Benchmarking Machine Learning Models for Dementia Prediction

Table of Contents

1. Project Title and Team Members

1.1 Project Title: Benchmarking Machine Learning Models for Dementia Prediction

1.2 Team Members

1. **Neha Goud Baddam (Nehagoudbaddam@my.unt.edu)**

2. **Purandhara Maharshi (purandharamaharshichidurala@my.unt.edu)**

3. **Sri Harsha Gurram (sriharshagurram@my.unt.edu)**

4. **Tejaswi Reddy Siddareddy (TejaswiReddySiddareddy@my.unt.edu)**

GitHub Project Repository Link:

https://github.com/nehabaddam/Machine-Learning-Project/

2. Abstract

Dementia is a neurological condition that worsens over time and mostly impacts the cognition, behavior, and memory of the patient. The aging of the world's population is clearly having an impact on the prevalence of Alzheimer's disease. Alzheimer's disease typically manifests as confusion, memory loss, personality changes, and a decline in day-to-day functioning. It is the main contributor to dementia and has a significant impact on people, families, healthcare systems, and societies globally.

Cognitive decline, which is usually linked to disorders like Alzheimer's disease, is of great concern among worldwide health authorities as it has no cure and has a long prodromal phase. Understanding the factors that contribute to dementia's progression is essential for an early diagnosis for effective management of symptoms. Due to the drastic progression in the field of AI, we can use ML techniques to predict Dementia in early stages. This can help in monitoring the patient from the early stage and control the deprivation of memory during the prodromal phase.

3. Problem Specification

Our objective is to use various ML models and benchmark each model for the Dementia Dataset [1]. We shall use the features in the dataset to predict if a patient has dementia. We

shall also compare the outputs from different ML models to see which one works better for the Dataset. By doing this we will learn the best model for predicting Dementia, which could help users or health organizations in the diagnosis of Dementia and understand the features that affect the most.

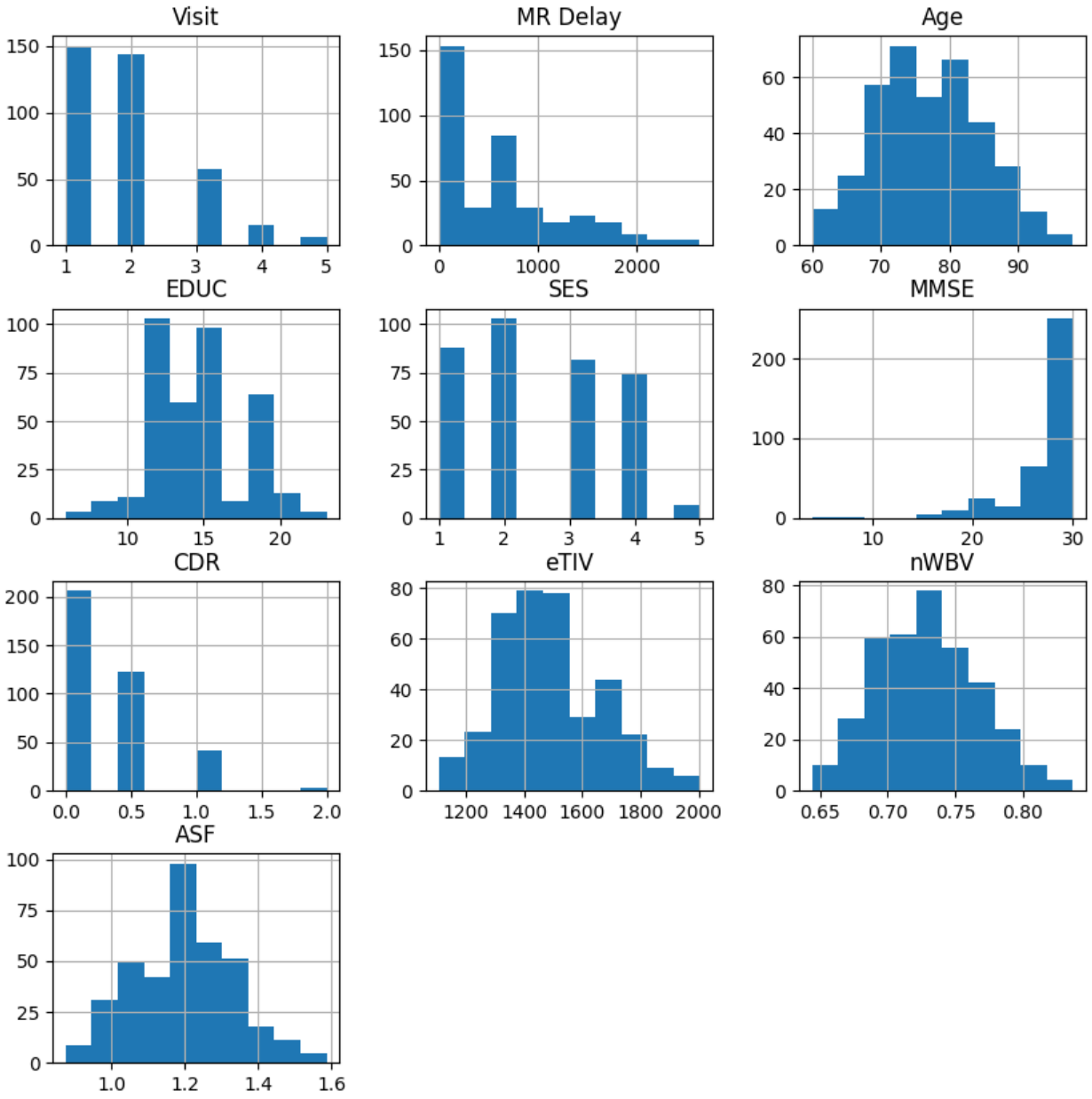
4. DATA SPECIFICATION

4.1 DATA

The 150 members of this cohort, whose ages range from 60 to 96, are collected longitudinally. Over the course of a minimum of two visits separated by a minimum of a year, each patient underwent 373 imaging sessions in total. Three to four separate T1-weighted MRI scans performed during a single scan session are shown to every patient. The subjects cover both right-handed males and women. 72 of the subjects were classified as nondemented throughout the experiment. 51 of the 64 participants had mild to moderate Alzheimer's disease at the time of their initial visits and stayed in that category for subsequent scans. An additional 14 participants were classified at their initial visit.

	count	mean	std	min	25%	50%	75%	max
Visit	373.0	1.882038	0.922843	1.000	1.000	2.000	2.000	5.000
MR Delay	373.0	595.104558	635.485118	0.000	0.000	552.000	873.000	2639.000
Age	373.0	77.013405	7.640957	60.000	71.000	77.000	82.000	98.000
EDUC	373.0	14.597855	2.876339	6.000	12.000	15.000	16.000	23.000
SES	354.0	2.460452	1.134005	1.000	2.000	2.000	3.000	5.000
MMSE	371.0	27.342318	3.683244	4.000	27.000	29.000	30.000	30.000
CDR	373.0	0.290885	0.374557	0.000	0.000	0.000	0.500	2.000
eTIV	373.0	1488.128686	176.139286	1106.000	1357.000	1470.000	1597.000	2004.000
nWBV	373.0	0.729568	0.037135	0.644	0.700	0.729	0.756	0.837
ASF	373.0	1.195461	0.138092	0.876	1.099	1.194	1.293	1.587

The below image visualizes the distribution of numerical data for each column. Each blue bar represents a bin, and the height of the bar indicates the number of observations that fall into each bin.



4.2 FEATURES

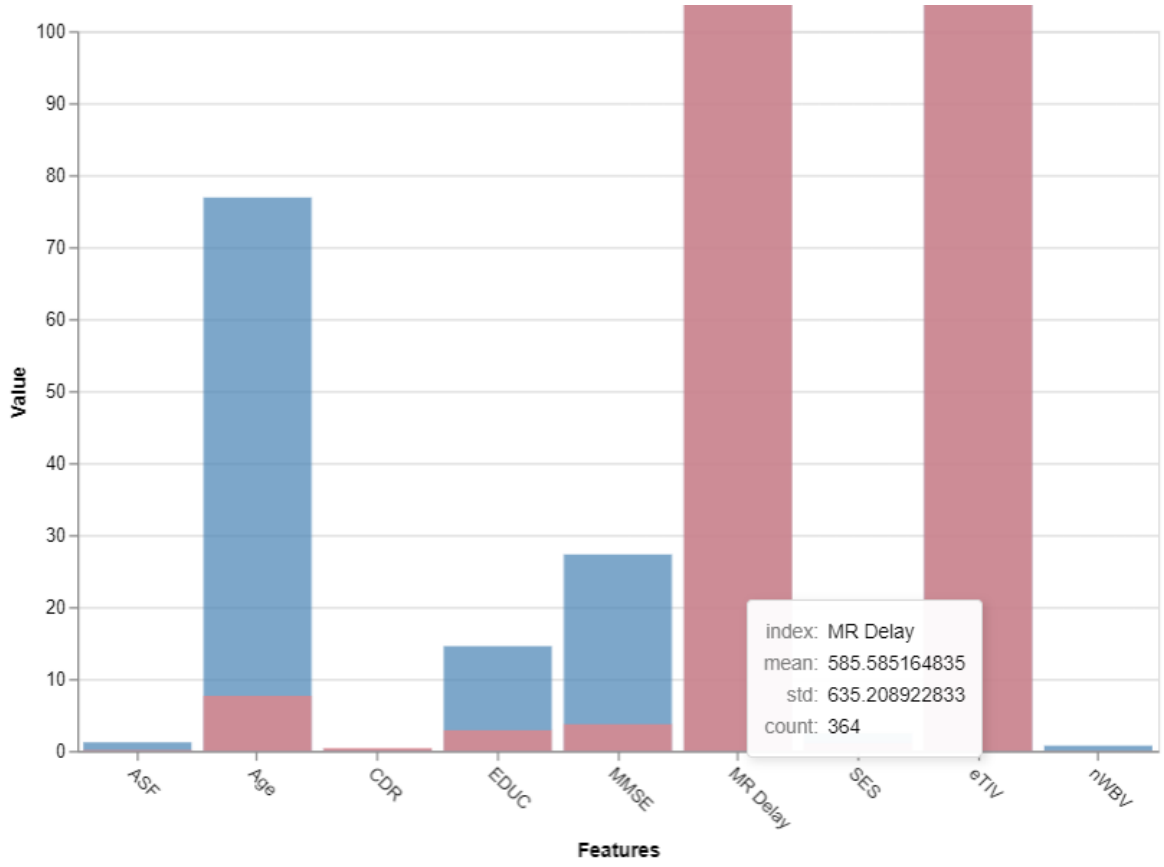
The sample training data included demographic values of Subject ID, MRI ID, Group, Visit, MR delay, Sex, Age, Social Economic Status (SES), Education level (EDUC), MMSE, Clinical Dementia Ratio (CDR), estimated Total Intracranial Volume (e-TIV), normalized Whole Brain Volume (n-WBV) and Atlas Scaling Factor (ASF).

The prediction here is Group. Output labels are Converted, Demented and Non-Demented. Prediction Labels from the model are Demented = 1, Non-Demented = 2. We will be imputing the Converted values.

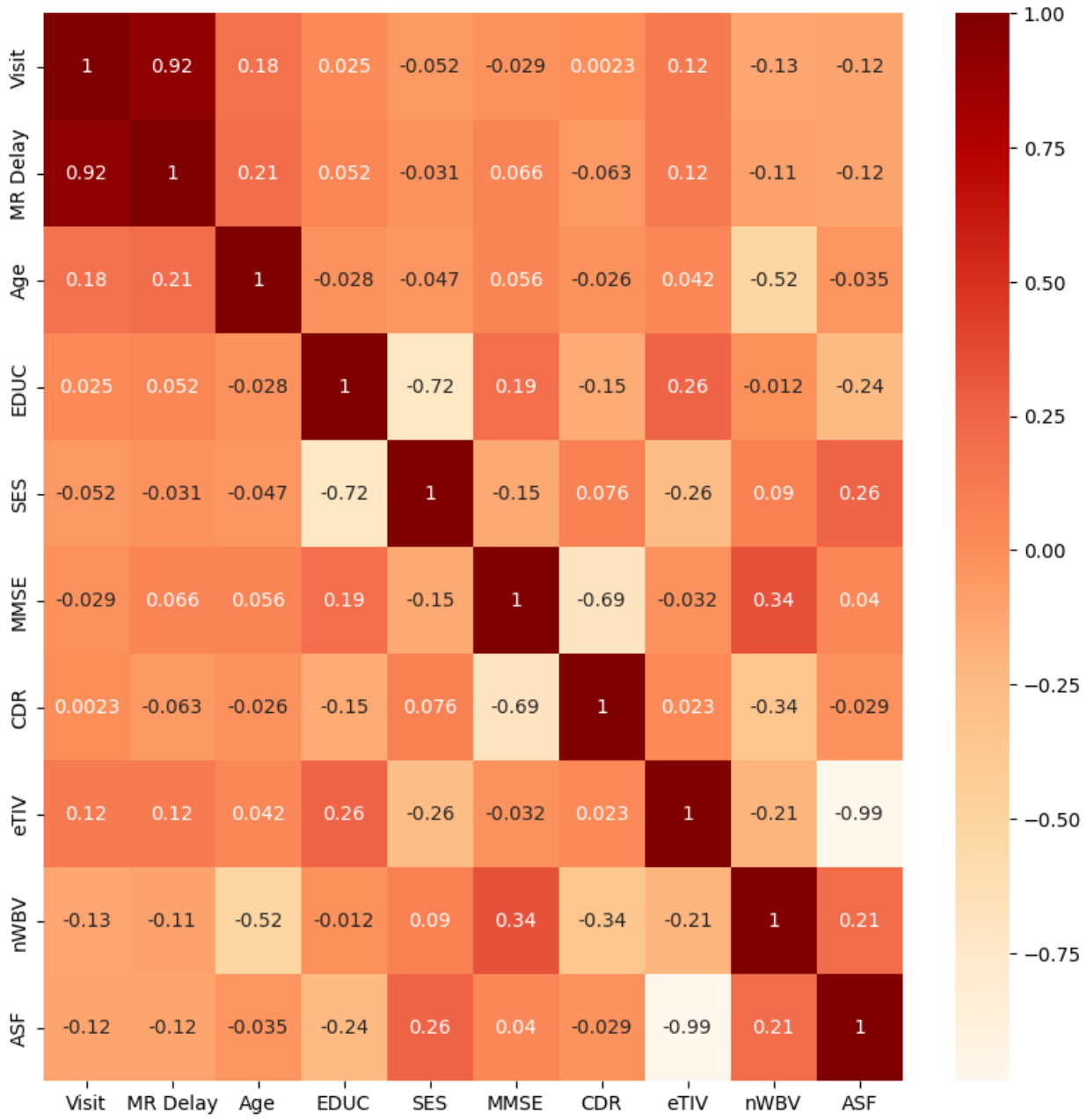
ATTRIBUTE DESCRIPTIONS:

1. Subject ID - A unique Identifier for each subject or individual in the dataset. Represented as a string.
2. **MRI ID**-Identification code for the MRI data associated with each subject. Represented as a string.
3. **Group**- Categorizing the patients into 3 groups Demented, Non demented, or Other(converted). Represented as a string. (Prediction column)
4. **Visit**- Denotes the number of assessments in the form of visits. Represented as an integer.
5. **MR delay**- Time delay or duration between MRI scans. Represented as an Integer
6. **Sex**- Gender of the patient or subject. Represented as M/F(string).
7. **Age**- Age of the subject represented as integer.
8. **Hand**- Right handed or Left-handed. Represented as R/L(string).
9. **EDUC**- Number of years of education. Represented as integer.
10. **SES** - Social Economic Status of the subject. Represented as Integer
11. **MMSE** (Mini-Mental State Examination): Results of MMSE, A widely used cognitive screening test. Represented as an integer.
12.CDR: Cognitive
13. **e-TIV** (Estimated Total Intracranial Volume)-An estimation of the total volume inside the skull. Represented as an Integer value.
14. **n-WBV** (Normalized Whole Brain Volume)- volume measurement of entire brain. Represented as a float value.
15. **ASF** (Atlas Scaling Factor)- A factor used in brain imaging analysis. Represented as a float value.

Here is the mean and standard deviation for features that we shall use for model training.

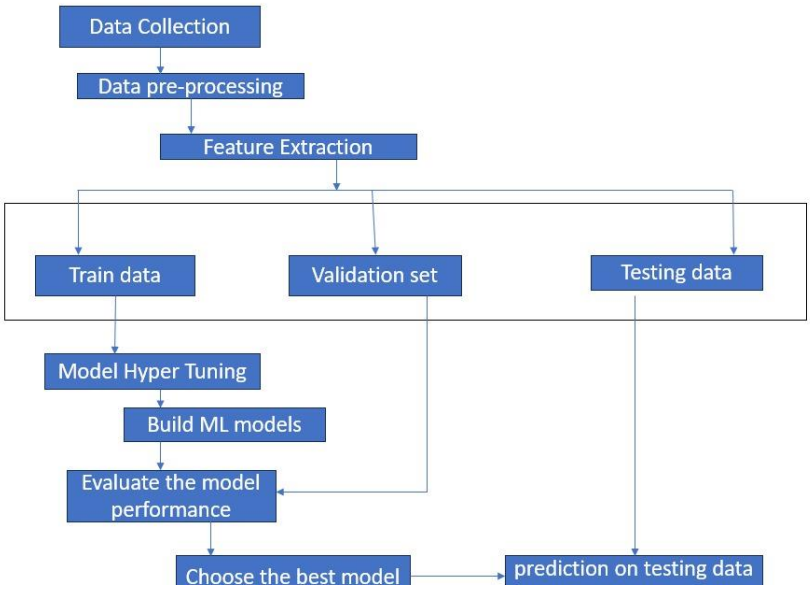


Below is the Correlation matrix for features. It correlates every feature with all other features in the dataset. Used to impute missing values.
From the below matrix MR delay and visit seems to be highly correlated with each other



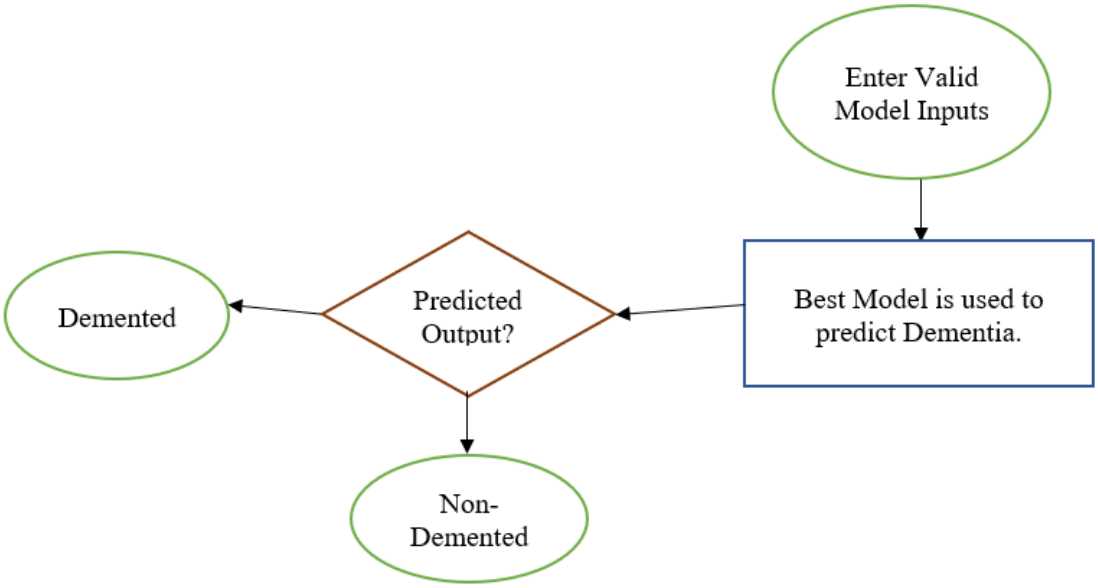
5. Design and Milestones

For every machine learning project, the workflow is almost similar which includes collecting the dataset followed by checking for any missing data and preprocessing the data based on our requirement and then feature extraction which is followed by splitting the dataset into train, test, and validation and later on hyperparameter tuning which is selecting the optimal combination of hyperparameters that minimizes the model's error and building models based on our optimized hyperparameters and choosing the best model followed by fitting the data to our model and testing the model for accuracy and finally using it to make predictions on new data.



5.1 Work-Flow Diagram

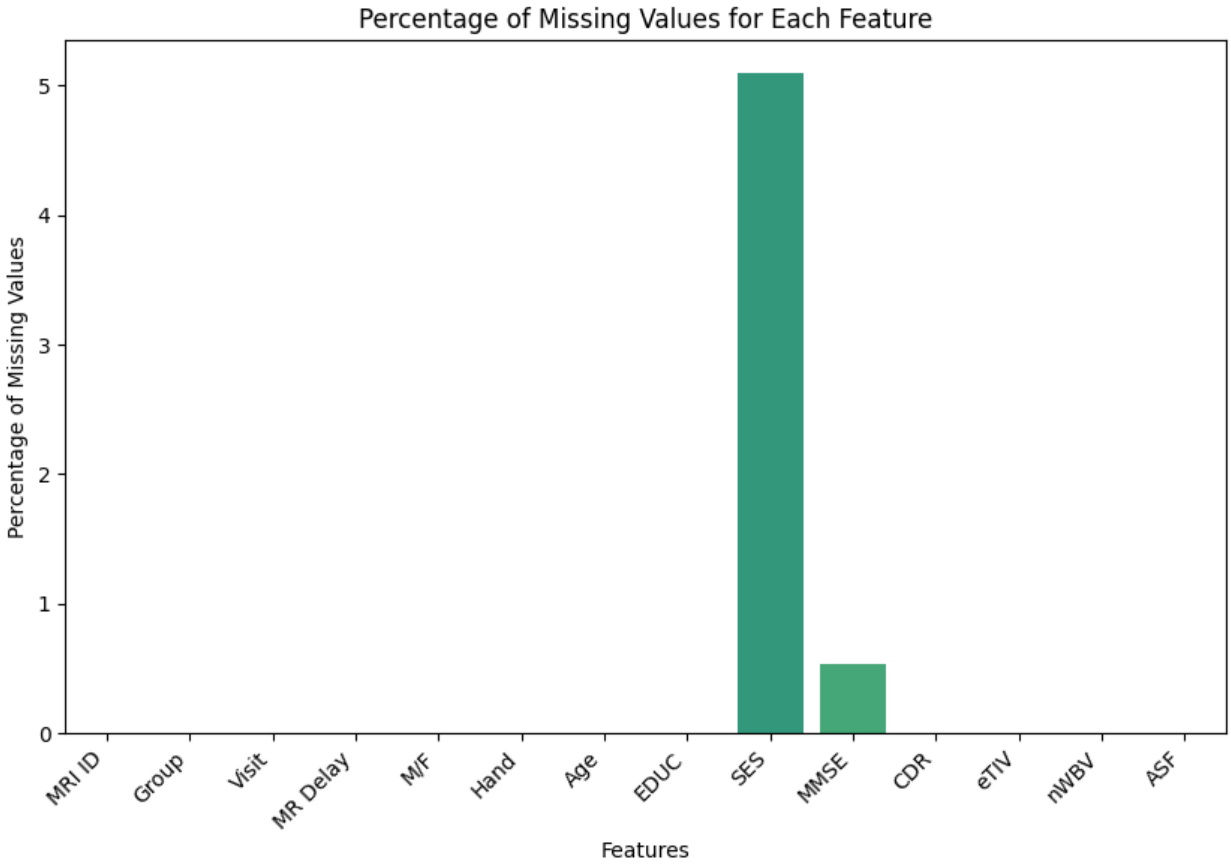
The web application utilizes previous and current research data on dementia related factors to predict dementia using AI algorithms.



5.2 Data Preprocessing

5.2.1 MISSING VALUES IMPUTATION

We have missing values in SES AND MMSE features, and we need to impute them to proceed further. For this, we use KNN Imputer and Impute these missing values by replacing them with KNN Imputer with 5 Neighbors.



The percentage of missing values in SES COLUMN IS 5.09 and MMSE column is 0.53 before imputation.

5.2.2 LABEL ENCODING

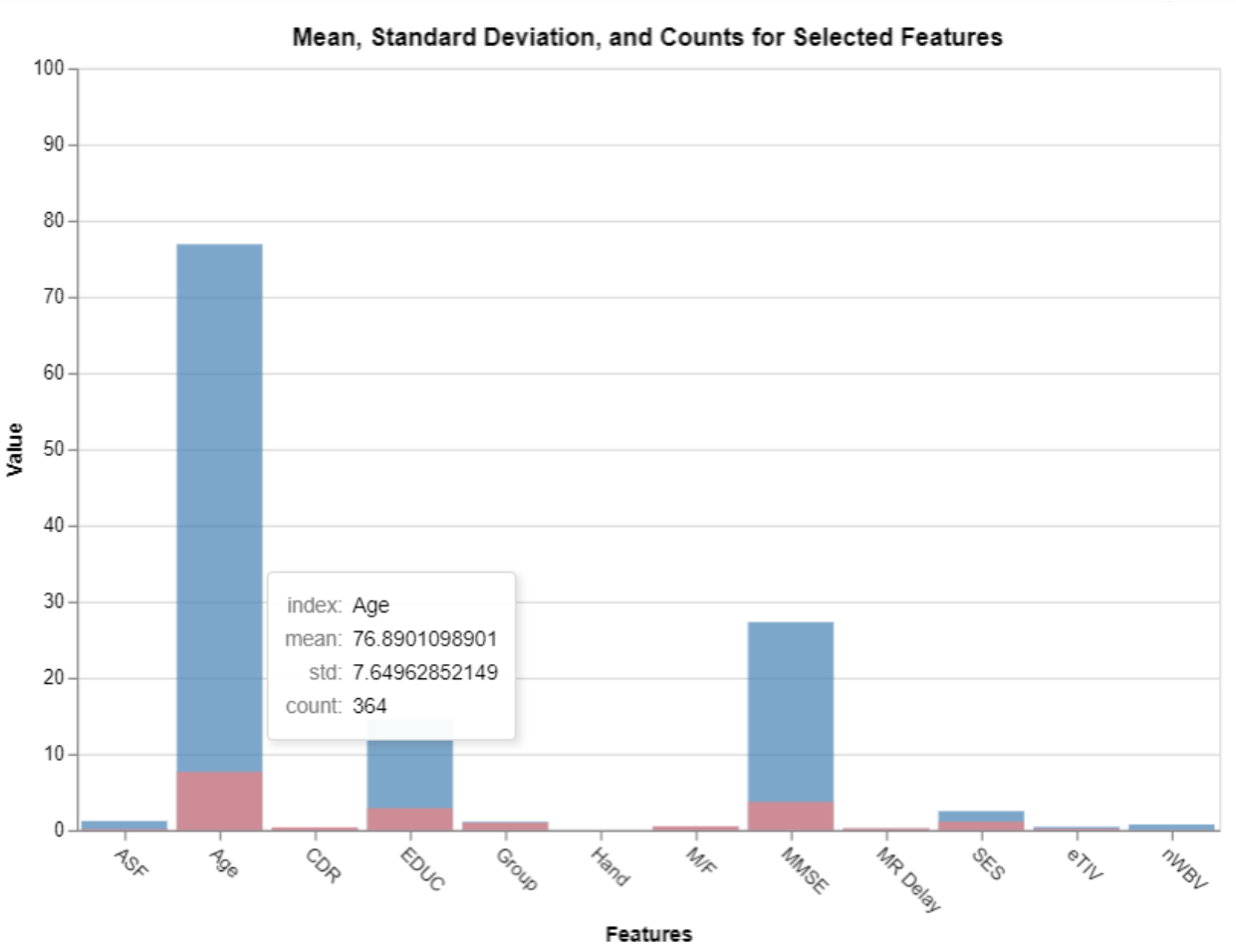
As Many machine learning algorithms work with numerical data, we use Label encoding to convert categorical labels into numerical format.

Sex – M/F male =1, female =2
In Sex column male has been encoded to 1 and female has been encoded to 2

Hand, Right =0, Left =1
In Hand column right hander has been encoded to 0 and left handers to 1

5.2.3 MIN-MAX SCALING

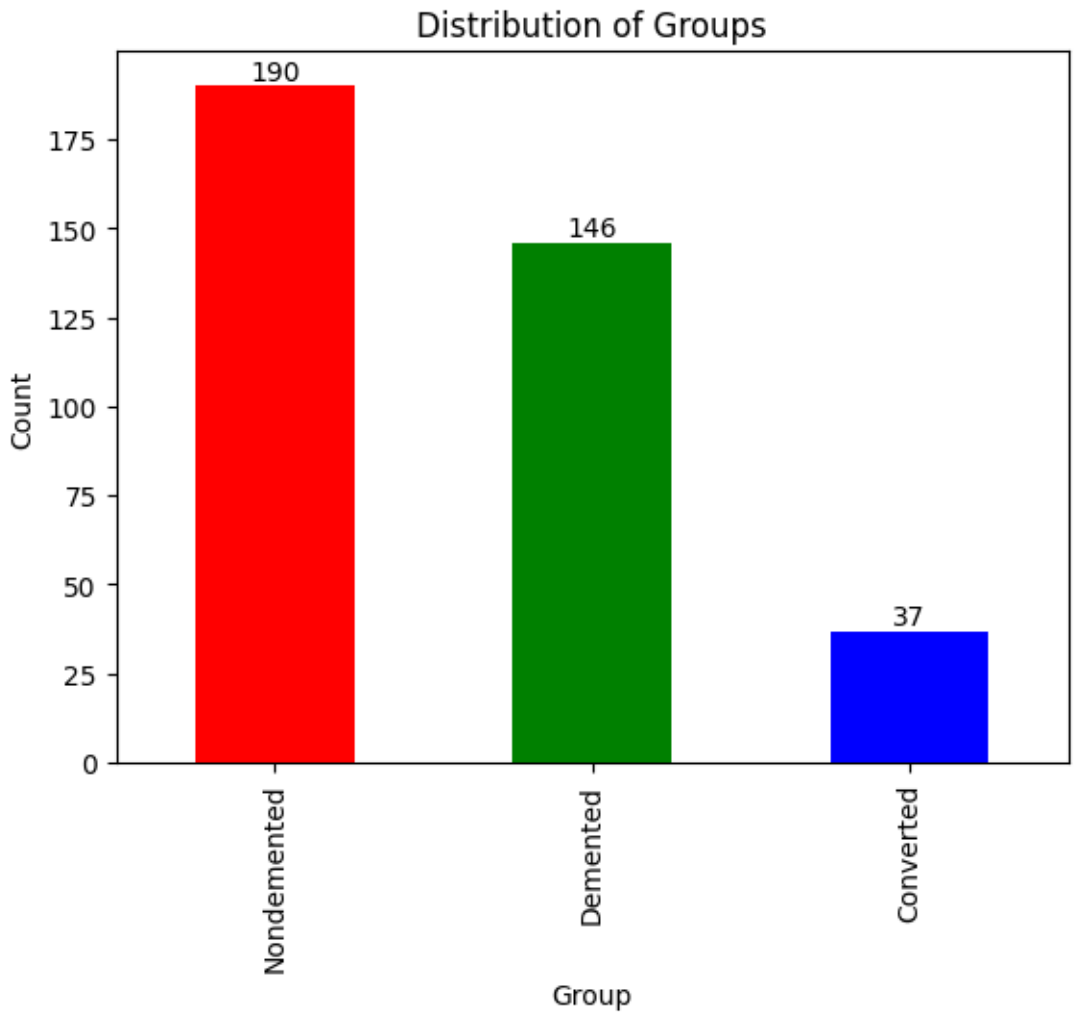
For MR Delay and eTIV features, the standard deviation is too high, so we use min-max scaling to transform these features.



5.2.4 CONVERTED GROUP IMPUTATION

We have preprocessed data for samples with prediction label “Converted” as converted indicates that a patient was initially Non-demented but gradually progressed to Demented. This kind of data would need longitudinal prediction. Hence, we shall avoid these samples by converting the first visit of Converted patients to Non-Demented and Last Visit of such patients is changed to Demented. All the samples in between the first and last are dropped.

Overall, we had 15 subjects with 37 records, after imputation out of 37 records only 9 records have been deleted, rest all have either been changed to Non-Demented or Demented.



5.3 MODELS

Using multiple machine learning models starting from foundational methods like Logistic Regression and Linear Discriminant Analysis, the selection extends to more complex techniques such as K-Nearest Neighbors and Decision Trees. The inclusion of probabilistic models like Gaussian Naive Bayes and Support Vector Machines. Ensemble methods, including Random Forest, XGBoost, Gradient Boosting, and Extra Trees Classifier are used for aggregating weak learners.

5.4 HYPER-PARAMETER TUNING

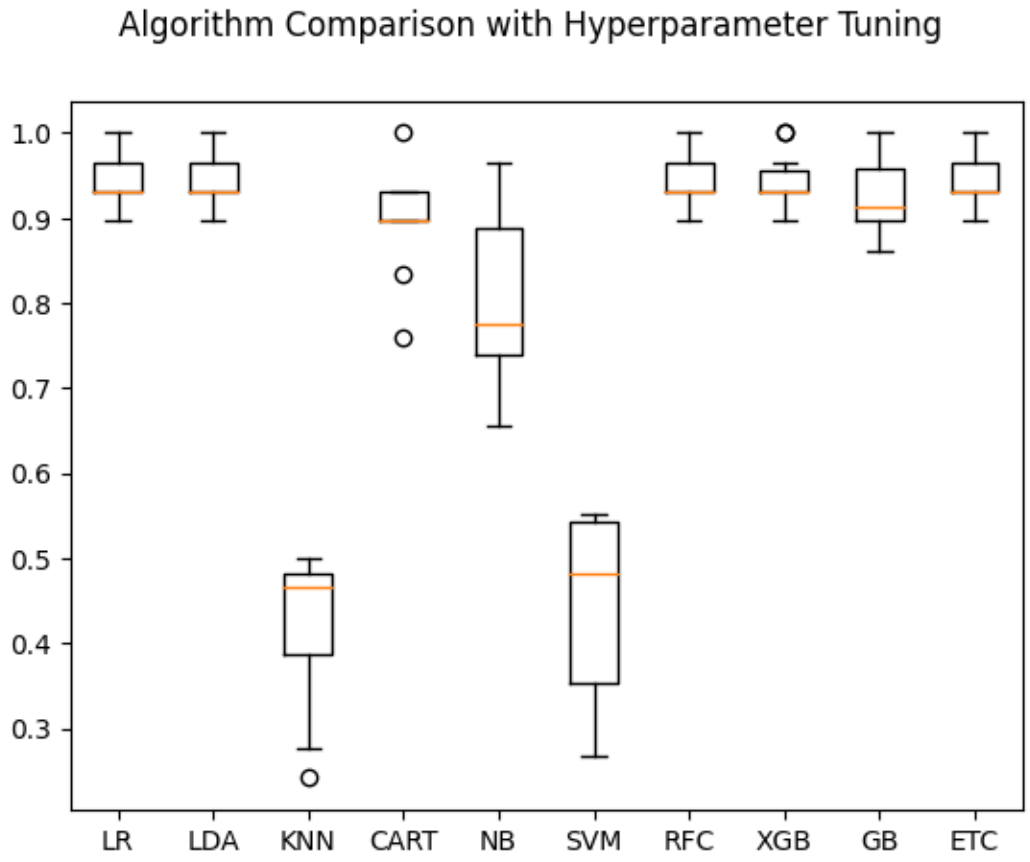
The hyperparameter tuning process applied a methodical approach to each model. For Logistic Regression, an exploration across a range of regularization strengths (C values) and penalties (L1 and L2) were conducted. -Nearest Neighbors underwent tuning on the number of neighbors and different weightings. The Support Vector Machine's tuning uses various combinations of C values, gamma values, and kernel types. Tuning for Random

Forest and Extra Trees Classifier involved optimizing the number of estimators And tree depth parameters. The boosting models - Boost and Gradient Boosting - were fine-tuned on learning rates, the number of estimators, and tree depths. Each GridSearchCV instance performed a 10-fold cross-validation while seeking the parameter configuration that maximized accuracy scores. For models with predefined hyperparameters or no specified tuning parameters, they proceeded without further modification.

5.5 Model Training and Model Performance Evaluation

The training process involved employing a 10-fold cross-validation technique across the collection of tuned models. It iterates over each tuned model, employing KFold cross-validation with 10 splits by using 9 splits to train and the rest split to validate the model.

The model is validated on the leftover split during each iteration and then evaluates model performance for each model by computing the cross-validated accuracy scores. It also calculates the mean and standard deviation of the accuracy scores obtained from the cross-validation for each model and based on these scores we will identify the best model.



5.6 Best Model Selection

Determining the best-performing model from the collection of tuned models based on their mean accuracy scores calculated during cross-validation. Using the **np.argmax** function, it identified the model with the highest mean accuracy score among all the evaluated models and it was selected as the best model.

As per the below image **Linear discriminant analysis** has been selected as the best model due to its high accuracy.

```
LR: 0.986207 (0.022873)
LDA: 0.989655 (0.022080)
KNN: 0.453678 (0.089882)
CART: 0.979310 (0.022873)
NB: 0.986207 (0.016893)
SVM: 0.549195 (0.111034)
RFC: 0.989655 (0.022080)
XGB: 0.989655 (0.022080)
GB: 0.982759 (0.023132)
ETC: 0.989655 (0.022080)
```

5.7 Prediction on Test Set

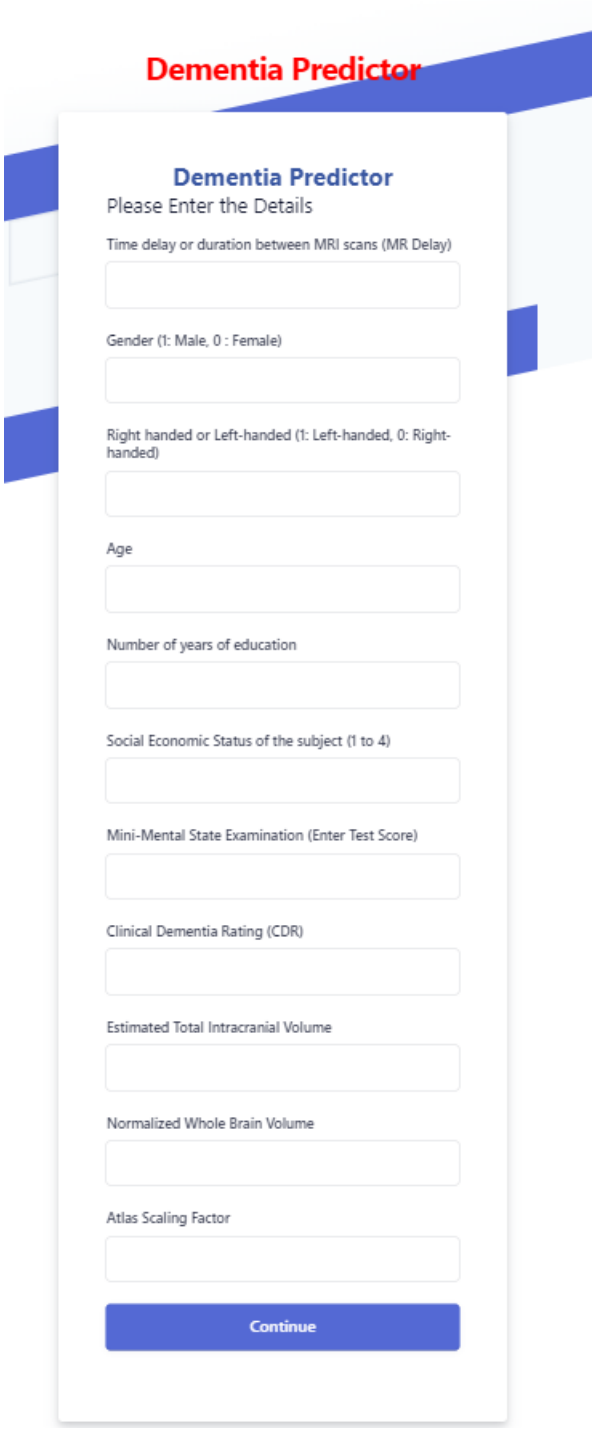
After the best model is selected, it will be fitted or trained on the provided training dataset and tested using the test dataset and later the model is used to make predictions on new data and new user input.

The selected model has an accuracy of 98% when tested on the testing dataset.

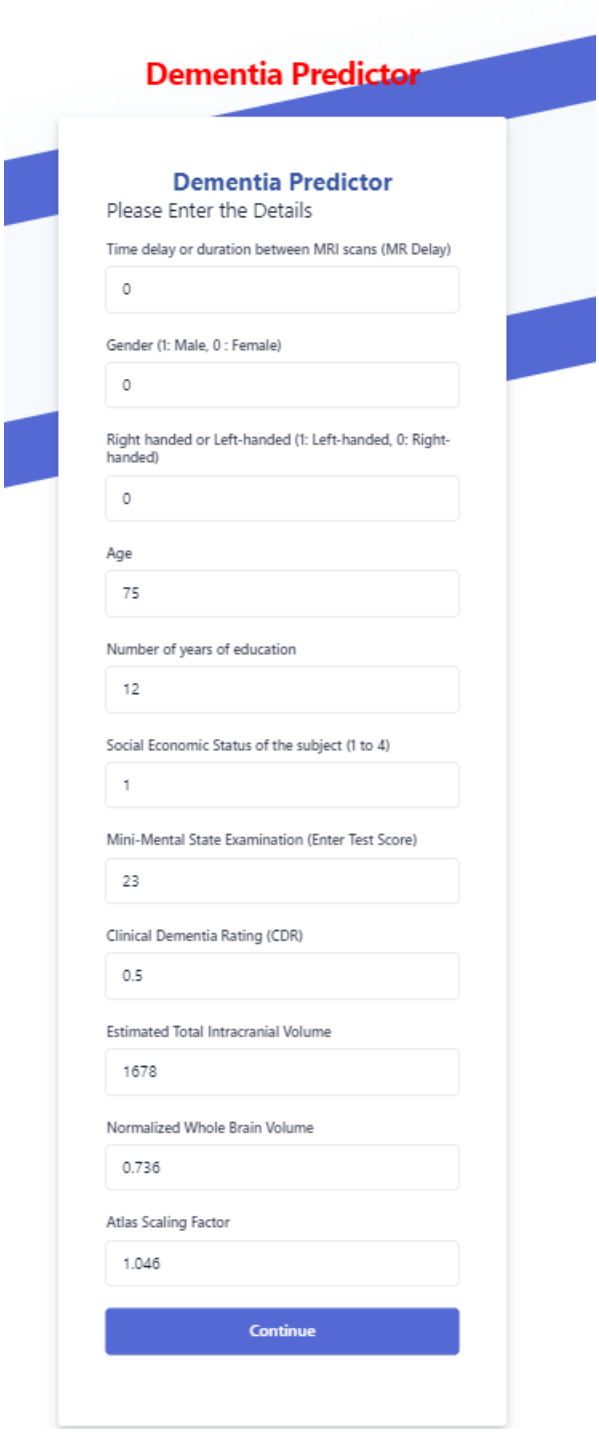
```
Test Accuracy: 0.9863013698630136

Classification Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        25
           1       1.00      0.98      0.99        48

    accuracy                           0.99        73
   macro avg       0.98      0.99      0.98        73
weighted avg       0.99      0.99      0.99        73
```

6. Results

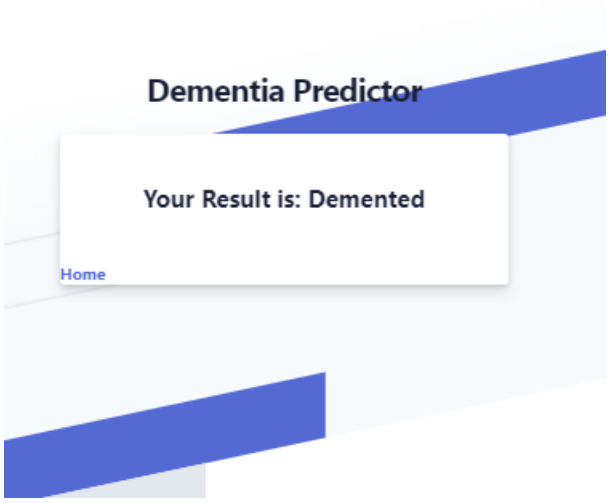
The below image shows the webapp for dementia prediction. All the values mentioned above must be given as input and the model predicts if the person has dementia or not and displays it as the result.


Dementia Predictor
Dementia Predictor
Please Enter the Details
Time delay or duration between MRI scans (MR Delay)
Gender (1: Male, 0 : Female)
Right handed or Left-handed (1: Left-handed, 0: Right-handed)
Age
Number of years of education
Social Economic Status of the subject (1 to 4)
Mini-Mental State Examination (Enter Test Score)
Clinical Dementia Rating (CDR)
Estimated Total Intracranial Volume
Normalized Whole Brain Volume
Atlas Scaling Factor
Continue

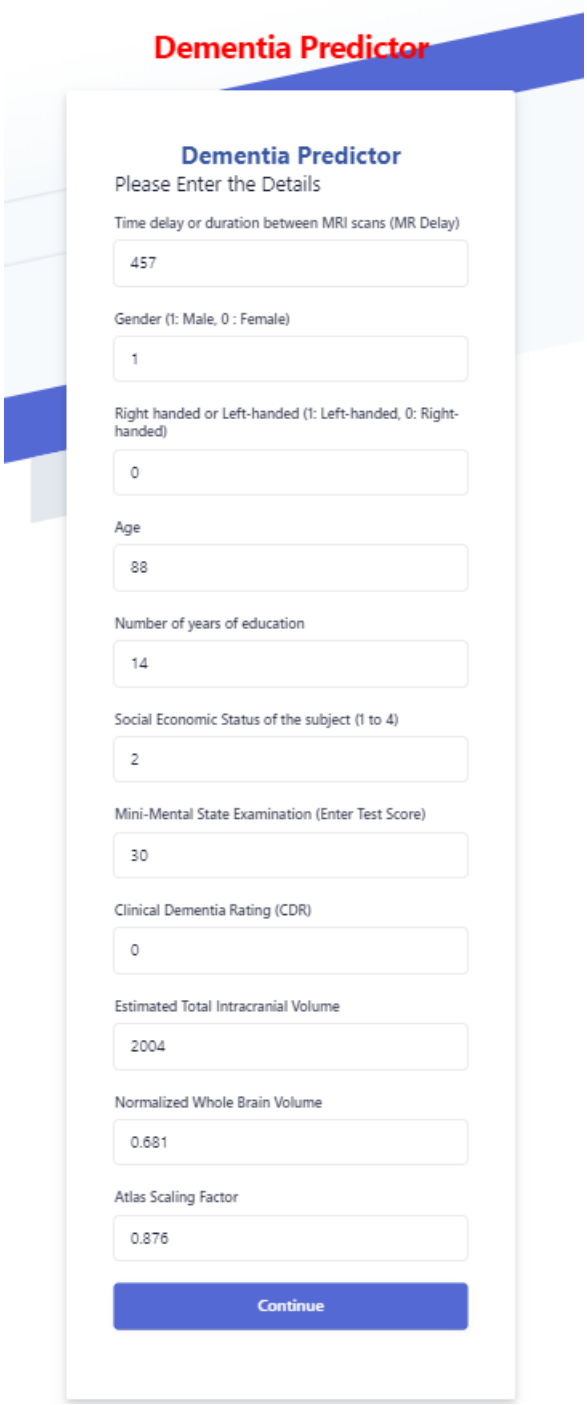
6.1 DEMENTED



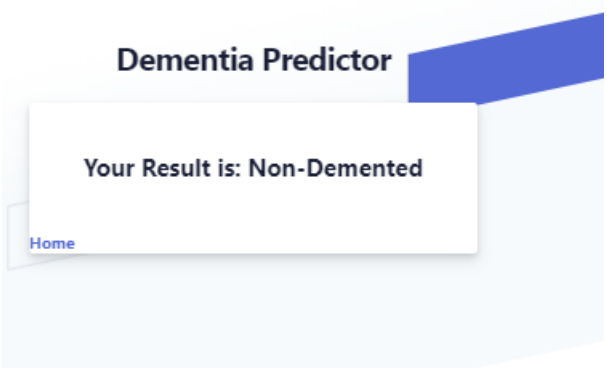
Given the above inputs the predicted output label is demented.



6.2 NON-DEMENTED



Given the above inputs the output variable is non demented.



7. IMPLEMENTATION TOOL

7.1FLASK ARCHITECTURE

1. User: The user enters information on the website in answer to the questions. When the user inputs all his information and clicks the proceed button, the model predicts the result and displays it on the webpage.
2. Flask Application: The flask application is made up of three main parts.
3. Websites: Flask hosts the home page and the result page.
4. Data: The available data is used to train the model.
5. Model: The model is trained using the data using a random forest classifier. It is compiled, run with the flask run command, and then used to generate predictions.

7.2PANDAS

Pandas is a robust Python package with analytic and data manipulation features. It adds two new data structures to the Python language, Series and DataFrame, which are both based on NumPy. Because of this, manipulating data in Python is comparable to and equally potent as manipulating data in R. Pandas greatly facilitates the importing, cleaning, and analysis of data.

7.3NUMPY

NumPy, short for Numerical Python, the Python programming language can now handle enormous, multi-dimensional arrays and matrices along with a wide range of high-level mathematical operations to manipulate these arrays. It is a vital tool for scientific computing and is often used alongside other programs like SciPy (a scientific Python package) and Matplotlib (a charting library).

7.4SEABORN

Matplotlib is the foundation of the Seaborn Python data visualization library. It offers a sophisticated drawing tool for creating informative and engaging statistics graphics. It is connected to Panda's data structures and is constructed on top of the Matplotlib framework. Using Seaborn might help you better explore and comprehend your data because it simplifies a lot of simple visual exploration activities.

7.5 SKLEARN

A free Python machine learning library is called Scikit-learn. It uses several methods, including support vector machines, random forests, and k-neighbors. Two scientific and numerical libraries for Python, NumPy and SciPy, are also supported. For data mining and analysis, it is an excellent tool that can be used by both individuals and businesses.

7.5.1 METRICS

Accuracy Score

The sklearn.metrics module contains a function called accuracy_score that calculates the accuracy classification score. It is the ratio of the total number of input samples to the number of accurate predictions.

Classification report

The sklearn.metrics function classification_report creates a text report displaying the primary classification metrics. It comprises support for every class, recall, F1-score, and precision.

7.5.2 ENSEMBLE TECHNIQUES

Extra Trees Classifier

The sklearn.ensemble module's ensemble learning technique. Several randomised decision trees are fitted on different subsamples of the dataset, and over-fitting is controlled, and the predicted accuracy is increased by averaging.

Random Forest Classifier

A classifier in the sklearn.ensemble module that employs averaging to increase predictive accuracy and manage over-fitting. It fits several decision tree classifiers on different sub-samples of the dataset.

Gradient Boosting Classifier

A classifier in the sklearn.ensemble module that enables the optimization of any differentiable loss function; it constructs an additive model in a forward, stage-wise manner.

7.5.3 MODEL SELECTION

KFold
A model validation method that divides data into train and test sets by providing train/test indices in the sklearn.model_selection module. The dataset is divided into k consecutive folds (by default, without shuffling).

Cross Validation
A cross-validation function in the sklearn.model_selection module assesses a score. The data is divided into a train and test set, the estimator is fitted, and the score times for each split are calculated.

Grid Search CV
A function in the sklearn.model_selection module that determines the optimal model thoroughly.

7.5.4 MODELS

Logistic Regression
A classifier that applies logistic regression, a kind of probabilistic statistical classification model, is found in the sklearn.linear_model module. For issues involving binary categorization, it is helpful.

Linear Discriminant Analysis
In the pre-processing stage of pattern-classification and machine learning applications, the classifier in the sklearn.discriminant_analysis module known as Linear Discriminant Analysis is most frequently employed as a dimensionality reduction algorithm.

K Neighbors Classifier
This classifier, which is part of the sklearn.neighbor's module, applies learning based on each query point's k nearest neighbours, where k is an integer value that the user specifies.

Decision Tree Classifier
This classifier in the sklearn.tree module implements a decision tree, a structure akin to a flowchart where each leaf node denotes a result, each internal node a feature, and each branch a decision rule.

Gaussian Naïve Bayes
A classifier that applies the Gaussian Naive Bayes algorithm for classification in the sklearn.naive_bayes package. It is assumed that the features have a Gaussian likelihood.

Support Vector

A classifier in the sklearn.svm module that applies the multi-class classification technique known as "one-vs-one," wherein k(k-1)/2 binary classifiers are built, and each one is trained using data from two classes.

XG Boost Classifier

An XGBoost model implementation classifier was found in the xgboost module. A distributed gradient boosting library optimised for maximum efficiency, versatility, and portability is called XGBoost.

8. PROJECT RUN INSTRUCTIONS

To run the Flask application, follow the below instructions:

1. Extract the project zip folder.
2. from the file directory of the project open the terminal.
3. Run the below commands.
   i. py -m venv env (to set up the Python environment)
   ii. Set-ExecutionPolicy Unrestricted -Scope Process
   iii. .\env\Scripts\activate (to activate the environment)
   iv. flask run (to run flask application)
4. We can run the application by clicking on the local host path that is displayed on the terminal.

9. PROJECT MANAGEMENT

1. **Neha Goud Baddam (Nehagoudbaddam@my.unt.edu)**

Role: Model fine-tuning, Model Performance Evaluation. (for XGBoost and Random Forest), Visualizing Model performance, Reporting Model performance for all the considered models, documentation

2. **Purandhara Maharshi (purandharamaharshichidurala@my.unt.edu)**

Role: Data Preprocessing and Feature Selection, Visualizing Data mean and Standard Deviation, documentation

3. **Sri Harsha Gurram (sriharshagurram@my.unt.edu)**

Role: Model fine-tuning, Model Performance Evaluation. (for Linear Regression), Visualizing Model performance, documentation

4. **Tejaswi Reddy Siddareddy (TejaswiReddySiddareddy@my.unt.edu)**

Role: Model fine-tuning, Model Performance Evaluation. (for SVM) , Visualizing Model performance, documentation

10. REFERENCES

1. Battineni, G., Chintalapudi, N. and Amenta, F. (2019). Machine learning in medicine: Performance calculation of dementia prediction by support vector machines (SVM). *Informatics in Medicine Unlocked*, 16, p.100200. doi:https://doi.org/10.1016/j.imu.2019.100200.

2. https://www.kaggle.com/code/gkitchen/predicting-dementia

3. https://www.kaggle.com/code/bayunova/dementia-prediction