: MediaQuery.of(context).size.height / 4,
              width: MediaQuery.of(context).size.width,
              child: Lottie.asset("images/Location.json"),
            ),
            Container(
              child: Padding(
                padding: const EdgeInsets.fromLTRB(20.0, 20.0, 0.0, 20.0),
                child: Text(
                  "Address: 404 Raviraaj Green Clouds Apartments, Lane No. 3, Right
Bhusari Colony, Kothrud, Pune - 411038",
                  style: TextStyle(
                      fontSize: 20,
                      color: Colors.teal[1000],
                      fontWeight: FontWeight.w500),
                ),
              ),
            ),
            Container(
              child: Padding(
                padding: const EdgeInsets.fromLTRB(20.0, 20.0, 0.0, 20.0),
                child: Align(
                  alignment: Alignment.topLeft,
                  child: Text(
                    "Contact: +91 7028033863",
                    style: TextStyle(
                        fontSize: 20,
                        color: Colors.teal[1000],
                        fontWeight: FontWeight.w500),
                  ),
                ),
              ),
            ),
            Container(
              child: Padding(
                padding: const EdgeInsets.fromLTRB(10.0, 0.0, 20.0, 0.0),
                child: Align(
                  alignment: Alignment.topLeft,
                  child: TextButton(
                    onPressed: () {},
                    child: Text(
                      "email: rutuja.kajave.rk@gmail.com",
                      style: TextStyle(
                          fontSize: 20,
                          color: Colors.teal[1000],
                          fontWeight: FontWeight.w500),
                    ),
                  ),
                ),
              ),
            ),
            SizedBox(height: 90.0),
```

```
            Padding(
              padding: const EdgeInsets.all(0.0),
              child: CupertinoButton(
                  color: Colors.teal[400],
                  borderRadius: BorderRadius.circular(50.0),
                  child: Text(
                    "     Go Back to Login     ",
                    style: TextStyle(fontSize: 20),
                  ),
                  onPressed: () {
                    Navigator.push(context,
                        CupertinoPageRoute(builder: (context) => LoginPage()));
                  }),
            ),
          ],
        ),
      ),
    );
  }
}
```

DoctorLogin.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:satyak_app/pages/PatientDashboard.dart';
import 'package:satyak_app/pages/PatientReg1.dart';
import 'package:satyak_app/pages/UploadPhotos.dart';

class DoctorLogin extends StatefulWidget {
 @override
 _DoctorLoginState createState() => _DoctorLoginState();
}

class _DoctorLoginState extends State<DoctorLogin> {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
       resizeToAvoidBottomInset: false,
       backgroundColor: Colors.teal[100],
       appBar: AppBar(
         backgroundColor: Colors.teal[400],
         elevation: 0.0,
         title: Text(
           "SATYAK",
           style: TextStyle(fontSize: 30.0),
         ),
         centerTitle: true,
       ),
       body: SingleChildScrollView(
           child: Column(children: [
         Center(
           child: Container(
               child: Padding(
             padding: const EdgeInsets.fromLTRB(20.0, 40.0, 0.0, 20.0),
             child: Text(
               "Click here to Register a new Patient:",
               style: TextStyle(
                   fontSize: 25,
                   color: Colors.teal[600],
                   fontWeight: FontWeight.w400),
```

```
    ),
  )),
),
SizedBox(height: 20.0),
Padding(
  padding: const EdgeInsets.all(0.0),
  child: CupertinoButton(
      color: Colors.teal[400],
      borderRadius: BorderRadius.circular(50.0),
      child: Text(
        "      Register      ",
        style: TextStyle(fontSize: 20),
      ),
      onPressed: () {
        Navigator.push(context,
            CupertinoPageRoute(builder: (context) => PatientReg1()));
      }),
),
Center(
  child: Container(
      child: Padding(
    padding: const EdgeInsets.fromLTRB(20.0, 50.0, 20.0, 20.0),
    child: Column(
      children: [
        Text(
          "Search Patient:",
          style: TextStyle(
              fontSize: 25,
              color: Colors.teal[600],
              fontWeight: FontWeight.w400),
        ),
        Padding(
          padding: const EdgeInsets.all(30.0),
          child: TextFormField(
            decoration: InputDecoration(
              labelText: "Patient Health Number (PHN)",
              hintText: "001",
            ),
          ),
        ),
        SizedBox(
          height: 20.0,
        ),
        Padding(
          padding: const EdgeInsets.all(0.0),
          child: CupertinoButton(
              color: Colors.teal[400],
              borderRadius: BorderRadius.circular(50.0),
              child: Text(
                "      Search      ",
                style: TextStyle(fontSize: 20),
              ),
              onPressed: () {
                Navigator.push(
                    context,
                    CupertinoPageRoute(
                        builder: (context) => PatientDashboard()));
              }),
        ),
        Padding(
          padding: const EdgeInsets.all(30.0),
          child: TextFormField(
            decoration: InputDecoration(
              labelText: "Patient Name",
            ),
```

```
                    ),
                  ),
                  SizedBox(
                    height: 20.0,
                  ),
                  Padding(
                    padding: const EdgeInsets.all(0.0),
                    child: CupertinoButton(
                        color: Colors.teal[400],
                        borderRadius: BorderRadius.circular(50.0),
                        child: Text(
                          "      Search      ",
                          style: TextStyle(fontSize: 20),
                        ),
                        onPressed: () {
                          Navigator.push(
                              context,
                              CupertinoPageRoute(
                                  builder: (context) => PatientDashboard()));
                        }),
                  ),
                  SizedBox(
                    height: 40,
                  ),
                  Padding(
                    padding: const EdgeInsets.all(0.0),
                    child: CupertinoButton(
                        color: Colors.teal[400],
                        borderRadius: BorderRadius.circular(50.0),
                        child: Text(
                          "Upload Photos",
                          style: TextStyle(fontSize: 20),
                        ),
                        onPressed: () {
                          Navigator.push(
                              context,
                              CupertinoPageRoute(
                                  builder: (context) => UploadPhotos()));
                        }),
                  )
                ],
              ),
            )),
          ),
        ])));
  }
}
```

GuardianLogin.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:satyak_app/pages/ContactUs.dart';
import 'package:satyak_app/pages/MedicalReport.dart';
import 'package:satyak_app/pages/PatientPhotos.dart';

class GuardianLogin extends StatefulWidget {
 @override
 _NurseOtherLoginState createState() => _NurseOtherLoginState();
}

class _NurseOtherLoginState extends State<GuardianLogin> {
 @override
```

```
Widget build(BuildContext context) {
  return Scaffold(
      resizeToAvoidBottomInset: false,
      backgroundColor: Colors.teal[100],
      appBar: AppBar(
        backgroundColor: Colors.teal[400],
        elevation: 0.0,
        title: Text(
          "SATYAK",
          style: TextStyle(fontSize: 30.0),
        ),
        centerTitle: true,
      ),
      body: SingleChildScrollView(
          child: Column(children: [
        Padding(
          padding: const EdgeInsets.all(15.0),
          child: Align(
            alignment: Alignment.topCenter,
            child: Container(
              child: Image.asset(
                'images/Rutuja.jpg',
                height: 300,
                width: 300,
              ),
            ),
          ),
        ),
        Text(
          "PHN: 001",
          style: TextStyle(fontSize: 25),
        ),
        Text(
          "Rutuja Prakash Kajave",
          style: TextStyle(fontSize: 25),
        ),
        Text(
          "Age: 22",
          style: TextStyle(fontSize: 25),
        ),
        SizedBox(
          height: 40,
        ),
        Padding(
            padding: const EdgeInsets.all(10.0),
            child: CupertinoButton(
                color: Colors.teal[400],
                borderRadius: BorderRadius.circular(50.0),
                child: Text(
                  "     Medical Report     ",
                  style: TextStyle(fontSize: 25),
                ),
                onPressed: () {
                  Navigator.push(
                      context,
                      CupertinoPageRoute(
                          builder: (context) => MedicalReport()));
                })),
        SizedBox(
          height: 20,
        ),
        Padding(
            padding: const EdgeInsets.all(10.0),
            child: CupertinoButton(
                color: Colors.teal[400],
```

```
                borderRadius: BorderRadius.circular(50.0),
                child: Text(
                  "      Photo Gallery      ",
                  style: TextStyle(fontSize: 25),
                ),
                onPressed: () {
                  Navigator.push(
                      context,
                      CupertinoPageRoute(
                          builder: (context) => PatientPhotos()));
                })),
          SizedBox(
            height: 20,
          ),
          Padding(
              padding: const EdgeInsets.all(10.0),
              child: CupertinoButton(
                  color: Colors.teal[400],
                  borderRadius: BorderRadius.circular(50.0),
                  child: Text(
                    "        Contact Us        ",
                    style: TextStyle(fontSize: 25),
                  ),
                  onPressed: () {
                    Navigator.push(context,
                        CupertinoPageRoute(builder: (context) => ContactUs()));
                  })),
        ])));
  }
}
```

PatientDashboard.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:satyak_app/pages/DailyActivities.dart';
import 'package:satyak_app/pages/DietChart.dart';
import 'package:satyak_app/pages/DoctorAssessment.dart';
import 'package:satyak_app/pages/MedicalReport.dart';
import 'package:satyak_app/pages/MedicineChart.dart';
import 'package:satyak_app/pages/NurseAssessment.dart';
import 'package:satyak_app/pages/PhotoGallery.dart';
import 'package:satyak_app/pages/SensoryAssessment.dart';

class PatientDashboard extends StatefulWidget {
 @override
 _PatientDashboardState createState() => _PatientDashboardState();
}

class _PatientDashboardState extends State<PatientDashboard> {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
       resizeToAvoidBottomInset: false,
       backgroundColor: Colors.teal[100],
       appBar: AppBar(
         backgroundColor: Colors.teal[400],
         elevation: 0.0,
         title: Text(
           "SATYAK",
           style: TextStyle(fontSize: 30.0),
```

```
    ),
    centerTitle: true,
  ),
  body: SingleChildScrollView(
    child: Column(
      children: [
        Image.asset(
          'images/Rutuja.jpg',
          height: 300,
          width: 300,
        ),
        Text(
          "PHN: 001",
          style: TextStyle(fontSize: 22),
        ),
        Text(
          "Rutuja Prakash Kajave",
          style: TextStyle(fontSize: 22),
        ),
        Text(
          "Age: 22",
          style: TextStyle(fontSize: 22),
        ),
        SizedBox(height: 10),
        Row(
          children: [
            CupertinoButton(
                color: Colors.teal[400],
                padding:
                    const EdgeInsets.fromLTRB(60.0, 34.0, 60.0, 34.0),
                child: Text(
                  "Nursing \nAssessment",
                  style: TextStyle(fontSize: 18),
                ),
                onPressed: () {
                  Navigator.push(
                      context,
                      CupertinoPageRoute(
                          builder: (context) => NurseAssessment()));
                }),
            CupertinoButton(
                color: Colors.teal[500],
                padding:
                    const EdgeInsets.fromLTRB(60.0, 34.0, 50.0, 34.0),
                child: Text(
                  "Doctor \nAssessment",
                  style: TextStyle(fontSize: 18),
                ),
                onPressed: () {
                  Navigator.push(
                      context,
                      CupertinoPageRoute(
                          builder: (context) => DoctorAssessment()));
                }),
          ],
        ),
        Row(
          children: [
            CupertinoButton(
                color: Colors.teal[500],
                padding:
                    const EdgeInsets.fromLTRB(72.0, 34.0, 47.0, 34.0),
                child: Text(
                  "Activities of \nDaily Life",
                  style: TextStyle(fontSize: 18),
```

```
          ),
          onPressed: () {
            Navigator.push(
                context,
                CupertinoPageRoute(
                    builder: (context) => DailyActivities()));
          }),
      CupertinoButton(
          color: Colors.teal[400],
          padding:
              const EdgeInsets.fromLTRB(60.0, 34.0, 50.0, 34.0),
          child: Text(
            "Sensory \nAssessment",
            style: TextStyle(fontSize: 18),
          ),
          onPressed: () {
            Navigator.push(
                context,
                CupertinoPageRoute(
                    builder: (context) => SensoryAssessment()));
          }),
    ],
  ),
  Row(
    children: [
      CupertinoButton(
          color: Colors.teal[400],
          padding:
              const EdgeInsets.fromLTRB(70.0, 34.0, 70.0, 34.0),
          child: Text(
            "Medicine \nChart",
            style: TextStyle(fontSize: 18),
          ),
          onPressed: () {
            Navigator.push(
                context,
                CupertinoPageRoute(
                    builder: (context) => MedicineChart()));
          }),
      CupertinoButton(
          color: Colors.teal[500],
          padding:
              const EdgeInsets.fromLTRB(86.0, 34.0, 78.0, 34.0),
          child: Text(
            "Diet \nChart",
            style: TextStyle(fontSize: 18),
          ),
          onPressed: () {
            Navigator.push(
                context,
                CupertinoPageRoute(
                    builder: (context) => DietChart()));
          }),
    ],
  ),
  Row(
    children: [
      CupertinoButton(
          color: Colors.teal[500],
          padding:
              const EdgeInsets.fromLTRB(75.0, 32.0, 75.0, 32.0),
          child: Text(
            "Medical \nReport",
            style: TextStyle(fontSize: 18),
          ),
```

```
                    onPressed: () {
                      Navigator.push(
                          context,
                          CupertinoPageRoute(
                              builder: (context) => MedicalReport()));
                    }),
                CupertinoButton(
                    color: Colors.teal[400],
                    padding:
                        const EdgeInsets.fromLTRB(80.0, 32.0, 73.0, 32.0),
                    child: Text(
                      "Photo \nGallery",
                      style: TextStyle(fontSize: 18),
                    ),
                    onPressed: () {
                      Navigator.push(
                          context,
                          CupertinoPageRoute(
                              builder: (context) => PhotoGallery()));
                    }),
              ],
            )
          ],
        ),
      ));
 }
}
```

Part B:

```
# -*- coding: utf-8 -*-
"""DementiaDataAnalysis.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1BQTQ-aaRg-5a2N4UFm3RB1_GLyVgE9WR

## Dataset and Analysis

Import all the necessary libraries and modules for machine learning and deep learning
"""

# Commented out IPython magic to ensure Python compatibility.
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.model_selection import cross_val_score

import tensorflow as tf
from keras.models import Model, Sequential
from keras.layers import Input,Dense

from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, recall_score, roc_curve,
auc

# %matplotlib inline

sns.set()

"""Read the .csv file for our data"""

df = pd.read_csv('oasis_longitudinal.csv')
sns.countplot(x='Group', data=df)
df.head()

"""Consise summary of Data (Mean values, standard deviations and other values)"""

df.describe()

"""Filter all the patients according to their visits and show patients whose visit input
== 1"""

df = df.loc[df['Visit']==1]
sns.countplot(x='Group', data=df)
df.head()

"""Reset the index values so that they are in sequence"""

df = df.reset_index(drop=True)
df

"""Drop unnecessary columns and replace the data which cannot be read"""

df['M/F'] = df['M/F'].replace(['F','M'], [0,1])
sns.countplot(x='Group', data=df) # M/F column

df['Group'] = df['Group'].replace(['Converted'], ['Demented']) # Target variable
df['Group'] = df['Group'].replace(['Demented', 'Nondemented'], [1,0])
sns.countplot(x='Group', data=df) # Target variable
df = df.drop(['MRI ID', 'Visit', 'Hand'], axis=1) # Drop unnecessary columns
df.head()

"""Drawing Function for the Bar Graph"""

def bar_chart(feature):
    Demented = df[df['Group']==1][feature].value_counts()
    Nondemented = df[df['Group']==0][feature].value_counts()
    df_bar = pd.DataFrame([Demented,Nondemented])
    df_bar.index = ['Demented','Nondemented']
    df_bar.plot(kind='bar',stacked=True, figsize=(8,5))

"""Executing the function and plotting the graph"""

# Gender  and  Group ( Femal=0, Male=1)
bar_chart('M/F')
plt.xlabel('Group')
plt.ylabel('Number of patients')
plt.legend()
plt.title('Gender and Demented rate')

"""MMSE Scores"""

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)

"""Brain Volume ratio analysis"""

#bar_chart('ASF') = Atlas Scaling Factor
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)

#eTIV = Estimated Total Intracranial Volume
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)

#'nWBV' = Normalized Whole Brain Volume
# Nondemented = 0, Demented =1
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)

"""The chart indicates that Nondemented group has higher brain volume ratio than
Demented group. This is assumed to be because the diseases affect the brain to be
shrinking its tissue.

Age Analysis
"""

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)

"""There is a higher concentration of 70-80 years old in the Demented patient group than
those in the nondemented patients. We guess patients who suffered from that kind of
disease has lower survival rate so that there are a few of 90 years old.

Education Analysis
"""

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.figure(figsize=(10,5))
sns.violinplot(x='M/F', y='CDR', data=df)
plt.title('Violin plots of CDR by Gender',fontsize=14)
plt.xlabel('Gender',fontsize=13)
plt.ylabel('CDR',fontsize=13)
plt.show()

"""The above graph implies that men are more likely to suffer from dementia as to
women"""

plt.figure(figsize=(10,5))
sns.violinplot(x='CDR', y='Age', data=df)
plt.title('Violin plot of Age by CDR',fontsize=14)
```

```
plt.xlabel('CDR',fontsize=13)
plt.ylabel('Age',fontsize=13)
plt.show()

def outliers_iqr(ys):
    quartile_1, quartile_3 = np.percentile(ys, [25, 75])
    iqr = quartile_3 - quartile_1
    lower_bound = quartile_1 - (iqr * 1.5)
    upper_bound = quartile_3 + (iqr * 1.5)
    return np.where((ys > upper_bound) | (ys < lower_bound))

list_atributes = ['MR Delay','EDUC', "SES", "MMSE", 'eTIV', "nWBV", "ASF"]
print("Outliers: \n")
for item in list_atributes:
    print(item,': ',outliers_iqr(df[item]))

from pylab import rcParams
rcParams['figure.figsize'] = 8,5
cols = ['Age','MR Delay', 'EDUC', 'SES', 'MMSE', 'CDR','eTIV','nWBV','ASF']
x=df.fillna('')
sns_plot = sns.pairplot(x[cols])

#boxplots which shows the IQR(Interquartile Range )
fig, axes = plt.subplots(2,3,figsize = (16,6))
fig.suptitle("Box Plot",fontsize=14)
sns.set_style("whitegrid")
sns.boxplot(data=df['SES'], orient="v",width=0.4, palette="colorblind",ax = axes[0][0]);
sns.boxplot(data=df['EDUC'], orient="v",width=0.4, palette="colorblind",ax = axes[0]
[1]);
sns.boxplot(data=df['MMSE'], orient="v",width=0.4, palette="colorblind",ax = axes[0]
[2]);
sns.boxplot(data=df['CDR'], orient="v",width=0.4, palette="colorblind",ax = axes[1][0]);
sns.boxplot(data=df['eTIV'], orient="v",width=0.4, palette="colorblind",ax = axes[1]
[1]);
sns.boxplot(data=df['ASF'], orient="v",width=0.4, palette="colorblind",ax = axes[1][2]);
#xlabel("Time");

#convet the charecter data into numeric
group_map = {"Demented": 1, "Nondemented": 0}

df['Group'] = df['Group'].map(group_map)

def plot_correlation_map( df ):
    corr = df.corr()
    _ , ax = plt.subplots( figsize =( 12 , 10 ) )
    cmap = sns.diverging_palette( 240 , 10 , as_cmap = True )
    _ = sns.heatmap(corr,cmap = cmap,square=True, cbar_kws={ 'shrink' : .9 }, ax=ax,
annot = True, annot_kws = { 'fontsize' : 12 })

plot_correlation_map(df)

""" Intermediate Result Summary

1. Men are more likely with demented, an Alzheimer's Disease, than Women.

2. Demented patients were less educated in terms of years of education.

3. Nondemented group has higher brain volume than Demented group.

4. Higher concentration of 70-80 years old in Demented group than those in the
nondemented patients.


"""
```

```
        # -*- coding: utf-8 -*-
"""DementiaAlgorithms.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1vxR6eQ9ycxZEOyIPqg4uAYQmAElV677a

## Dataset and Analysis

Import all the necessary libraries and modules for machine learning and deep learning
"""

# Commented out IPython magic to ensure Python compatibility.
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.model_selection import cross_val_score

import tensorflow as tf
from keras.models import Model, Sequential
from keras.layers import Input,Dense

from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, recall_score, roc_curve,
auc

# %matplotlib inline

sns.set()

"""Read the .csv file for our data"""

df = pd.read_csv('oasis_longitudinal.csv')

"""Filter all the patients according to their visits and show patients whose visit input
== 1"""

df = df.loc[df['Visit']==1]

"""Reset the index values so that they are in sequence"""

df = df.reset_index(drop=True)

"""Drop unnecessary columns and replace the data which cannot be read"""

df['M/F'] = df['M/F'].replace(['F','M'], [0,1]) # M/F column
df['Group'] = df['Group'].replace(['Converted'], ['Demented']) # Target variable
df['Group'] = df['Group'].replace(['Demented', 'Nondemented'], [1,0]) # Target variable
df = df.drop(['MRI ID', 'Visit', 'Hand'], axis=1) # Drop unnecessary columns

"""## Data Preprocessing

Identify fields which have unreadable datatypes
"""
```

```
pd.isnull(df).sum()
#df["SES"].fillna(df.groupby("EDUC")["SES"].transform("median"), inplace=True)

"""Drop fields which have unreadable datatypes"""

df_dropna = df.dropna(axis=0, how='any')
pd.isnull(df_dropna).sum()
#pd.isnull(df).sum()

"""Differentiate patients as to demented and nondemented"""

df_dropna['Group'].value_counts()

"""Reset indexes in sequential order"""

df = df_dropna.reset_index(drop=True)
df

"""Set x and y inputs for Training and Validation Data"""

Y = df['Group'].values # Target for the model
X = df[['M/F', 'Age', 'EDUC', 'SES', 'MMSE', 'eTIV', 'nWBV', 'ASF']] # Features we use

# splitting into three sets
X_trainval, X_test, Y_trainval, Y_test = train_test_split(
    X, Y, random_state=10,stratify=df['Group'].values)

# Feature scaling
scaler = StandardScaler().fit(X_trainval)
X_trainval_scaled = scaler.transform(X_trainval)
X_test_scaled = scaler.transform(X_test)

"""Print the Data"""

print('Number of demented samples in training data:',(np.asarray(Y_trainval)==1).sum())
print('Number of non-demented samples in training data:',
(np.asarray(Y_trainval)==0).sum())
print('Traning data features:',X_trainval.shape)
print('Training data labels:',Y_trainval.shape)

print('Testing data features:',X_test_scaled.shape)
print('Testing data labels:',Y_test.shape)

"""The above section of the code included fetching the file and filtering and modifying
the data according to our use.

## Simple Feed-Forward Neural Network

Initialising the model
"""

# model = Sequential()
# model.add(Dense(12,activation='relu',input_dim=X_train.shape[0]))
# model.add(Dense(8,activation='relu'))
# model.add(Dense(1,activation='sigmoid'))
input_shape = (X_trainval_scaled.shape[-1])
i = Input(shape = input_shape)
x = Dense(12,activation='relu',kernel_initializer='he_normal')(i)
x = Dense(8,activation='relu',kernel_initializer='he_normal')(x)
x = Dense(1, activation='sigmoid')(x)
model = Model(i,x)
model.summary()

"""Compiling the model"""
```

```
model.compile(loss='binary_crossentropy',optimizer='adam',metrics='accuracy')
model.fit(X_trainval_scaled,Y_trainval,epochs=75,batch_size=4,validation_data=(X_test_sc
aled,Y_test))

"""Printing the Output"""

loss,acc = model.evaluate(X_test_scaled,Y_test)
PredictedOutput = (model.predict(X_test_scaled) > 0.5).astype("int32")
print(PredictedOutput)
test_recall = recall_score(Y_test, PredictedOutput, pos_label=1)
fpr, tpr, thresholds = roc_curve(Y_test, PredictedOutput, pos_label=1)
test_auc = auc(fpr, tpr)
print("Test accuracy is", acc)
confusion_matrix(Y_test, PredictedOutput)

"""## SVM

Building a SVM model
"""

best_score = 0
kfolds = 5
for c_paramter in [0.001, 0.01, 0.1, 1, 10, 100, 1000]: #iterate over the values we need
to try for the parameter C
    for gamma_paramter in [0.001, 0.01, 0.1, 1, 10, 100, 1000]: #iterate over the values
we need to try for the parameter gamma
        for k_parameter in ['rbf', 'linear', 'poly', 'sigmoid']: # iterate over the
values we need to try for the kernel parameter
            svmModel = SVC(kernel=k_parameter, C=c_paramter, gamma=gamma_paramter)
#define the model
            # perform cross-validation
            scores = cross_val_score(svmModel, X_trainval_scaled, Y_trainval, cv=kfolds,
scoring='accuracy')
            # the training set will be split internally into training and cross
validation

            # compute mean cross-validation accuracy
            score = np.mean(scores)
            # if we got a better score, store the score and parameters
            if score > best_score:
                best_score = score #store the score
                best_parameter_c = c_paramter #store the parameter c
                best_parameter_gamma = gamma_paramter #store the parameter gamma
                best_parameter_k = k_parameter


# rebuild a model with best parameters to get score
SelectedSVMmodel = SVC(C=best_parameter_c, gamma=best_parameter_gamma,
kernel=best_parameter_k).fit(X_trainval_scaled, Y_trainval)

"""Printing output"""

test_score = SelectedSVMmodel.score(X_test_scaled, Y_test)
PredictedOutput = SelectedSVMmodel.predict(X_test_scaled)
print(PredictedOutput)
test_recall = recall_score(Y_test, PredictedOutput, pos_label=1)
fpr, tpr, thresholds = roc_curve(Y_test, PredictedOutput, pos_label=1)
test_auc = auc(fpr, tpr)
print("Best accuracy on cross validation set is:", best_score)
print("Best parameter for c is: ", best_parameter_c)
print("Best parameter for gamma is: ", best_parameter_gamma)
print("Best parameter for kernel is: ", best_parameter_k)
print("Test accuracy with the best parameters is", test_score)
confusion_matrix(Y_test, PredictedOutput)
```

```
"""## Decision Tree Classification

Building a Decision Tree Classification Model
"""

best_score = 0

for md in range(1, 9): # iterate different maximum depth values
    # train the model
    treeModel = DecisionTreeClassifier(random_state=0, max_depth=md, criterion='gini')
    # perform cross-validation
    scores = cross_val_score(treeModel, X_trainval_scaled, Y_trainval, cv=kfolds,
scoring='accuracy')

    # compute mean cross-validation accuracy
    score = np.mean(scores)

    # if we got a better score, store the score and parameters
    if score > best_score:
        best_score = score
        best_parameter = md

# Rebuild a model on the combined training and validation set
SelectedDTModel =
DecisionTreeClassifier(max_depth=best_parameter).fit(X_trainval_scaled, Y_trainval )

"""Printing Output"""

test_score = SelectedDTModel.score(X_test_scaled, Y_test)
PredictedOutput = SelectedDTModel.predict(X_test_scaled)
print(PredictedOutput)
test_recall = recall_score(Y_test, PredictedOutput, pos_label=1)
fpr, tpr, thresholds = roc_curve(Y_test, PredictedOutput, pos_label=1)
test_auc = auc(fpr, tpr)
print("Best accuracy on validation set is:", best_score)
print("Best parameter for the maximum depth is: ", best_parameter)
print("Test accuracy with best parameter is ", test_score)
confusion_matrix(Y_test, PredictedOutput)

"""## Random Forest Classification

Building a random forest classification model
"""

best_score = 0

for M in range(2, 15, 2): # combines M trees
    for d in range(1, 9): # maximum number of features considered at each split
        for m in range(1, 9): # maximum depth of the tree
            # train the model
            # n_jobs(4) is the number of parallel computing
            forestModel = RandomForestClassifier(n_estimators=M, max_features=d,
n_jobs=4,
                                                 max_depth=m, random_state=0)

            # perform cross-validation
            scores = cross_val_score(forestModel, X_trainval_scaled, Y_trainval,
cv=kfolds, scoring='accuracy')

            # compute mean cross-validation accuracy
            score = np.mean(scores)

            # if we got a better score, store the score and parameters
            if score > best_score:
```

```
                best_score = score
                best_M = M
                best_d = d
                best_m = m

# Rebuild a model on the combined training and validation set
SelectedRFModel = RandomForestClassifier(n_estimators=M, max_features=d,
                                         max_depth=m,
random_state=0).fit(X_trainval_scaled, Y_trainval )

"""Printing Output"""

PredictedOutput = SelectedRFModel.predict(X_test_scaled)
print(PredictedOutput)
test_score = SelectedRFModel.score(X_test_scaled, Y_test)
test_recall = recall_score(Y_test, PredictedOutput, pos_label=1)
fpr, tpr, thresholds = roc_curve(Y_test, PredictedOutput, pos_label=1)
test_auc = auc(fpr, tpr)
print("Best accuracy on validation set is:", best_score)
print("Best parameters of M, d, m are: ", best_M, best_d, best_m)
print("Test accuracy with the best parameters is", test_score)
confusion_matrix(Y_test, PredictedOutput)

"""## AdaBoost Classification

Building AdaBoost classification Model
"""

best_score = 0

for M in range(2, 15, 2): # combines M trees
    for lr in [0.0001, 0.001, 0.01, 0.1, 1]:
        # train the model
        boostModel = AdaBoostClassifier(n_estimators=M, learning_rate=lr,
random_state=0)

        # perform cross-validation
        scores = cross_val_score(boostModel, X_trainval_scaled, Y_trainval, cv=kfolds,
scoring='accuracy')

        # compute mean cross-validation accuracy
        score = np.mean(scores)

        # if we got a better score, store the score and parameters
        if score > best_score:
            best_score = score
            best_M = M
            best_lr = lr

# Rebuild a model on the combined training and validation set
SelectedBoostModel = AdaBoostClassifier(n_estimators=M, learning_rate=lr,
random_state=0).fit(X_trainval_scaled, Y_trainval )

"""Printing Output"""

PredictedOutput = SelectedBoostModel.predict(X_test_scaled)
print(PredictedOutput)
test_score = SelectedRFModel.score(X_test_scaled, Y_test)
test_recall = recall_score(Y_test, PredictedOutput, pos_label=1)
fpr, tpr, thresholds = roc_curve(Y_test, PredictedOutput, pos_label=1)
test_auc = auc(fpr, tpr)
print("Best accuracy on validation set is:", best_score)
print("Best parameter of M is: ", best_M)
print("best parameter of LR is: ", best_lr)
print("Test accuracy with the best parameter is", test_score)
```

```
confusion_matrix(Y_test, PredictedOutput)

"""## Logistic Regression

Building a Logisting Regression Model
"""

best_score=0
kfolds=5 # set the number of folds

for c in [0.001, 0.1, 1, 10, 100]:
    logRegModel = LogisticRegression(C=c)
    # perform cross-validation
    scores = cross_val_score(logRegModel, X_trainval_scaled, Y_trainval, cv=kfolds,
scoring='accuracy')

    # compute mean cross-validation accuracy
    score = np.mean(scores)

    # Find the best parameters and score
    if score > best_score:
        best_score = score
        best_parameters = c

# rebuild a model on the combined training and validation set
SelectedLogRegModel = LogisticRegression(C=best_parameters).fit(X_trainval_scaled,
Y_trainval)

"""Printing output"""

test_score = SelectedLogRegModel.score(X_test_scaled, Y_test)
PredictedOutput = SelectedLogRegModel.predict(X_test_scaled)
print(PredictedOutput)
test_recall = recall_score(Y_test, PredictedOutput, pos_label=1)
fpr, tpr, thresholds = roc_curve(Y_test, PredictedOutput, pos_label=1)
test_auc = auc(fpr, tpr)
print("Best accuracy on validation set is:", best_score)
print("Best parameter for regularization (C) is: ", best_parameters)
print("Test accuracy with best C parameter is", test_score)
confusion_matrix(Y_test, PredictedOutput)
```

CHAPTER 8: RESULT ANALYSIS

Part A:

In the Part A of our project scope we have analyzed the assessments using scores, scales and behavior of the patient. The data is AutoSummarized at a regular interval of time. Based on the analysis of scores and behaviors data is visualized to come to a conclusion and line of treatment. There are three levels of patients in dementia. So based on the values calculated and analyzed the patients are classified into these three levels.

Part B:

a) Comparative Analysis of algorithms

Source: aisoma.de Algorithm	Primary Problem	Predictors	Power	Raw Implementation	Inter-pretabilty	Regression also	Normalization
k-NN	Multiclass or binary	Numeric	Medium	Easy	Good	No	Required
perceptron	Binary	Numeric	Low	Easy	Good	No	No
Logistic Regr.	Binary	Numeric	Low	Easy	Good	No	No
Linear Discr. Analysis	Binary	Numeric	Low	Medium	Medium	No	No
Naive Bays	Multiclass or binary	Categorical	Medium	Medium	Good	No	Required
Decision Tree	Multiclass or binary	Numeric or categorical	High	Difficult	Good	Yes	No
Random Forest	Multiclass or binary	Numeric or categorical	High	Difficult	Good	Yes	No
Adaboost	Multiclass or binary	Numeric or categorical	High	Medium	Medium	Yes	Usually
SVM	Binary	Numeric or categorical	High	Very Difficult	Medium	No	Yes
Neural Networks	Multiclass or binary	Numeric or categorical	Very High	Very Difficult	Weak	Yes	Yes

AISOMA©

Table. 8.1. Comparative Analysis - I

Name	Type	Description	Advantages	Disadvantages
Linear Regression		-The best fit line through all data points	-Easy to understand -you can clearly see what the biggest drivers odf the model are.	-sometimes to simple to capture cpmöex relationships between variables, -Tendency für the model to overfit.
Logistic Regression		-The adoption for linear regression to problembs of classification	-Easy to understand	-sometimes to simple to capture cpmöex relationships between variables, -Tendency für the model to overfit.
Decision Tree		-A graph that uses branching method to match all possible outcomes of a decision	-Easy to understand and implement.	-Not often use of ist own for prediction because it's also often too simple and not powerful enough for complex data.
Random Forest		- Takes the average of many decision trees. Each tree is weaker than the full decision tree, but combining them we get better overall performance.	-A sort of „wisdom of the crowd", Tend to result in very high quality results. -Fast to train	-Can be slow to output predictions relative to other algorithms. -Not easy to understand predictions.
Gradient Boosting		-Uses even weaker decision trees that increasingly focused on „hard examples"	-High-performing	-A small change in the future set or training set can create radical changes in the model. -Not easy to understand predictions.
Neural Networks		-Mimics the behaviour of the brain. NNs are interconnected Neurons that pass messages to each other. Deep Learning uses severak layers of NNs to put one after the other.	-Can handle extremely complex tasks. No other alsgorithm comes close in image recognition.	-very very slow to train. Because they have so many layers. Require a lot of power. -Almost impossible to understand predictions.

Table. 8.2. Comparative Analysis - II

The best algorithm available for classification of medical data is Feed Forward Neural Network when it comes to Deep learning and Logistic Regression or Decision Tree when it comes to Machine Learning.

According to our implementation we found out that the best algorithm for classification of database of patients into two categories namely demented and nondemented is Logistic regression. Further, we have listed all the algorithms starting from the best accuracy to worst for our database:

1. Logistic Regression
2. Decision Tree
3. Feed forward Neural Network
4. Random Forest
5. AdaBoost
6. SVM

CHAPTER 9: CONCLUSION AND FUTURE SCOPE

Conclusion

In our project, we have learnt to apply the concepts of Data Visualization to analyze the data. Also, we have used Machine Learning and Deep Learning algorithms to classify the patients are demented and non-demented.

Dementia is a degenerative disease that eventually affects a person's ability to live independently. There are many types of dementia, although Alzheimer's disease is the most common type. Delirium and depression can be confused with dementia and a thorough evaluation should rule out other causes of cognitive loss prior to making a diagnosis of dementia.

Challenging behaviors can be caused by unmet needs and may be a means of communication. By carefully observing what comes directly before and after a behavior, the caregiver may be able to determine the underlying need and learn how to alleviate the challenging behavior.

People with dementia need to be treated with kindness and with the knowledge that they can still enjoy life. Physical and chemical restraints should be used only as a last resort. There are many proven alternatives to physical and chemical restraints that are the mainstays of individualized care.

Activities of daily living are disrupted in those with dementia. As the dementia gets worse, family members and caregivers must step in an assist with personal care and household management. Individual and group activities can provide a sense of accomplishment and well-being.

Caregiver training is an essential component for anyone caring for a person with dementia. Family caregivers play a critical and often-overlooked role in the care of loved ones with dementia—especially in the early-to-moderate stages. Caregivers often experience stress,

which does not abate simply by placing their family member in a care facility. In a facility, professional caregivers must be trained to view the person with ADRD in the context of a family.

Facilities built around a philosophy of person-centered care can have a profound and positive effect on challenging behaviors associated with dementia. Providing a safe, clean, home-like environment in which residents and staff work together has been shown to improve outcomes in those with dementia.

Caregivers—both family and professional—experience many ethical conflicts when caring for a person with dementia. Education and training in ethical decision making and conflict resolution are invaluable tools to improve the experience of those with dementia.

Future Scope

As a part of future scope is to explore other scenarios and apply them in practice to enhance the current healthcare data management:

a) Add voice interface that can be used as a medium of communication with the patient.
b) Establish an IOT based sensor system to monitor the activities of the patient.
c) Collaborate with other organizations and services as required.
d) Add Payment gateways.

APPENDIX A: PLAGARISM REPORT

APPENDIX B: DOCTOR'S LETTERHEAD

REFERANCES

1. https://sourceforge.net/software/electronic-medical-records/android/
2. http://article.sapub.org/10.5923.j.se.20170601.01.html
3. https://www.researchgate.net/profile/Patrick_Ngulube/publication/324871047_ENTERPRISE_CONTENT_MANAGEMENT_SYSTEM_IMPLEMENTATION_READINESS_TO_IMPROVE_MEDICAL_RECORDS_MANAGEMENT_IN_LIMPOPO_PROVINCE_SOU TH_AFRICA/links/5af93eb00f7e9b026bf6dd12/ENTERPRISE-CONTENTMANAGEMENT-SYSTEM-IMPLEMENTATION-READINESS-TO-IMPROVEMEDICAL-RECORDS-MANAGEMENT-IN-LIMPOPO-PROVINCE-SOUTHAFRICA.pdf
4. https://www.lovelycoding.org/hospital-management-system-project-for-final-year/
5. https://www.academia.edu/5202538/HOSPITAL_MANAGEMENT_SYSTEM_A_Project_work_submitted_to_the_DEPARTMENT_OF_COMPUTER_APPLICATIONS_Guided_by
6. 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
7. Marcus, DS, Wang, TH, Parker, J, Csernansky, JG, Morris, JC, Buckner, RL. Open Access Series of Imaging Studies (OASIS): CrossSectional 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
8. 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.

9. 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.
10. Magnin, B., Mesrob, L., Kinkingnéhun, S. et al. Support vector machinebased classification of Alzheimer's disease from whole-brain anatomical MRI. Neuroradiology (2009) 51: 73. doi.org/10.1007/s00234-008-0463x
11. http://scikit-learn.org/stable/modules/preprocessing.html#imputation
12. 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.
13. Filipovych, R., Davatzikos, C., 2011. Semi-supervised pattern classification of medical images: application to mild cognitive impairment (MCI). Neuroimage 55 (3), 1109–1119. https://doi.org/ 10.1016/j.neuroimage.2010.12.066.
14. Zhang, D., Shen, D., 2012. Predicting future clinical changes ofMCI patients using longitudinal and multimodal biomarkers. PLoS One 7 (3), e33182. https://doi.org/ 10.1371/journal.pone.0033182
15. 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
16. 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