Journal of King Saud University – Computer and Information Sciences 34 (2022) 2688–2706

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

An Improved Multi-Modal based Machine Learning Approach for the Prognosis of Alzheimer's disease

Afreen Khan, Swaleha Zubair *

Department of Computer Science, Aligarh Muslim University, Aligarh, India

ARTICLE INFO

Article history:
Received 30 September 2019
Revised 10 January 2020
Accepted 2 April 2020
Available online 23 April 2020

Keywords:
Alzheimer's disease
Dementia
Machine learning
Pattern recognition
Pipeline

ABSTRACT

Alzheimer's disease (AD) is the most common type of neurological disorder that leads to the brain's cell death overtime. It is one of the major important causes of memory loss and cognitive decline in elderly subjects around the globe. Early detection and streamlining of diagnostic practices are the prime domains of the interest to the healthcare community. Machine learning (ML) algorithms and numerous multivariate data exploratory tools have been extensively used in the field of AD research. The primary purpose of this study is to present an automated classification system to retrieve information patterns. We proposed a five-stage ML pipeline, where each stage was further categorized in different sub-levels. The study relied on the Open Access Series of Imaging Studies (OASIS) database of MRI (Magnetic Resonance Imaging) brain images for the analysis. The dataset comprised of 343 MRI sessions involving 150 subjects. Three different scores namely, MMSE (Mini-Mental State Examination), CDR (Clinical Dementia Rating), and ASF (Atlas Scaling Factor) were used in the analysis. The proposed ML pipeline constitutes a classifier system along with data transformation and feature selection techniques that have been embedded inside an experimental and data analysis design. Performance metrics for Random Forest (RF) classifier showed the highest output in the classification accuracy.

1. Introduction

Recent advancement in the health sector has tremendously contributed in improving the living standard of human beings. However, sedentary lifestyle and extra comfort, in turn led to various physiological diseases. For example, dementia and other related neurological disorders influence elderly people mostly. Dementia is a chronic/progressive disease, in which cognitive functions, i.e. ability to process thoughts, deteriorate slowly ahead of what could be expected from natural ageing (Robinson et al., 2015). Thinking ability, memory, comprehension and orientation, learning capacity and calculation, judgement and language, and other cognitive abilities also get affected under the influence of dementia (Chapman et al., 2006).

Under the Mental Health Gap Action Programme, (World Health Organization, 2008), declared dementia as a priority health condition that needs attention (World Health Organization, 2008). Moreover, prevalence and incidence prognosis studies performed by WHO claim that the number of individuals with dementia will grow incessantly, mostly amongst older people. According to global statistics of Alzheimer's Association 2019 report, 47 million people worldwide are living with AD; making it a significant public health problem in today's society (Alzheimer's Disease Facts and Figures, 2019). Moreover, the total number of cases is likely to reach 76 million by 2030. The total number of new cases of dementia each year worldwide is nearly 7.7 million, implying one new case every four seconds (Prince et al., 2013). The growing trend in AD cases is due to numerous reasons, such as aging, population growth, and the changing behavior associated with social and economic development.

Alzheimer's disease (AD) is the most common type of dementia and contributes to 60–70 percent of dementia cases almost. There exist various kinds of dementias, likewise, dementia with lewy bodies, vascular dementia, and a collection of certain diseases that may cause front temporal dementia (Nichols, 2019). AD is degenerative in nature, which is indicated by a progressive slow weakening of the cognitive functions that worsen with time. It takes around

* Corresponding author.
E-mail address: swalehazubair@yahoo.com (S. Zubair).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2020.04.004
1319-1578/© 2020 The Authors. Published by Elsevier B.V. on behalf of King Saud University.

20 years or more before the actual symptoms appear, with minor alterations in the brain that remain hidden to the affected individual (Alzheimer's Disease Facts and Figures, 2019). The development of AD often begins with trivial symptoms and eventually end up in acute brain damage. Usually, the symptoms arise when the nerve cells (neurons present in the brain) involved in memory (cognitive function), thinking, and learning underwent autophagy (Robinson et al., 2015). With time, these symptoms increase and begin to interfere with individuals' capability to carry out day-to-day activities. Ultimately, it reaches a point where it becomes fatal. Hours of care provided to people with AD or related dementias estimated by Alzheimer's Association is of 604 billion value (Alzheimer's Disease Facts and Figures, 2019). Analysis of such data tells that there is a serious need for remedial action as AD has become a major public health concern.

Currently, there is no treatment available that can reverse or end the development of this disease. However, the disease can be controlled if diagnosed in the earlier stage. Presently available tools do not assure its diagnosis with 100 percent certainty. Brain imaging such as MRI scans together with the clinical assessment that test for symptoms of memory impairment are generally employed to diagnose patients with AD (Pellegrini et al., 2018). Absolute diagnosis can only be achieved after the autopsy of patients' brain tissues. In recent years, advancements in neuroimaging provide opportunities to have better insight regarding neurological-related complexities that help in the early and correct detection of AD (Hanyu et al., 2010; Gray et al., 2012; Liu et al., 2014).

Diagnosis of AD entails numerous medical tests that in turn produce a large amount of multivariate heterogeneous data (Khajehnejad et al., 2017). To compare, analyze and visualize this data manually, seems to be a difficult task due to the heterogeneity in medical tests. Conventionally, Alzheimer's diagnosis is achieved by carrying out a neuropsychological examination that supports structural imaging, such as Magnetic resonance imaging (MRI). It is a promising tool extensively used in AD-associated studies due to its non-invasive characteristic and absence of pain to patients (Lama et al., 2017; Shamonin et al., 2014). Additionally, MRI-based analysis can be combined later with additional medical investigations to obtain a consistent classification of data (Klöppel et al., 2012; O'Brien, 2007).

The precise classification of dementia can avoid patients to undergo unnecessary treatments. Thus, accurate diagnosis of AD and correct classification of patients is the subject of research, as of now. Machine learning (ML), because of its distinctive advantages in critical feature detection from multivariate AD datasets, is widely considered as the approach towards pattern classification and forecast modelling of AD (Khan and Zubair, 2018). Several ML algorithms have been successfully applied to differentiate AD patients from that of elderly control (otherwise healthy) subjects by employing various biomarkers (Falahati et al., 2014). Among them, ML tools and techniques have been fabricated to distinguish MR images from healthy versus inflicted patients (Shao et al., 2014; Nasiri et al., 2014). These approaches need training set of the population that contains well-categorized subjects i.e. healthy and inflicted patient along with correct diagnosis so as to group the newly tested subjects into one class of the training set (Alam et al., 2017). A given analytical strategy can only be successful when the classifier is able to predict the correct classification of unseen data.

In the present study, we propose a ML pipeline for identifying demented and non-demented patients. The proposed hypothetical model envisages MRI-based data of 343 subjects, with a demented and non-demented group of individuals, with an additional refinement that contains the preclinical features of the disease. The present study emphasizes on comparing and presenting the classification methods efficiently by working on a relatively smaller dataset robustly. As an outline, the pipeline constitutes a classifier along with a comprehensive model evaluation system together with data transformation and feature selection techniques that have been embedded inside an experimental and data analysis design. The end result of the pipeline strategy is a list of accuracies with the maximum ability to detect and classify the right set of patients as shown by the Random Forest, an ensemble classifier. A noteworthy point of this workflow is the individuality of each component i.e. independence among each of its modules, such as the classifier, the feature transformation and extraction algorithm, the data exploration procedure, the data segregation method, and the modelling technique can all be exchanged with substitute methods.

To further validate our approach, we altered various experimental conditions by changing the classifier parameters through various involved stages. This paper has been organized in several sections. The subsequent section summarizes the methods and results of previous research performed on AD diagnosis. Section 3 illustrates the materials and methods. Section 4 describes the proposed method: the pipeline. Section 5 presents the experimental results. Finally, Section 6 includes discussion followed by a conclusion along with outlining future directions.

2. Materials and methods

2.1. Dataset

The proposed pipeline was built employing MRI data acquired from the Open Access Series of Imaging Studies (OASIS) database. We confined our study on the longitudinal collection of MRI data in both demented and non-demented older adults. OASIS is a public domain database that compiles MRI datasets and makes them available to the scientific communities.

2.2. Details of the acquitted MR images

All structural MRI scans are T1-weighted and were acquired on a 1.5 Tesla Vision Scanner. A high-resolution MP-RAGE (Magnetization Prepared Rapid Acquired Gradient Echo) sequence were used for analyzing the classification of 150 subjects for 343 MRI sessions. The MRI acquisition details have been reported in Table 1 (Marcus et al., 2010).

2.3. Subjects

The dataset consisted of 343 sessions performed on 150 subjects, aged between 60 and 96 years. It includes a longitudinal-section of the studied population. Table 2 lists the demographics of all these subjects.

Table 1
MRI acquisition details.

MR characteristics	Values
TR (repetition time)	9.7 msec
TE (echo time)	4.0 msec
Flip angle	10°
TI	20 msec
TD	200 msec
Orientation	Sagittal
Thickness	1.25 mm
Gap	0 mm
Slice number	128
Resolution	256 × 256 (1 × 1 mm)

Table 2
Summary of demographics status of the subjects included in the study.

No of subjects	78	72
	Demented (D)	Non-demented (ND)
Male	40 D	22 ND
Female	38 D	50 ND
Age:		
Range (in years)	60–96	
Mean ± SD	77.01 ± 7.64	
Median	77.0	

2.4. Dataset description

The dataset includes 373 observations and 15 attributes (features). A detailed description of the attributes is listed in Table 3. In the dataset, the target variable, ‘Group’ is a binary classifier that specifies the status of the patient with or without dementia. In the present study, we used various scores to determine the state of the healthy vs inflicted brain. The scoring rules are listed in Table 4.

The subjects included in the study have been clinically diagnosed with very mild to moderate state of Alzheimer’s disease. The subjects comprised of both genders and all right-handed. All patients underwent same analysis procedure. All control subjects underwent a neuropsychological assessment including the Mini Mental State Examination (MMSE) and other tests. All control subjects had an MRI examination using the same scanner and the same procedure as AD patients.

At first, the dataset contains 373 MRI sessions which include non-demented, demented and converted set of patients. On their first visit i.e. Visit = 1, certain patients were categorized as *Non-demented* and on a later visit, these patients were diagnosed with Dementia. Hence, they were grouped into *Converted* patients. Thus, only the subjects with Visit = 1 are being taken into account during the course of this study. Table 5 shows the number of patients for each category, making it to a total of 150 subjects that have been studied under this analysis.

3. Workflow design

Recently, ML has made notable advancements in numerous application domain, thereby, encouraging its demand efficiently by novices in ML (Feurer et al., 2015). In addition, a powerful ML system is ought to solve the fundamental challenges by determining a specific ML algorithm to apply on a dataset, in what manner the preprocessing should be done, and how to tune its hyperparameters.

Table 3
Detail of dataset attributes.

Attribute name	Attribute description
Age	Patient’s age during the scanning
ASF	Atlas Scaling Factor
CDR	Clinical Dementia Rate score
EDUC	Educational years of a patient
eTIV	experimental Total Intracranial Volume result
Group	Demented, Non-demented or converted
Hand	Right-handed or left-handed
M/F	Patient’s gender
MMSE	Mini Mental State Examination score
MR Delay	Magnetic Resonance (MR) delay is the delay time that is given before the image procurement is performed in real.
MRI ID	Imaging identification number of each patient
nWBV	normalized Whole Brain Volume result
SES	Socio Economic Status of a patient
Subject ID	Patient’s identification number
Visit	Number of visits of each patient

Table 4
Scoring rules.

Attribute name	Description
SES (Socio-Economic Status)	According to the Hollingshead Index describing the Social Position, SES is classified into groups of the highest status and lowest status (Lynch et al., 2000).
1	Highest status
0	lowest status
MMSE (Mini-Mental State Examination)	In the MMSE score, the values range from 0 to 30 (Arevalo-Rodriguez et al., 2015).
Below 10	Extreme impairment
10–19	Moderate dementia
19–24	Early-stage Alzheimer’s ailment
25 or higher	Normal
CDR (Clinical Dementia Rating)	It is obtained from a semi-structured discussion with the patient. The scores range from 0 to 3 (Morris, 1993).
0	None
0.5	Very mild
1	Mild
2	Moderate
3	Extreme dementia

Table 5
Number of patients for each group on the first visit.

Group (target variable)	Number of patients
Non-demented	72
Demented	64
Converted	14

In this paper, we propose a sequential approach for AD classification as a tool to diagnose AD. The proposed model learns data using a ML algorithm and classifies data into healthy or non-healthy AD patient. We used Spyder platform of Anaconda as an experimental environment, which employs Python libraries. The Spyder platform exhibits a well-defined skeleton for developers to process, build and assess their models. Python is an interpreted and higher-level programming language encompassing dynamic semantics.

4. The pipeline: Architecture of the proposed model

Pipelines operate by taking into account a linear sequence of data transformation, which was linked together in such a way that culminates in a model that can be assessed and implemented further. The objective was to make certain that all steps in the pipeline are followed. Moreover, it was confined to the available dataset for the evaluation, for instance, the train dataset, test dataset, or cross-validation dataset.

The proposed pipeline is a five-level sequential model. Each level consists of sub-levels, which is maintained as a linear sequence. Fig. 1 illustrates the various steps of the proposed method.

Following is the working program of the model employed (Fig. 1).

The first stage comprised of data preparation, which involves the pre-processing of MRI data. This involved the following steps: data collection, data visualization, feature selection, and data transformation. This initial level dealt with the data in a way that made it simpler. The approach handles missing data, remove existing outliers, normalization to a specific range, and selection of features on the basis of their influencing power. Furthermore, the data visualization helps us to envisage the raw data at a broader level, demonstrating the distribution, correlation, skewness among the data. The output of the first level serves as the input for the second stage i.e. the clean data. The second level comprised of the data

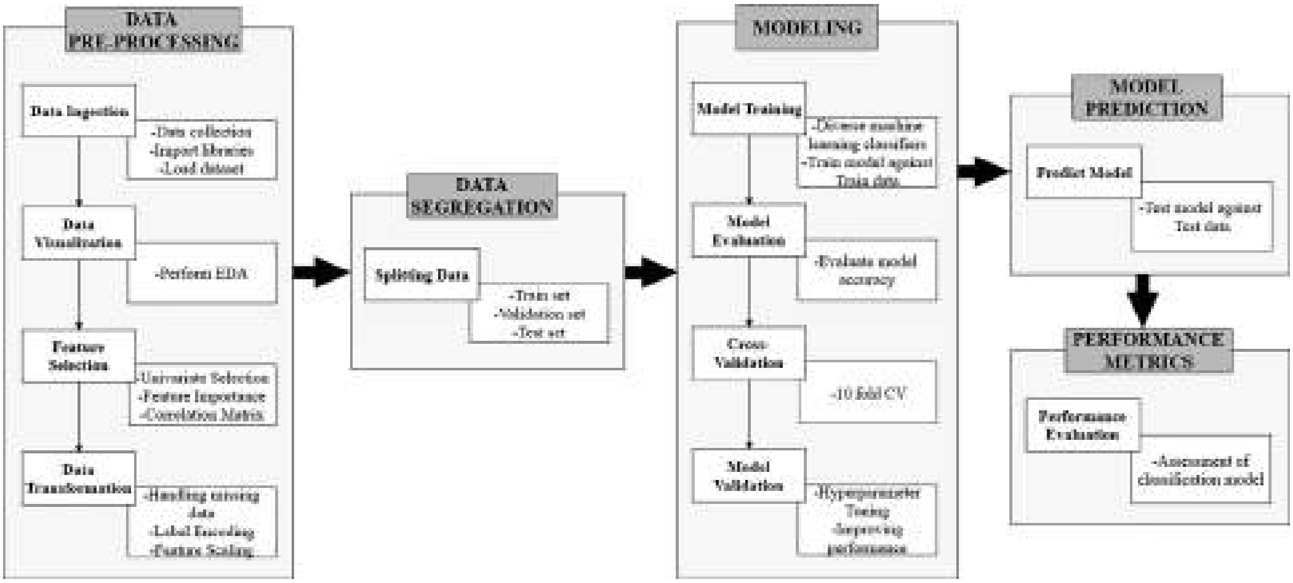


Fig. 1. Recommended pipeline of the proposed model.

segregation, which involved the splitting of the dataset into train data, test data and validation data. The split data is then used in the third level for model building. The third stage further involved 4 sub-levels: model training, model evaluation, cross-validation and hyperparameter tuning via model validation. This included the actual working of ML, where various ML classifiers are trained, the model is evaluated on the basis of the accuracy generated, performing cross-validation and tuning of parameters in order to improve the accuracy. The model evaluation was performed by employing various ML algorithms for learning and classification of data for model generation. Next level is the model prediction level, which evaluates the model generated in the third step. It predicts the model on the test set, thus classifying the group into AD or non-AD patients respectively. The last stage i.e. the fifth one was the performance evaluation level, which provided insights of the model by illustrating the model performance graphically. The 5-level workflow design of the pipeline was maintained by following the order sequentially.

In the following sections, we discuss the above-introduced approach in detail.

4.1. Data pre-processing

Data are the fuel of technology. The vast majority of the data that exists as of today are inconsistent, noisy and lack certain trends and behavior, as it consisted of numerous errors which make it unstructured (Sivarajah et al., 2017). The removal of such noise occurrences has remained the most difficult and challenging task in inductive ML (Teng, 1999). In order to convert this unstructured data into organized data i.e. structured data, data preprocessing step is applied (Khan et al., 2019). Also known as *Data Preparation* because it is the first and the most significant stage headed for structuring a working ML model. Furthermore, it significantly affects the generalization performance of a ML algorithm (Kotsiantis et al., 2006). It incorporates 4 fundamental levels which is accomplished by maintaining a sequential order as illustrated in Fig. 2.

In this step, the raw MRI data was made to run through the 4 steps (mentioned in the above figure). It is followed by an analysis of dataset to figure out how to change it into useful information that can be plugged as an input into the chosen model. This procedure is followed in a linear manner, yet it is in all respects prone to be iterative with many loops till a clean data is accomplished.

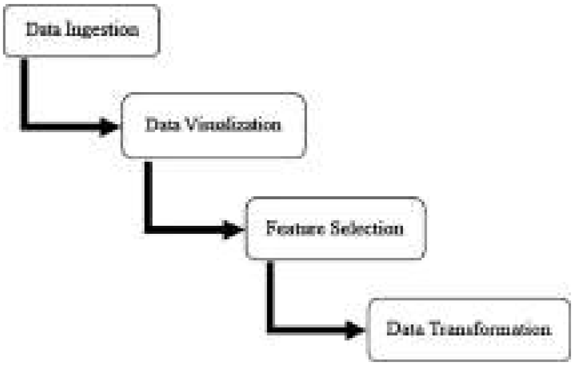


Fig. 2. Schematic representation of the data pre-processing stage.

4.1.1. Step 1: Data ingestion

Data ingestion is a procedure of collecting data and prepare it for analysis. It consists of 3 broad steps, called as ETL (Extract: data acquisition from its location, Transform: data cleaning and normalization, Load: data is put in a database from where it can be explored further). We broke down this step into 3 collective sub-steps, described as follows:

1. **Gathering of data**: Collected the required data from the OASIS database.
2. **Importing the libraries**: A library is a collection of methods and functions that enables to execute various actions. Three essential libraries we employed are- Numpy, Matplotlib, Pandas and Seaborn. Numpy is used for scientific computing, Matplotlib is the library that is used for plotting the charts and graphs, Pandas is used for data analysis and data manipulation, and Seaborn is the best tool available for data visualization (upgraded version of Matplotlib).
3. **Importing the dataset**: We imported the dataset into the Spyder platform of Anaconda environment in the CSV format. The Pandas library was used to import the dataset.

4.1.2. Step 2: Data visualization

Data visualization relates to the very notion of performing Exploratory Data Analysis (EDA). An EDA is a data analysis technique which itself is a collection of various tools and methods that are employed to gain the graphical and statistical insight of the available data. NIST/SEMATECH e-Handbook of Statistical Methods defines EDA as a data visualization process that expands understanding of the data, uncovers underlying structure, detects inconsistencies and outliers, root out vital features and determines best factor settings (NIST/SEMATECH e-Handbook of Statistical Methods, 2003).

In the data ingestion step, we focused on determining a correlation between different MRI test features and the patient's classification group by performing EDA in advance before moving to the next step. It helped us in understanding the data sub-classification and enabled in selecting the correct analysis method for the model at later stage. In order to gain the overall insight of the MRI dataset, we performed numerous EDA methods, out of which 3 are discussed below.

1. **Uncovering Outliers**: Outlier is a data point that varies significantly from other observations (Kwak and Kim, 2017). It presents the distribution of quantitative data in a way that aids in comparisons amongst features. Fig. 3 demonstrates the box-whisker plot for outliers check, from where it can be inferred that Age, EDUC, SES, MMSE, eTIV, and nWBV feature columns exhibit outliers while other feature columns are outlier-free.
2. **Determining Skewness**: The linearity of the features was ascertained by plotting a distribution plot. The plot was used to study the skewness of both the dependent and independent variables. From the Fig. 4, it can be concluded that Group, M/F, Age feature columns appear to be normally distributed while EDUC, SES, MMSE, CDR, eTIV, nWBV, and ASF all independent variables experience skewness.
3. **Discovering Structural Correlation**: It is determined by scatter-plot matrix which is used to plot the multiple pairwise bivariate distributions of all the features in a dataset. It constructs on two figures-the scatter plot and the distribution plot (via histogram). The scatter plot on the lower and upper triangles displays the correlation among two variables, as dots in two dimensions while the histogram on the diagonal shows the spread of a particular variable. Thus, Fig. 5 illustrates the pair plot of the entire dataset which gives a valuable insight suggesting that most of the features are normally distributed except MMSE, which is heavily left-skewed.

Additionally, other EDA techniques were applied for studying the effect of independent variables on dependent variable i.e. patient classification group as AD or Non-AD. We carried out a comprehensive exploratory data analysis in our previous study (Khan and Zubair, 2020). The following characteristics were extrapolated: age is between 60 and 90 years, demented patients were less educated, significant increment in the occurrence of dementia for SES as we move from highest status (1) to lowest status (5), non-demented group got much higher MMSE scores, more number of individuals with CDR score of 0.5 (very mild dementia), slight less number of individuals with score of 1 (mild dementia), and very few with 0 score (no dementia), eTIV was found to be higher for demented patients, non-demented group has higher nWBV ratio, and demented patients have higher ASF than non-demented ones. The disparities in nWBV among CDR = 0 (non-demented), CDR = 0.5 (very mild dementia), CDR = 1 (mild dementia) occurs to be significant i.e. p less than 0.01.

4.1.3. Step 3: Feature selection

In this step, the machine automatically selects those features that help in predicting the variable or output. The feature set that is used in training the ML models greatly affects the accuracy thus influencing the performance of the model. In general, features are categorized as relevant (effect the output), irrelevant (no effect on the output), and redundant (when a feature can play the task of other feature). The chief objective of feature selection is to drop the irrelevant and redundant features, thus reducing the data dimensionality and allowing ML algorithms to execute efficiently (Kotsiantis et al., 2006). Performing feature selection before modelling, aids in reducing overfitting, reduces train time, and eventually improves accuracy.

In the present dataset, we found certain relevant and certain irrelevant features. No redundant feature set existed in the same. To discover the significant features, we performed the below-mentioned feature selection methods.

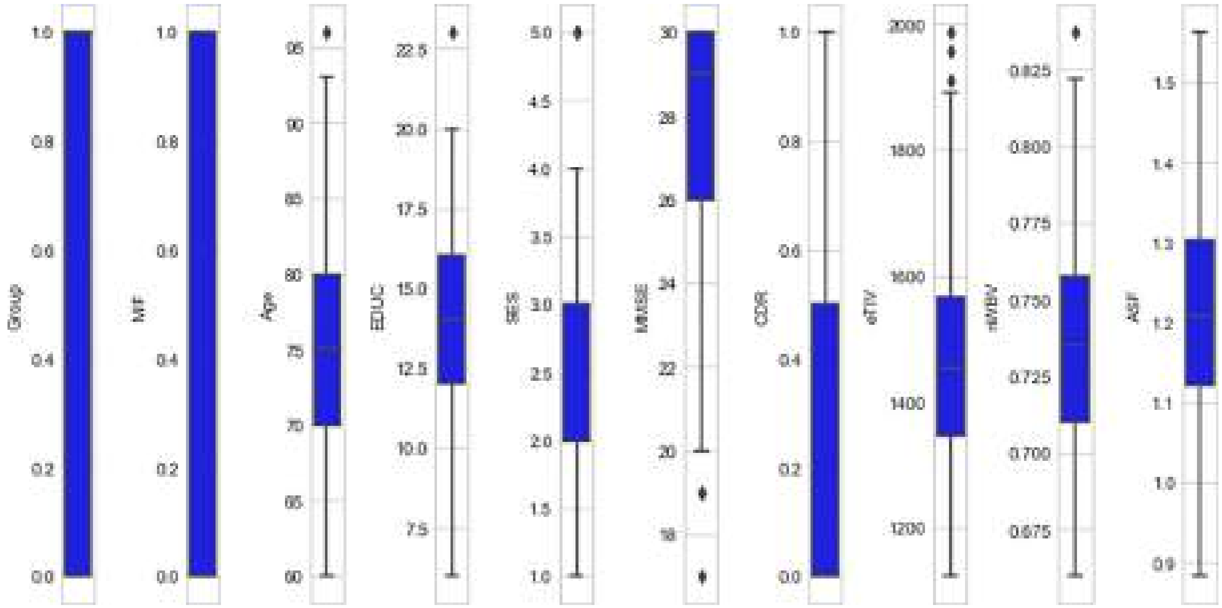


Fig. 3. Outliers detection with box-whisker plot.

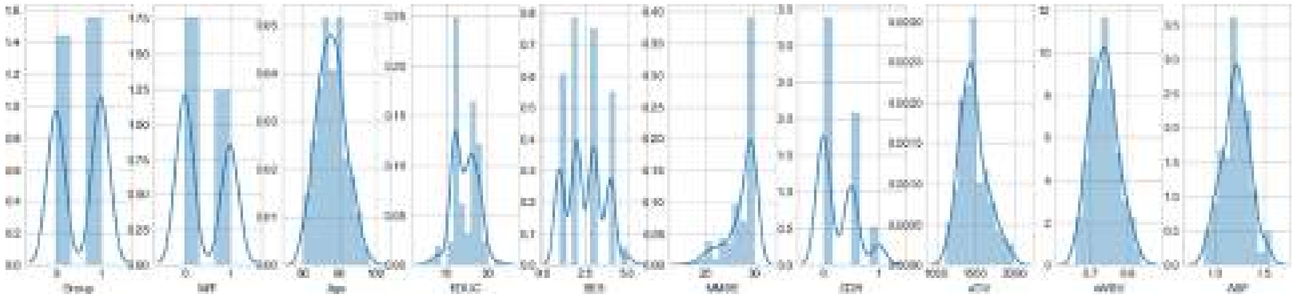


Fig. 4. Determination of skewness with distribution plot.

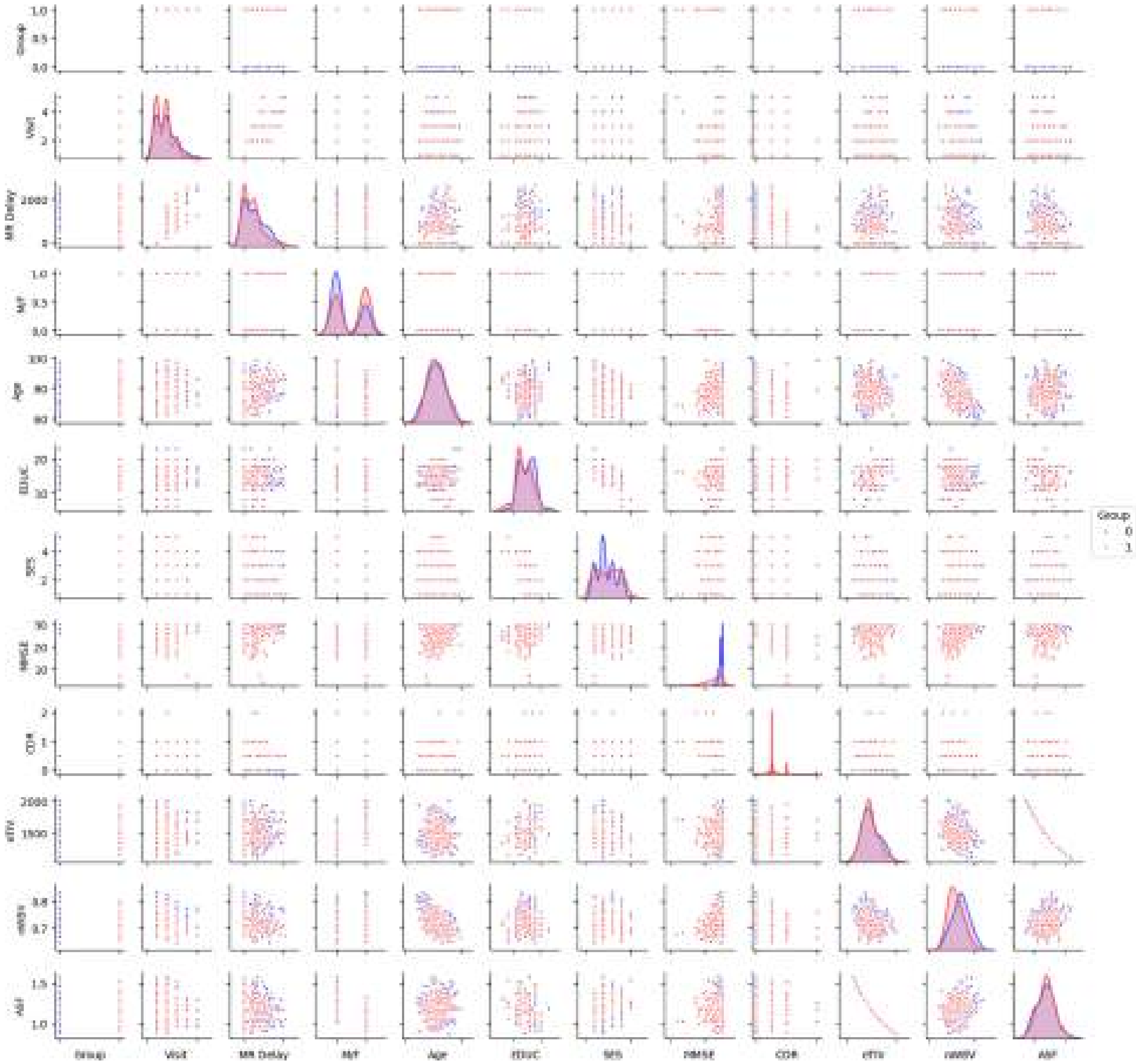


Fig. 5. Pairwise bivariate distribution.

1. **Univariate Selection**: This feature selection type applies the chi-squared statistical test to select a certain number of features from the dataset that have the strongest relationship with the dependent variable. We applied this method to the MRI dataset to select 8 of the best features. This resulted in the selection of a set of features, namely, M/F (0), Age (1), EDUC (2), SES (3), MMSE (4), eTIV (5), nWBV (6), and ASF (7), as depicted in Fig. 6. The result suggests that these features have the strongest relationship with that of demented/non-demented Group.

2. **Feature Importance**: When applied, this method results in a 'feature score' for each of the feature in the dataset. The highest score suggests the relevance and importance of that particular feature towards the dependent variable. The result of this feature selection method supports CDR with the highest score, as can be seen in Fig. 7. However, we dropped the CDR feature in our study before building of the ML model because CDR is a rating factor which may result in the less accurate models. We selected features manually prior to this automated selection method. The selected set of features resulted in the development of higher accurate models. Moreover, CDR is a dementia rating factor which is categorized into 3 ratings namely: 0, 1, and 2, which does not aid in a model building but rather it is helpful prior to the model building phase, during the division of Group of demented and non-demented patients.
3. **Correlation Matrix**: Correlation is a measure that determines the degree to which the features are associated with each other, more specifically, the target variable (Alhaj et al., 2016). It can be positive or negative. A positive correlation means that if the single value of feature increases, the target variable's value increases too, while the negative correlation indicates that if the particular value of feature increases, it results into the decrement of the target variable's value. Thus, in order to build the ML model, we determined the correlation matrix using the Heatmap. Heatmap is a graphical illustration of data, which aids in identifying the set of features that are highly related to the dependent variable. An essential condition for model building is to remove the correlated variables. The correlation matrix with Heatmap is demonstrated in Fig. 8.

The darker shades represent positive correlation while lighter shades signify a negative correlation. The target variable i.e. *Group* is dropped while testing for the correlated independent variables. From the figure, it can be seen that *Visit* and *MR Delay* are strongly correlated. However, we drop this feature too in our model building as it plays no role in determining the class of AD or non-AD group. Therefore, we can deduce that *eTIV* is positively correlated with *M/F* while negatively correlated with *ASF* amongst all.

After performing the univariate and multivariate analysis, we fed relevant feature set into the ML model for training are- Group, M/F, Age, EDUC, SES, MMSE, eTIV, nWBV, and ASF. We drop the rest of the irrelevant features because these might affect negatively on our model performance thereby decreasing the accuracy.

4.1.4. Step 4: Data transformation

In order to avoid usage of unclean data, which mostly occupies irrelevant features, outliers and duplicates, dataset is cleaned up. Such data must be transformed into an alternate scale to facilitate their usage. The data gathered from the previous step (Step 3) was not fit to be used by our ML algorithm, as this data was incomplete, inconsistent and contained many errors and missing values. After taking care of all the inconsistencies, duplicates, errors and missing data in our dataset we moved to data transformation step, also known as *Feature Engineering*. Following are the steps we performed in this stage:

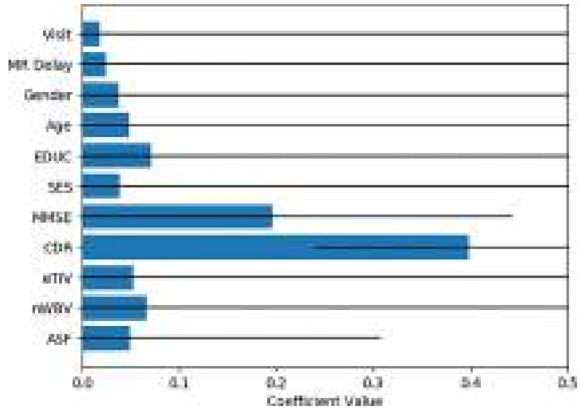


Fig. 7. Feature selection using feature importance.

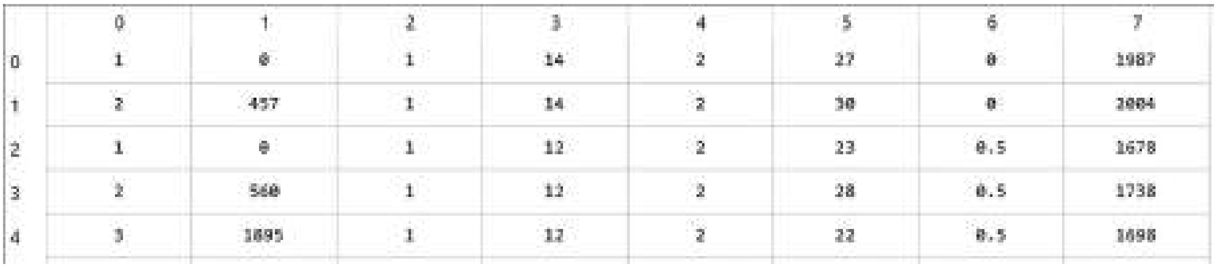
1. Handling the Missing Data: In one of our earlier study, we did a broad evaluation on the effect of impuation and non-imputation in the diagnosis of AD on the same longitudinal MRI dataset (Khan and Zubair, 2019). As stated in Section 3, only the first visit subjects have been considered throughout this study, making it to a total of 150 subjects that have been studied under this analysis, several missing values were present in the dataset. Missing values are those values in which one/more rows of certain features contain no value. In our MRI dataset, only SES feature column contained missing values, for the 150 subjects for Visit = 1 (represented by yellow lines on a purple background). Heatmap representing the count of missing values is illustrated in Fig. 9.

As there were 8 missing records in the column *SES*, these should be either deleted or imputed in the data preprocessing stage only. To overcome this issue, we applied both the approaches on the dataset- removal of missing values and imputing missing values. This helped later in building our model because there comes a huge difference in accuracies between both methods.

a) Dropping missing data: In this, rows pertaining to 8 missing values were deleted, thereby leaving the dataset for train, validation, test purposes equal to 142 subjects only.

b) Imputation of missing data: This method involves the prediction of a set of data values which are missing in the dataset. Generally, it substitutes the missing values by suitable estimates, for instance, mean or median. Next, it applies standard complete-data techniques to the filled-in data. The chief objective of imputation is to reduce the biasness owing to missing values, thus resulting in enhanced model efficiency (Pampaka et al., 2014). In this, we applied *imputation by median* method on missing data values, thereby leaving the dataset for train, validation, and test purposes for all the 150 subjects.

2. Label Encoding: Till this stage, we did the pre-processing for continuous numeric features. The dataset had categorical features



	0	1	2	3	4	5	6	7
0	1	0	1	14	2	27	0	1987
1	2	457	1	14	2	30	0	2004
2	1	0	1	12	2	23	0.5	1678
3	2	560	1	12	2	28	0.5	1738
4	3	1895	1	12	2	22	0.5	1698

Fig. 6. Univariate feature selection using chi-squared statistical test.

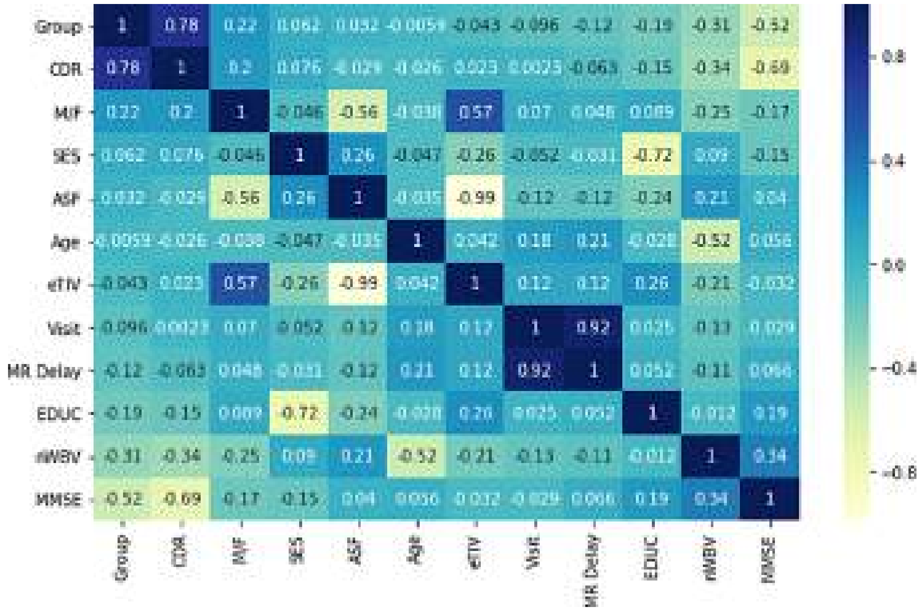


Fig. 8. Feature selection using correlation matrix with heatmap.

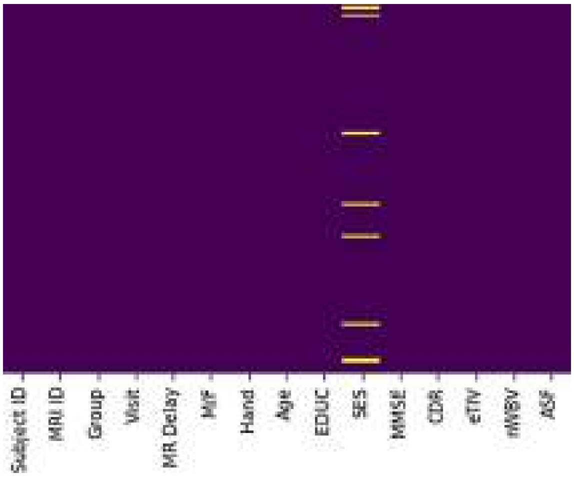


Fig. 9. Schematic representation of missing values.

too, such as patient's *Group* and *Gender* which encompass string values. Likewise, Group has two levels- demented or non-demented and Gender too has two levels viz male and female. These string values need to be converted into numeric form since the ML models are based on mathematical calculations. Therefore, we encoded both the categorical features to numeric form.

3. Feature Scaling: The final step of data preprocessing is to apply the feature scaling technique. It is the method that limits the variable range to a particular scale in order to compare them on common grounds i.e. standardize the scale of independent features. More usually, two techniques are used for feature scaling: Normalization and Standardization. Normalization is a method that scales features in between 0 and 1, holding their relative range to each other, calculated by using Eq. (1). Among the set of features, there is often a huge difference between the maximum and minimum values, for example, 0.001 and 1000. Such a range of values needs to be normalized to scale them to appreciably low values (Kotsiantis et al., 2006).

$$Normalization : x_n = \frac{x - \min(x)}{\max(x) - x} \quad (1)$$

where:

xn = new value

x = original value

Second is the standardization method, which scales to a range that results in a mean of 0 and standard deviation of 1, given by the following equation.

$$Standardization : x_s = \frac{x - m}{sd} \quad (2)$$

where:

xs = new value

x = original value

m = mean

sd = standard deviation

Thus it is necessary to convert all the features to the same scale. We applied the standardization approach for every observation of the selected column so as to fit it to a definite scale. This approach makes the model to execute much faster, therefore employed before building our ML model. Hence, the first stage of the proposed pipeline ends with data transformation approach where we obtain a clean dataset. All changes to the data from once it has been cleaned up is then ingested into the machine learning model in the further stages.

4.2. Data segregation

The clean data obtained from the data preparation step is further segregated. The primary purpose of this stage is to avoid overfitting, which focuses on secondary details and noise. They only optimize the train dataset's accuracy. Therefore, we need such a model that executes correctly on a dataset that it has never seen before i.e. test data. This is termed as a generalization. We attain this by the following approach called as *splitting data.*

Splitting Data: It is the process where available data is divided into three portions, usually for a cross-validatory purpose. One portion of the data is used to develop a predictive model while the other two are used to evaluate the model's performance. Fig. 10 describes the status of the division employed in our model.

We split our dataset into 3 sets- train set (TR), test set (TS) and validation set (VS), to train the model, test it and validate it as how it functions against new data. We trained our ML models on TR set

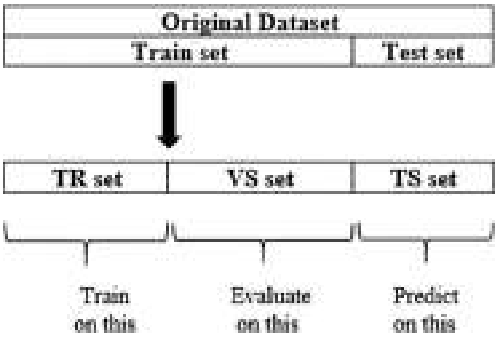


Fig. 10. Schematic representation of data segregation stage.

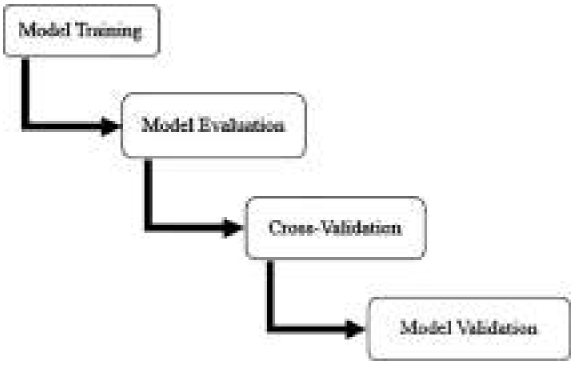


Fig. 11. Levelled pictorial representation of model building stage.

so that they look up for any correlations in the TR set, understand them and then the models were tested on TS set to check how accurately it can predict. We allocated a random selection of data for the TR set and TS set instead of 80:20 ratio as this helped the machine in making new combinations every time the model is made to run, thus making it possible to predict with higher accuracy.

Of the remaining TR data, it was again splitted into train set and validation set. The validation dataset was used to decide hyperparameters such as learning rate and regularization parameters. When the model performs efficiently on VS set, we stop learning using the train set. The problem arises when overfitting happens after many iterations for VS set, as we make use of VS set to tune parameters so as to improve its accuracy. Model's performance is measured and tested on the TS set, as it does not involve learning.

Therefore, in this data segregation stage, the overall process of data split we employed, sums up to- training model with the remaining fraction of the data, parameter tuning with the VS set and lastly, performance evaluation on the TS set.

4.3. Modeling

ML model is a mathematical description of a real-life process. Building a model relates to training a ML algorithm that can predict the labels (target variable) from the features (independent variables), tuning it, and validating it on holdout data. To generate such a model, training data is supplied to a ML algorithm from where it learns.

Proceeding our approach, the output of the second stage i.e. a clean set of split data serves as the input for this level where the actual model is built using the training set of data acquired from data segregation stage. The output of modeling stage is a trained model that can be used for interpretation, making predictions on newer data values. The objective of this phase is not to develop a model that operates correctly on train data rather its main aim is to satisfy the needs of the agenda behind creating this model and that can be deployed on real data. Following are the 4 levels illustrated linearly in Fig. 11 that fall under this third level of the proposed pipeline.

Each process is unfolded and explained in the following sections.

4.3.1. Step 1: Model training

It is a process in which a ML classifier obtains insights from the train set and learns its parameters over the training cycle that reduces the loss or how poor it executes on the train set. In this, while training to learn, an ML algorithm (classifier) is passed with train data. The classifier uncovers patterns in the train data so that the parameters inputted correspond to the target variable. The output of the model training process is a ML model which is further used to construct predictions. This practice is also named as *learning.*

In particular, our aim was to develop a system for an unambiguous task of input-output transformation where we can determine the classification of a demented and non-demented group of patients successfully with the highest accuracy. In our dataset, as we have the target variable (dependent variable) having only two sets of values: demented and non-demented, so in order to predict this set of values, given the independent set of features, we applied below described seventeen supervised ML classifiers for model training. We worked on selected models as it resulted in more accurate results. The models that showed an accuracy of less than 50%, were dropped and not included in the study.

1. **AdaBoost (Adaptive Boosting) classifier:** It focuses chiefly on binary classification problems. It is an ensemble classifier and a meta-estimator. Ensemble classifiers execute by combining several ML classifiers. AdaBoost operates on the weak ML classifiers that result in a strong classifier. It is a meta-estimator which implies that it commences by classifier fitting on the train dataset and then fitting the additional replicas of the classifier on the same train dataset. The difference lies in the fact that the assigned weights of wrongly categorized occurrences are altered in such a way that the subsequent classifiers emphasize more on challenging cases (Cao et al., 2013). This classifier is more often used to further advance the performance of any ML algorithm. Moreover, they attain higher accuracy as compared to random likelihood on a given classification problem.

2. **Extra Trees (Extremely Randomized Trees) classifier:** This classifier is an enormously randomized tree classifier which is used within ensemble methods. It differs from decision tree classifier in the way they are constructed. Moreover, they are much faster than random forest classifiers. It operates by aggregating the results of manifold dissociated decision trees which are collected in a forest-form structure so as to output its classification outcome. It employs a meta-estimator that operates by fitting several randomized decision trees viz. extra-tress on different dataset features (Geurts et al., 2006). It controls over-fitting and utilize the averaging method so as to improve the classifier's accuracy. Also, each of the decision tree in the forest of extra trees is built from the primary training set.

3. **Gradient Boosting classifier:** Gradient boosting is a ML technique that generates such a prediction model which is an ensemble of weak prediction models (usually decision trees) (Natekin and Knoll, 2013). It combines several weak ML classifiers together to construct a single strong classifier. It constructs a model in a forward stage-wise approach. Moreover, it provides optimization of randomized differentiable loss functions. This classifier is built on the principles of AdaBoost classifier and is ahead of AdaBoosting method. In this, the AdaBoost approach is fused with the weighted minimization, and then both the classifiers and weighted inputs

are recomputed. The primary aim of Gradient Boosting classifier is to minimalize the resultant loss i.e. the difference between the real class value of the train set and the predicted class value of the test set.

4. **Random Forest classifier:** It is a meta-estimator ML classifier, which operates as an ensemble technique. To be precise, it includes numerous distinct decision trees that function as an ensemble. It fits several decision tree classifiers on different subsamples of the original dataset (Denisko and Hoffman, 2018). In the random forest, each and every individual distinct tree gives the class prediction. The class which consists of the major votes turns out to be the final resultant model prediction. Moreover, it uses an averaging technique to enhance accuracy and also, it regulates over-fitting. It consists of a significant number of moderately uncorrelated trees that operate as one group. This ensemble group then perform better than any of the specific constituent classifiers. In this, the low correlation among the models is the strategic idea. The higher the number of uncorrelated trees which operate as a group, the better the results; when compared to the single constituent models.

5. **Gaussian Process classifier:** Gaussian process classification (GPC) is centred on the approach of Laplace approximation. The Laplace approximation is applied to approximate the non-Gaussian after Gaussian. A Gaussian process is a stochastic method i.e. it is a group of random variables which are indexed by time/space. Each and every determinate group of random variables consists of multivariate normal distribution. ML algorithm that includes a Gaussian process employs a lazy learning method and a similarity measure that predicts the target feature value from train data (Csató et al., 2000). Furthermore, it is a non-parametric method which is based on a Bayesian approach. In this, it presumes certain prior distribution based on the fundamental probability densities which ensure in improved efficiency. The resultant classification is then ascertained by the one that gives a good fit for the trained data.

6. **Logistic RegressionCV classifier:** Logistic Regression is one of the elementary ML classifier employed for classification problems. It uses the logistic function, which is built on the framework of the sigmoid function. A sigmoid function takes any real value in between 0 and 1. In this, the dependent variable operates on the outline of Bernoulli distribution; where the approximation is achieved through the maximum probability. Thus, Logistic RegressionCV classifier implements a regularized logistic regression classification algorithm. It has an in-built cross-validation feature. Moreover, it performs optimization by using the liblinear library. The liblinear library has an advantage over others as it supports both L1 and L2 regularization (Pedregosa and Varoquaux, 2011).

7. **Passive Aggressive classifier:** It is a group of online learning algorithms, meant for both ML classification and regression methods. In this, different algorithms each of binary and multiclass classification, regression, sequence prediction and uniclass prediction are analyzed in particular (Crammer and Dekel, 2006). This unified analysis permits to look for the worst-case loss constraints for these set of diverse algorithms. Herein, the classification is based on the approach where learning from data does not fit in the main memory. This online classifier is used employing partial-fit method, where the model is trained in batches. A HashingVectorizer is applied which ensures that the feature space continues to be same over the time. This vectorizer projects every data sample into the uniform feature space.

8. **Ridge ClassifierCV classifier:** Ridge classifier consists of a built-in cross-validation facility. It implements a generalized cross-validation method by default. This method is an approach of cross-validation where cross-validation is applied in a leave-one-out manner (Pedregosa and Varoquaux, 2011). The approach is different from that of Logistic Regression classifier. The difference is based on the L2 regularization employed in their functioning. Initially, a target variable is generated with +1 and −1 values, centred on the class to which it belongs. Next, a Ridge model is constructed to predict the target set of data. Here, the loss function is equal to the root mean square and L2 penalty. If the predicted value results into a value greater than 0, then the prediction performed by the model is categorized into a positive class otherwise negative class.

9. **Stochastic Gradient Descent (SGD) classifier:** It is an efficient ML classifier for discriminative-based learning of linear classifiers such as Logistic Regression and Support Vector Machines. It implements regularized linear classifiers along with SGD learning. The model it fits is usually regulated with the loss parameter. The gradient of the loss function is approximated by taking each one of the samples at a time, thereby updating the model simultaneously (Robbins and Monro, 1951). SGD classifier works best with the data categorized as a sparse or dense matrix of floating-point values (Ruder, 2016). This classifier is efficient and can be implemented easily as compared to several other supervised ML classifiers. However, it entails numerous hyperparameters. For instance several iterations and regularization parameters. Also, it is hypersensitive to feature scaling, which is one of the major drawbacks of SGD classifier.

10. **Perceptron classifier:** It is a generalized computational model which is used to employ linearly separable functions. It is a binary ML classifier which is based on the similar underlying concept as of SGD classifier. A binary classifier is a classifier that determines the specific class of given input that it belongs to (denoted by a vector of numbers). Generally, it aggregates the given input i.e. weighted sum and gives an output of 1, if the weighted sum comes to be greater than a threshold value; else returns a value equal to 0 (Gardner and Dorling, 1998). It employs threshold function, which operates by mapping its input value, a real-value vector to an output value, a single binary-value. Based on the output as 0 or 1, Perceptron classifier performs the task of positive or negative classification.

11. **Naïve Bayes (NB) classifier:** It is a statistical ML classification algorithm, grounded on Bayes' theorem. It operates by presuming that in a given class, the effect of a specific feature remains independent of other feature-set. This notion is called as class-conditional independence. Following two categories of NB classifiers are defined, based on the assumptions they formulate pertaining to the distribution.

(a) **BernoulliNB classifier:** It is a type of Naive Bayes classifier meant for multivariate modeling. It works with features that are boolean (binary) in nature. It implements NB classifier on the basis of the multivariate Bernoulli distributions. In other words, BernoulliNB assumes each feature as binary valued despite of the fact that the training set of data may contain multiple features.

(b) **GaussianNB classifier:** It is a type of Naive Bayes classifier, used when the features have continuous values. It presumes that all the feature-set follow a Gaussian distribution viz. normal distribution.

13. **KNeighbors classifier:** This classification type is a kind of instance-based learning. It gathers the instances of the train data instead of building a generalized internal model. KNeighbors classifier looks at only those observations that are in the close proximity of those instances that the classifier attempts to predict (Zhang, 2016). It operates learning centered on the K nearest neighbors, where K is highly data-dependent and denotes a numeric value which is specified by the user. Moreover, it is a non-parametric ML classifier. The term 'non-parametric' denotes that no presumptions are crafted for the underlying distribution of the data. It per-

forms effectively with lesser number of feature-set as compared to a large number of features.

14. **Decision Tree (DT) classifier:** It is a tree-like structure which consists of a root node, internal node, leaf node and branch. The top node is the root node, the internal node denotes the features (attributes), the leaf node represents the result and the branch corresponds to the decision rule. In addition, it learns the pattern and thereby to partition based on the feature values. DT is a supervised MLclassifier which is non-parametric in behaviour. The chief objective of the decision tree is to build such a model that can predict the target feature by learning the set of certain decision rules which are deduced from the data features (Amancio and Comin, 2014). It learns to divide on the basis of the feature value. The partitioning of the tree is performed in a recursive manner which further helps in decision making. These operate well on high dimensional data with better accuracy.

15. **Support Vector Machines (SVMs):** These classifiers are a set of supervised learning ML methods used for both classification and regression. They perform efficiently in high-dimensional datasets. A subset of train data is used in the support vectors viz. decision function, thus making it more memory proficient (Kotsiantis, 2007). It is versatile in behaviour as it uses various kernel functions for the decision function. According to the kernel parameter, the following three categories of these set of classifiers are defined:

(a) **Support Vector Classification (SVC):** This classifier is implemented using the 'libsvm' library. Libsvm is an integrated software designed for the SVC. SVC employs RBF (Radial Basis Function) kernel as a default. When the RBF kernel is used, SVC considers 2 parameters, namely, C and gamma. A 'C' parameter substitutes the misclassification of train data with that of decision surface. A higher value of C results in the classification of train data correctly. While the 'gamma' parameter denotes the effect of a single train data.

(b) **NuSVC:** This is similar to SVC as it is implemented using the libsvm library. But the only difference lies in the regularization parameter, NuSVC employs a parameter 'nu' which controls the number of decision functions while SVC uses a 'C' parameter (Pedregosa and Varoquaux, 2011).

(c) **LinearSVC:** This ML classifier is similar to SVC (Support Vector Classifier), the only difference lies in the kernel parameter, which is set as 'linear' in linear SVC. It is implemented in liblinear form instead of libsvm (Pedregosa and Varoquaux, 2011). Thus, it has more flexibility in selecting a suitable penalty and loss function parameter from the range of penalties and loss functions. Linear SVC fits the train data and gives a best-fit hyperplane as an output which divides and classifies the train set of data. From the attained hyperplane, it can then predict and categorize the right class of data. It supports both sparse and dense data as input.

4.3.2. Step 2: Model evaluation

After applying the above classification models to the train set, we obtained a diverse set of accuracies as can be seen in Fig. 12 (a) and (b). We applied for both the non-imputation and imputation of missing values.

So finally we have built our classification model and we can see that SGD classifier for non-imputation and Logistic RegressionCV classifier for imputation performs well on train data with the highest accuracy amongst all, thus producing the best results for our dataset. This is not always applicable and valid for every dataset. To choose a respective model, we always need to analyze and evaluate the dataset first and then apply the specific ML model.

4.3.3. Step 3: Cross-validation

It is a practice of finding a good model by avoiding training and testing on the same data, as the main objective of the model is to predict test data (out-of-sample data) where the model could become highly complex resulting into over-fitting. To avoid the aforementioned problem, we performed K-Fold Cross-Validation.

K-Fold is a technique that separates the dataset into *K* number of divisions, each being equally sized. For each ith division, the model is trained with the leftover *K-1* divisions and the produced model is then assessed on ith division (Arlot and Celisse, 2010). The result is the average of the obtained K scores. At last, the performance is measured and evaluated by averaging across all K-folds so as to estimate the ability of the learning classifier on the problem. This method is specifically beneficial when the performance of the model is considerably dissimilar from the train-test split.

To this, we loaded the dataset to cross-validation score function and it was the ML classifiers that yielded the best score. We chose a value of *K = 10* and applied 10-fold cross-validation that involved training and testing a model 10 times. The primary purpose was to maintain a balance among the magnitude and representation of data in the train and test sets thereby improving the accuracy of each classifier applied. Next step is to optimize the results by tuning its hyperparameters accordingly, explained in the following section.

4.3.4. Step 4: Model validation

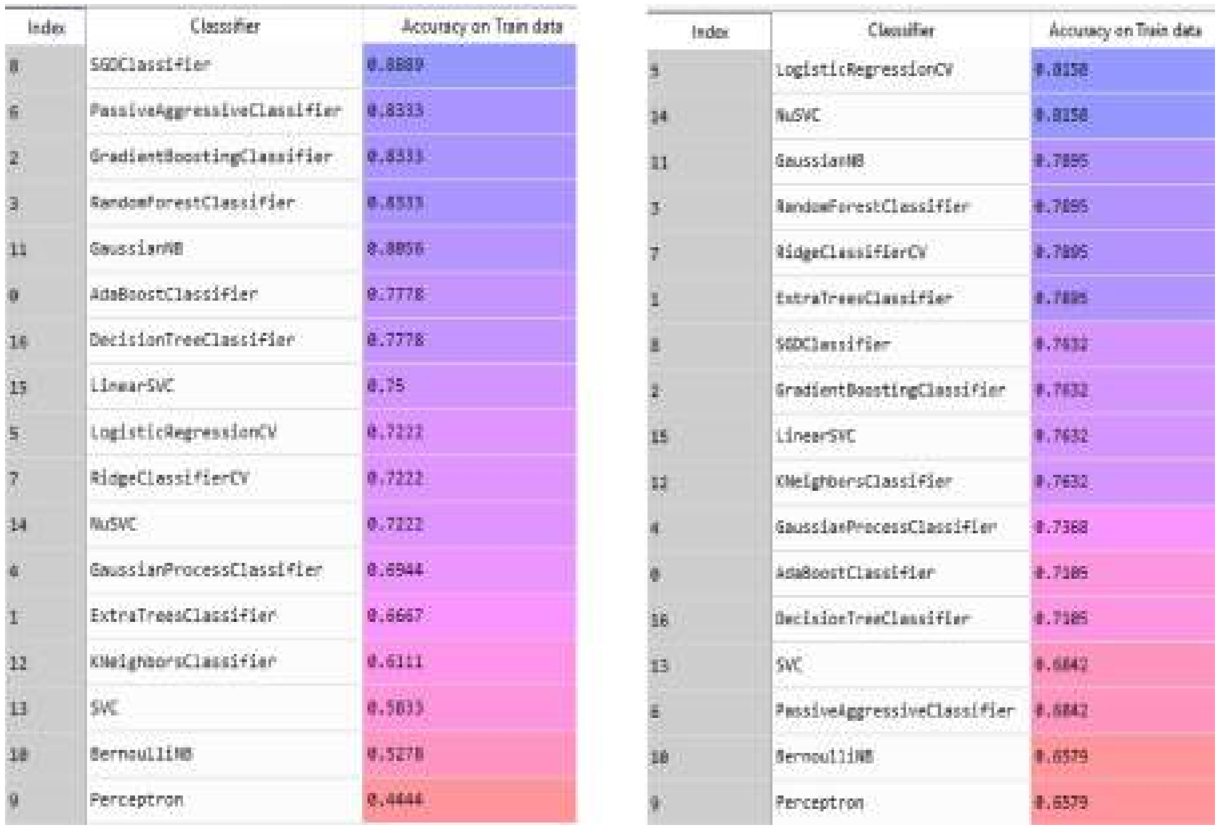
Once the cross-validation is achieved on the train set, a model validation step is performed on the validation set of train data. It is also known as Spot checking. It allows a quick way to perceive if any learnable structures exist in the data extending to estimate which classifiers may sound effectual on the problem. Also, it makes sure that the selected performance measure is appropriate or not.

ML algorithm has two types of parameters: parameters and hyperparameters. The first type are the parameters that are learned through the training phase and the second type are the hyperparameters that we pass to the ML model i.e. the validation phase. These learning classifiers optimize the parameters by various means while hyperparameters cannot be estimated from the train data. Hyperparameter tuning is applied to the validation set (VS) of train data. Fig. 13 shows the basic idea behind this step.

In the model validation step, we validated our ML model against the validation set. The output of the previous step (step 3) when applied on the dataset, generated a good model. After performing cross-validation, the next step was to tune its hyperparameters to obtain the models with that of best possible predictions. We applied the same and found a suitable combination of hyperparameters by Grid Search Cross-Validation technique. This method gives a set of hyperparameters that fits best in determining a set of group of healthy and inflicted patients. Overall, it gave an unbiased evaluation of a ML model fitted on the train set. Following are the transformed accuracies of various classifiers applied after applying cross-validation and tuning the hyperparameters (for VS set only) (Fig. 14).

Fig. 14(a) suggests that LinearSVC outperforms all other classifiers on VS dataset for non-imputation of missing values while GaussianNB has the highest CV accuracy for imputation of missing values on VS dataset. Looking at these results of non-imputation of missing values i.e. dropping them, we chose to mention only the results of the complete pipelining method for imputation of missing values (by median).

Among all this, the best-performing model from a set of models produced by different hyperparameter settings, metrics, and cross-validation techniques is chosen after testing and validating this



Index	Classifier	Accuracy on Train data
8	SGDClassifier	0.8889
6	PassiveAggressiveClassifier	0.8333
2	GradientBoostingClassifier	0.8333
3	RandomForestClassifier	0.8333
11	GaussianNB	0.8056
0	AdaBoostClassifier	0.7778
16	DecisionTreeClassifier	0.7778
15	LinearSVC	0.75
5	LogisticRegressionCV	0.7222
7	RidgeClassifierCV	0.7222
14	NuSVC	0.7222
4	GaussianProcessClassifier	0.6944
1	ExtraTreesClassifier	0.6667
12	KNeighborsClassifier	0.6111
13	SVC	0.5833
10	BernoulliNB	0.5278
9	Perceptron	0.4444

(a) By non-Imputation of missing values.

Index	Classifier	Accuracy on Train data
5	LogisticRegressionCV	0.8158
14	NuSVC	0.8158
11	GaussianNB	0.7895
3	RandomForestClassifier	0.7895
7	RidgeClassifierCV	0.7895
1	ExtraTreesClassifier	0.7895
8	SGDClassifier	0.7632
2	GradientBoostingClassifier	0.7632
15	LinearSVC	0.7632
12	KNeighborsClassifier	0.7632
4	GaussianProcessClassifier	0.7368
0	AdaBoostClassifier	0.7105
16	DecisionTreeClassifier	0.7105
13	SVC	0.6842
6	PassiveAggressiveClassifier	0.6842
10	BernoulliNB	0.6579
9	Perceptron	0.6579

(b) By Imputation of missing values.

Fig. 12. Accuracy on train data.

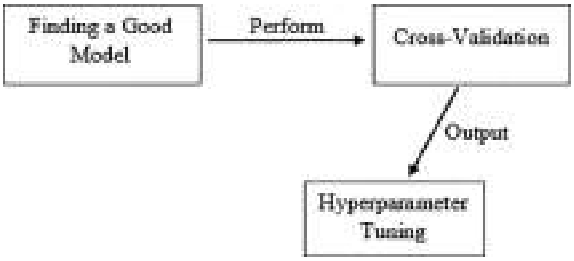


Fig. 13. An illustration for model validation stage.

produced model which is done in the next stage i.e. Model Prediction.

4.4. Model prediction

This is a stage where the performance of the model is assessed using test dataset in order to understand the prediction's accuracy. It is an iterative process where various ML classifiers are tested until we find a model that answers our question adequately.

Given the results of hyperparameter tuning (the last stage of Modeling) after performing cross-validation, we are now in the stage of assessing our model on the test set. Here, we evaluated our ML models by determining whether they are able to predict the target variable on new data or not. Once training the ML model was done successfully, the model was then directed to the held-out data for which the target values were known. Thereafter, the predictions returned by the ML classifiers were compared against the known target scores. Finally, the performance metric was calculated which reveals how well the predicted and true values match.

After the training of the model, it was next evaluated by making use of certain evaluation metrics on the test set. Hence, the model function on the test set exploiting only the information learned from the train set. We applied classification accuracy i.e. testing accuracy (Eq. (3)) for evaluating the set of learning models to determine which model performs better with the MRI dataset. We then predicted the test set results and checked the accuracy with each of our models.

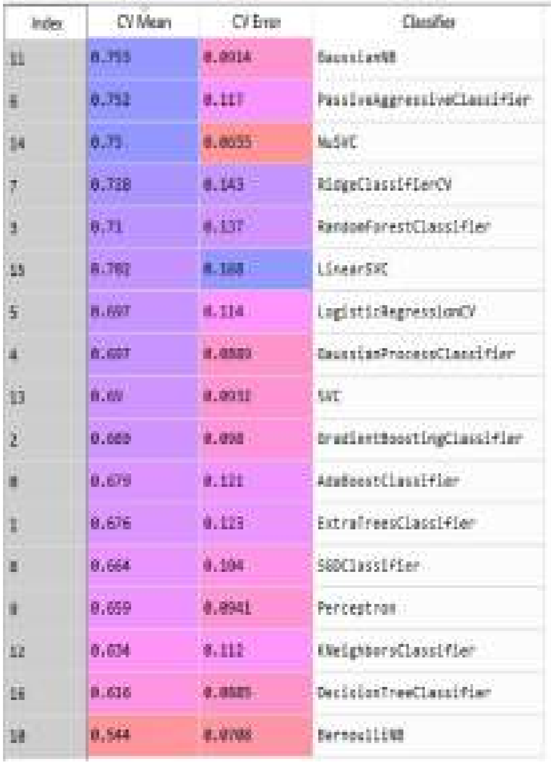
$$Accuracy = \frac{Numberofcorrectpredictions}{Totalnumberofpredictionsmade} \tag{3}$$

The results of model prediction are shown in Fig. 15.

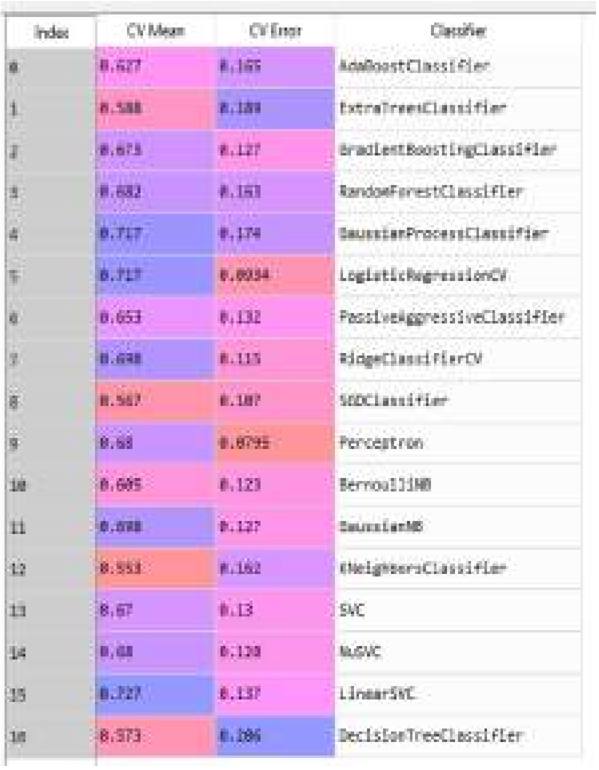
It can be deduced from the above figure that Random Forest (RF) classifier outperforms all other classifiers with the highest accuracy of 86.84 percent on the test dataset. This is the accuracy with which it predicts the group of demented and non-demented patients.

As we started the model training with 17 ML classifiers, it can be easily seen from Fig. 16, that accuracy result of only 10 ML classifiers is specified. The rest 7 of the classifiers don't require tuning. Hence, they are not displayed in the comparison results. Moreover, they resulted in lesser accuracies, as presented in Table 6 (obtained from Figs. 12(b) and 14(b)).

With our ML pipeline completed with the modeling and testing phase, we now evaluate the overall performance of the ML model in diagnosing the AD group in the subsequent stage.



Index	CV Mean	CV Error	Classifier
11	0.755	0.0914	GaussianNB
6	0.752	0.117	PassiveAggressiveClassifier
14	0.75	0.0655	NuSVC
7	0.728	0.143	RidgeClassifierCV
3	0.71	0.137	RandomForestClassifier
15	0.702	0.168	LinearSVC
5	0.697	0.114	LogisticRegressionCV
4	0.697	0.0889	GaussianProcessClassifier
13	0.69	0.0931	SVC
2	0.689	0.098	GradientBoostingClassifier
0	0.679	0.121	AdaBoostClassifier
1	0.676	0.123	ExtraTreesClassifier
8	0.664	0.104	SGDClassifier
9	0.659	0.0941	Perceptron
12	0.634	0.112	KNeighborsClassifier
16	0.616	0.0885	DecisionTreeClassifier
10	0.544	0.0708	BernoulliNB



Index	CV Mean	CV Error	Classifier
0	0.627	0.165	AdaBoostClassifier
1	0.588	0.189	ExtraTreesClassifier
2	0.673	0.127	GradientBoostingClassifier
3	0.682	0.163	RandomForestClassifier
4	0.717	0.174	GaussianProcessClassifier
5	0.717	0.0934	LogisticRegressionCV
6	0.653	0.132	PassiveAggressiveClassifier
7	0.698	0.115	RidgeClassifierCV
8	0.567	0.107	SGDClassifier
9	0.68	0.0795	Perceptron
10	0.605	0.123	BernoulliNB
11	0.698	0.127	GaussianNB
12	0.553	0.162	KNeighborsClassifier
13	0.67	0.13	SVC
14	0.68	0.128	NuSVC
15	0.727	0.137	LinearSVC
16	0.573	0.206	DecisionTreeClassifier

(a) By non-Imputation of missing values. (b) By imputation of missing values.

Fig. 14. Accuracy on validation set of train data.

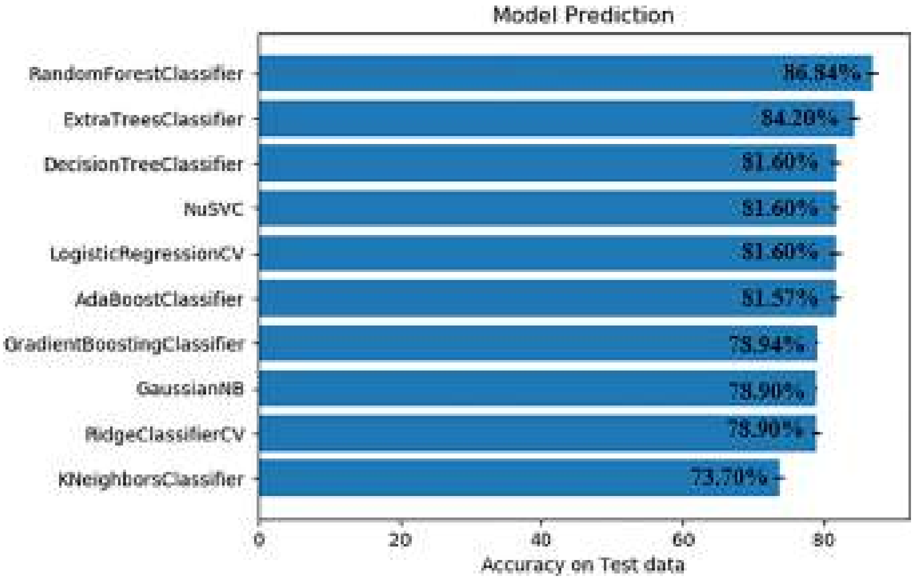
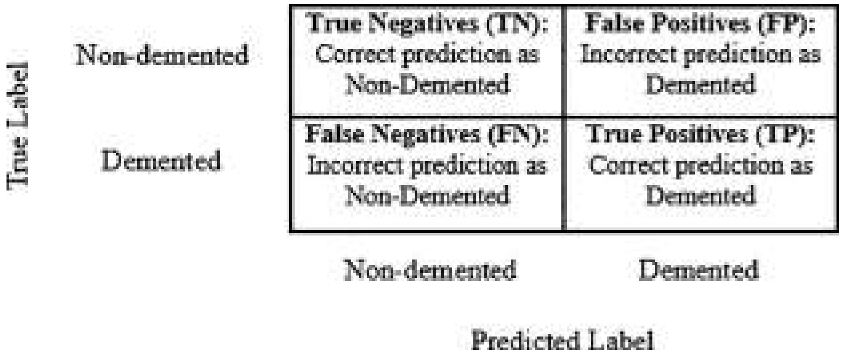


Fig. 15. Comparison results (by imputation of missing values).

4.5. Performance metrics

A performance evaluation method is an approach towards assessment of a ML model. It is the measurement that is executed for the predictions made by a trained model on the test set.

In this final stage of the pipeline, a detailed breakdown of performance for the diagnosis of AD patients (whether demented or not) is achieved by applying certain performance metrics on the results that were attained from the previous step. There exist many standard performance evaluation techniques to choose from. Following are the metrics we employed in determining the class of target variable while stating the best-achieved model for the diagnosis of AD among all applied methods so far in our pipeline on the MRI dataset: Classification Accuracy, Recall, Precision, ROC, and AUC.



True Label		Predicted Label: Non-demented	Predicted Label: Demented
	Non-demented	True Negatives (TN): Correct prediction as Non-Demented	False Positives (FP): Incorrect prediction as Demented
	Demented	False Negatives (FN): Incorrect prediction as Non-Demented	True Positives (TP): Correct prediction as Demented

Fig. 16. Confusion matrix distribution describing the performance of a classification model.

Table 6
Accuracy result of 7 ML classifiers (by imputation).

Classifier	Accuracy on train data	CV mean	CV error
LinearSVC	0.7632	0.702121	0.168373
GaussianProcess	0.7368	0.696818	0.0888593
PassiveAggressive	0.7368	0.680303	0.110296
SVC	0.6842	0.690303	0.931743
BernoulliNB	0.6579	0.544091	0.0707641
Perceptron	0.6579	0.658636	0.094108
SGD	0.5526	0.65197	0.0798215

To check for the correct predictions, confusion matrix (CM) is used normally. The CM is a way of presenting the number of misclassifications in a tabular form, as illustrated in Fig. 16. The misclassification implies for the predicted classes that end up in an incorrect classification bin based on the true classes.

Moreover, the results with respect to this classifier system are characterized with a confusion matrix, as true positive (TP), false positive (FP), true negative (TN) and false negative (FN) (Tharwat, 2018). Following are the metrics that are computed from the CM:

1. **Classification Accuracy:** It is the percentage of correct predictions.

$$ClassificationAccuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{4}$$

2. **Classification Error:** It is also known as *Misclassification rate*, which tells the classifier's incorrect predictions.

$$ClassificationError = \frac{(FP + FN)}{(TP + FP + FN + TN)} \tag{5}$$

3. **Precision:** It measures how often the prediction is correct when predicting positive instances.

$$Precision = \frac{TP}{(TP + FP)} \tag{6}$$

4. **Recall:** Also known as *Sensitivity* or *True Positive Rate*. It measures how often the prediction is correct when the actual value is positive.

$$Recall = \frac{TP}{(TP + FN)} \tag{7}$$

5. **Specificity:** It measures how often the prediction is correct when the actual value is negative.

$$Specificity = \frac{TN}{(TN + FP)} \tag{8}$$

6. **F1 Score:** It measures the accuracy of test set and is used to rate the model performance. It is the harmonic mean between precision and recall, maintaining a balance between both. The range lies between [0,1]. Robinson et al., 2015.

$$F1 = 2 * \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{9}$$

F1 score ascertains the correctness of the classifier i.e. how many samples it categorizes correctly. A higher value of F1 score implies that the model performs better.

7. **ROC Curve:** It a probability curve between the true positive rate (sensitivity) and false positive rate (1- specificity). It facilitates in choosing a threshold level that maintains a balance amongst sensitivity and specificity (Fawcett, 2006).

8. **AUC Score:** It signifies the degree of separability. Its value is computed from the area underneath the ROC plot. A higher value of AUC suggests that the classification model behaves better in predicting the target variable (Fawcett, 2006).

Using the equations above, evaluation metrics for the tested set of classifiers are tabulated in Table 7.

Given the results of Table 7, the highest accuracy, recall, precision, and AUC belongs to Random Forest classifier with 86.84%, 80.0%, 94.11%, and 87.22% respectively, followed by Extra Trees and Decision Tree classifiers. In addition, KNeighbors classifier has the lowest classification accuracy in the diagnosis of AD. Moreover, the accuracy of Decision Tree, NuSVC, Logistic RegressionCV, and AdaBoost was almost equal, meaning all these algorithms resulted in the same classification accuracy of 81.60%. Similarly, Gradient Boosting, GaussianNB and RidgeClassifierCV gave an accuracy of approximately 78.94%.

5. Experimental results and analysis

As we can see from Table 6 (Section 4.5), among all the applied algorithms, Random Forest classifier significantly outperform all other classifiers. The most significant improvements were reached by using the imputation of missing data by the median method. It resulted in a classification accuracy of 86.84% respectively, versus 73.70% of KNearestNeighbors with the least accuracy among all tested classifiers. However, this holds for classification accuracy only. Our goal was to increase the specificity and maintain a good balance between specificity and sensitivity. High specificity means that there are fewer number of false positive instances. As the ML classifiers are built to maximize the classification accuracy, in general, we cannot influence their sensitivity and specificity directly.

Our aim was to predict the results of dementia from the longitudinal MRI data with ML methods. For this, Random Forest classifier is selected, as we can see from the *Model Prediction* phase that this classifier resulted into the highest accuracy and proves to be the most suitable classifier for prediction of dementia as we will uncover all the metrics for this classifier in the following section.

In this section, all the metric results pertaining to the Random Forest (RF) classifier are presented to support our method.

Table 7
Performance result of binary classification.

ML classifier	Accuracy	Recall	Precision	AUC
Random Forest	0.8684	0.8000	0.9411	0.8722
Extra Trees	0.8420	0.7500	0.9375	0.8472
Decision Tree	0.8160	0.6500	1.0000	0.8250
NuSVC	0.8160	0.6500	1.0000	0.8250
LogisticRegressionCV	0.8160	0.7500	0.8800	0.8194
AdaBoost	0.8157	0.6500	1.0000	0.8250
Gradient Boosting	0.7894	0.7000	0.8750	0.7944
GaussianNB	0.7890	0.6500	0.9200	0.7972
RidgeClassifierCV	0.7890	0.7000	0.8750	0.7944
KNeighbors	0.7370	0.6500	0.8125	0.7417

5.1. Confusion matrix for RF classifier

The confusion matrix, both with and without normalization are illustrated in Fig. 17(a) and (b).

The following can be comprehended from above set of figure:

- Every observation in the Test (TS) set corresponds to one single box.
- **True Positives (TP):** RF correctly predicted the demented rate, as 16 patients. 80% of the MRI cases have been detected to suffer from this disease.
- **True Negatives (TN):** RF correctly predicted that 17 patients do not have dementia. 94% of the MRI cases have been detected that does not suffer from dementia.
- **False Positives (FP):** RF incorrectly predicted that only 1 patient (only 6%) have dementia i.e. it falsely predicts positive. This is called "Type I error".
- **False Negatives (FN):** RF incorrectly predicted that only 4 patients (only 2%) don't have dementia i.e. it falsely predicts negative. This is called "Type II error".

CM allows computing a range of metrics. In the subsequent sections, performance metrics are presented that are calculated from the confusion matrix.

5.2. Performance of RF classifier

The overall performance for the proposed method in the classification of AD/non-AD is reported in Fig. 18.

Classification accuracy of RF classifier is almost 87.0%. It is the number of correct predictions made from the total predictions. For robustness of our model, we predicted on test data i.e. unseen data employing techniques like cross-validation and hyperparameter tuning. From the above results and confusion matrix distribution's result, we can infer that our proposed method is good enough to solve the state of AD problem. However, at the same time, classification accuracy is not usually enough to make this decision. High accuracy does not indicate that it is a good classifier because there exist certain errors like the classification error which resulted in 13.0% error rate in our case.

From Fig. 18, it is easily observed that precision or positive predictive score comes out to be almost equal to 94.0%. High precision signifies that the RF classifier returned considerably more relevant outcome than irrelevant ones. We could also see the value of recall (sensitivity or true positive rate) as exactly equal to 80.0%. The higher the value, the better it is. A high value of recall suggests that the RF classifier returned maximum relevant results, but at the same time, it returned certain false instances also. Furthermore, it can be observed that the F1 score is equal to 87.0% approximately. This implies that RF classifier maintains a fair balance between precision and recall. The AUC score comes out to be equal to almost 87.0%. AUC is beneficial as a single number summary for the representation of a classifier's performance. A high value of AUC implies that our model is better in detecting the group of inflicted and healthy patients. The more detailed results pertaining to the class of non-demented (labeled as 0) and demented (labeled as 1) are shown in Fig. 19.

The above figure demonstrates the classification report for RF classifier. This presents the precision, recall, F1 score and support values of each group of the target variable (0 and 1) for the proposed model. This provides a deep insight towards the behavior of RF classifier over the total accuracy. The support value suggests

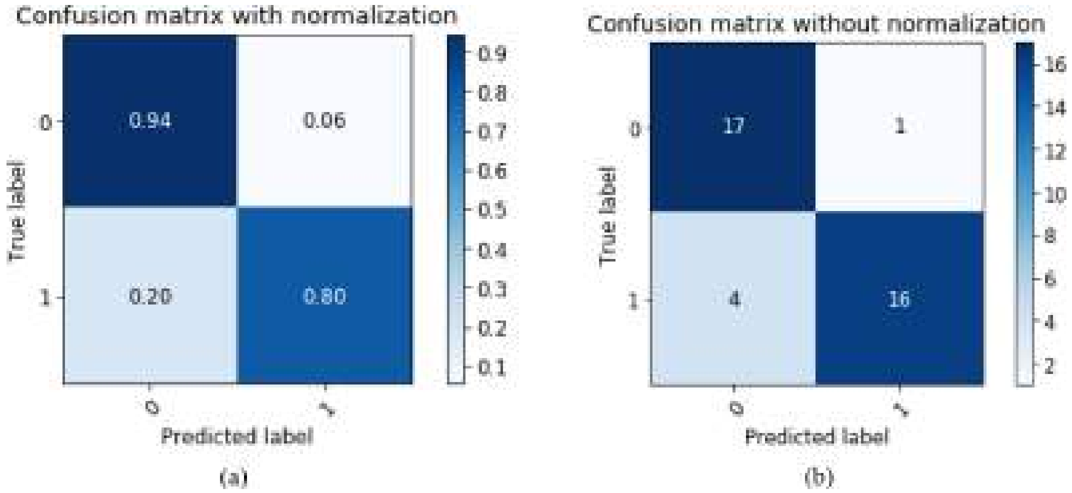
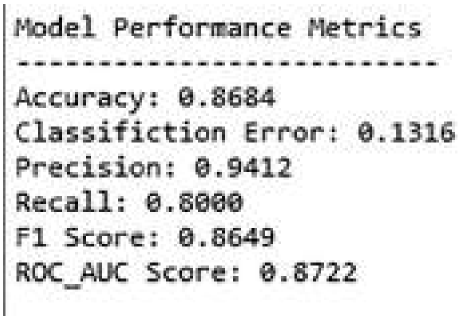


Fig. 17. Classification comparison based on confusion matrix.



```
Model Performance Metrics
-------------------------
Accuracy: 0.8684
Classifiction Error: 0.1316
Precision: 0.9412
Recall: 0.8000
F1 Score: 0.8649
ROC_AUC Score: 0.8722
```

Fig. 18. Performance result for RF classifier.

the total number of actual instances of the class in the test set. It aids in diagnosing the overall evaluation process.

The sensitivity and specificity of RF classifier were determined to be 80% and 88% respectively, as can be seen from Table 8. The sensitivity of AD test detects 80% of the patients with dementia i.e. true positives while 20% of the cases remain undetected i.e. false negatives. Additionally, AD test with 88% specificity means that the RF classifier correctly detects 88% of the patients as non-demented i.e. true negatives while 12% of the non-demented cases were incorrectly identified as false positives.

As mentioned in the first paragraph of this section, our goal was to increase the specificity and maintain a good balance between sensitivity and specificity. Consequently, we can assert on our results that RF classifier maintains this equilibrium thus resulting in a highly specific test. On the contrary, we cannot influence sensitivity and specificity directly.

5.3. Precision-recall (PR) curve

PR curve displays the relationship between precision and recall i.e. positive predictive value and the sensitivity (Saito and Rehmsmeier, 2015). From Fig. 20, we could easily note that precision falls at 94% while recall at 80%. This we have already observed in the above results. The primary aim of plotting PR curve is that through visualization we are able to get the grasp of working of RF classifier more clearly. As precision gives the fraction of data instances that our model claimed was relevant were actually relevant while at the same time, recall presents the ability of RF classifier to find all the relevant data points present in the MRI dataset.

5.4. Receiver operator characteristic (ROC) curve

In ROC curve, a graph between true positive rate i.e. sensitivity and false positive rate i.e. (1-specificity) is plotted (Fawcett, 2006).

Table 8
Sensitivity and specificity result.

Sensitivity	80.0%
Specificity	88.0%

Each data point on the ROC curve denotes sensitivity-specificity pair. For an overall accuracy, ROC curve is the most powerful plot that helps in determining an overall classification accuracy of the AD test in which the closer the curve is towards the upper left corner, the higher is the accuracy. From Fig. 21, it can be seen that the ROC curve for RF classifier lean towards the upper right corner, making it certain that we ended at higher accuracy for our AD detection model.

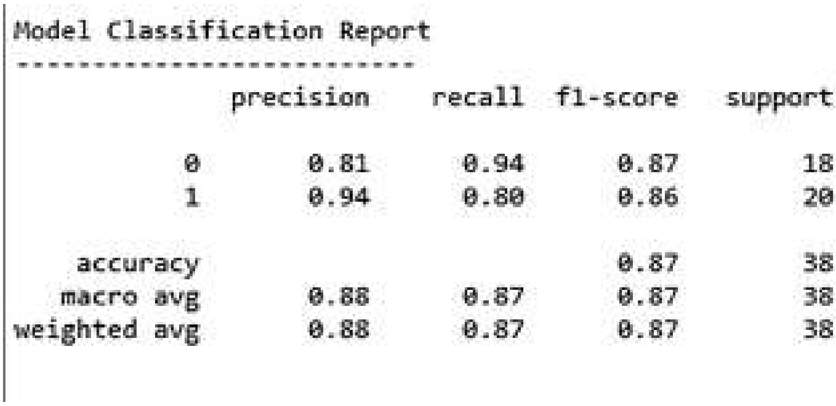
The area under the ROC curve is known as AUC. It can be seen from the above figure that the AUC score comes out to be almost equal to 87.0%. The same score is reported in model performance metrics, as illustrated in Fig. 18. Thus, this score of AUC is a higher number that indicates RF classifier performs much better in classifying the AD and non-AD patients.

6. Discussion

In the *Model Prediction* section (Section 4.4), for an unbiased comparison, the classification accuracies of all ML methods applied on MRI dataset have been stated. Out of which, RF classifier outperforms the other classifiers, which assert that our proposed method of pipelining in the detection of AD proves to perform better. The accuracy, recall and specificity of our model are significantly better than other approaches. Considering the fact that we are proposing a pipeline method, it can be comprehended that our approach is effective which can actually detect, perform an improved diagnosis and classify healthy and inflicted AD patients.

To evaluate the robustness, we have applied a 10-fold cross-validation technique and stated the accuracy scores as well. Our method attains high performance i.e. an accuracy above 80%. Usually, when employing computer-based methods for diagnosis, a little portion of data is present. Thus, in our modeling, we maintained the ratio of train-test through a random selection. The results of our work are promising. In our MRI dataset, the RF classifier by the imputation of missing values significantly improved the diagnostic accuracy of the prevalence of AD. When we compared to the results with that of a non-imputation method of missing values, the same RF classifier resulted in a decreased accuracy of only 84%.

The increase in specificity and sensitivity of our approach results in a significant output. Because of high specificity, few



```
Model Classification Report
---------------------------
              precision    recall  f1-score   support

           0       0.81      0.94      0.87        18
           1       0.94      0.80      0.86        20

    accuracy                           0.87        38
   macro avg       0.88      0.87      0.87        38
weighted avg       0.88      0.87      0.87        38
```

Fig. 19. Classification report for RF algorithm.

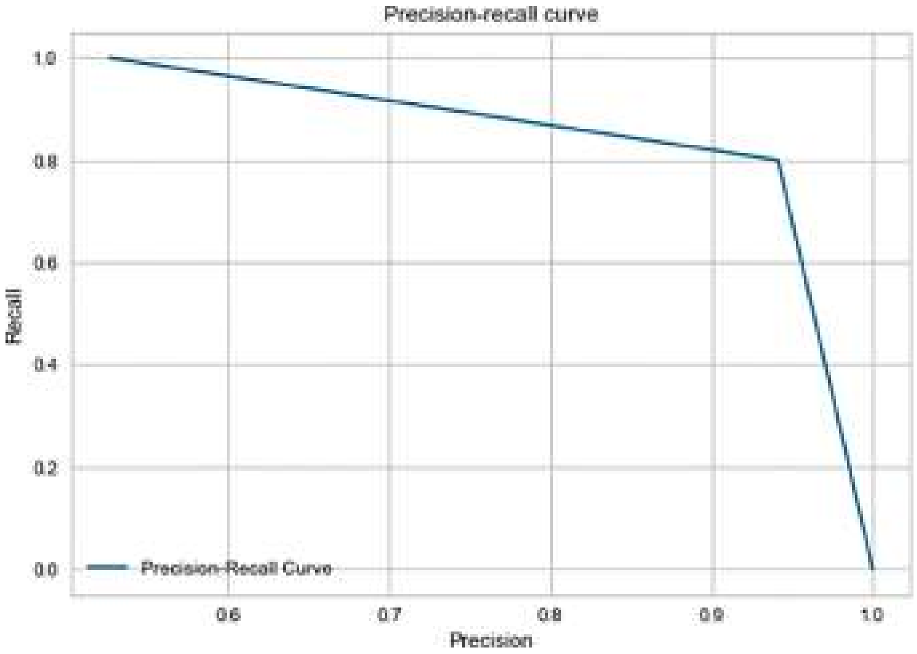


Fig. 20. Precision-recall curve of RF classifier.

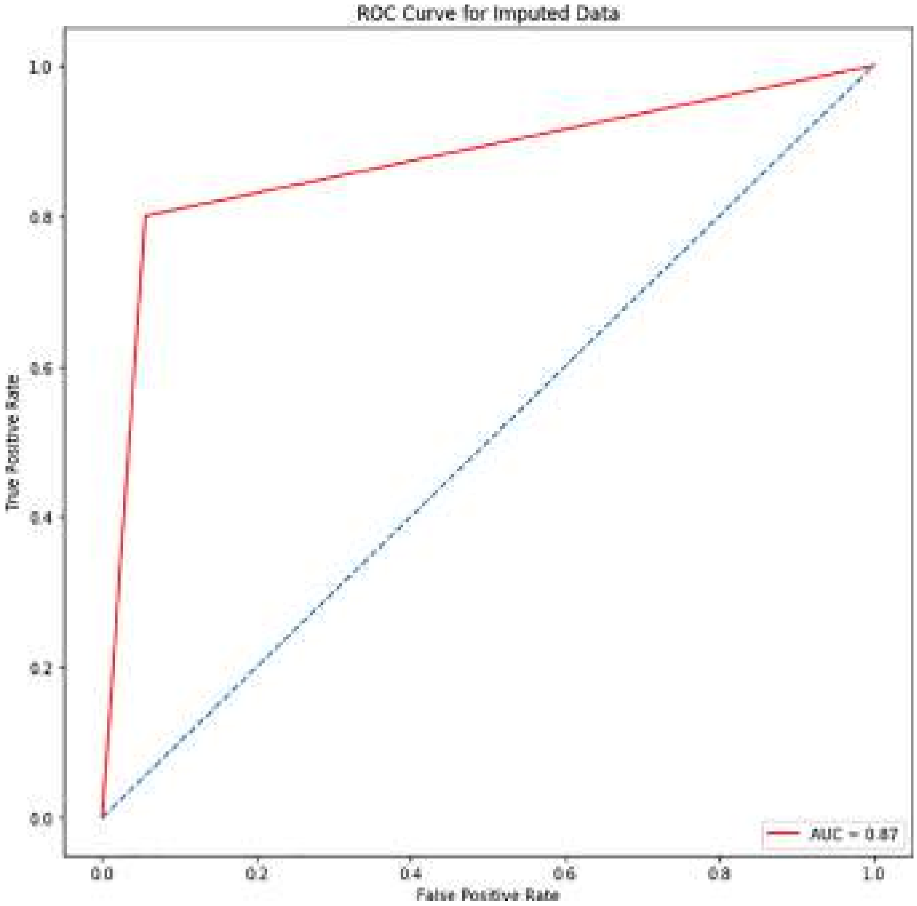


Fig. 21. ROC curve of RF classifier.

patients without dementia would have to be tested with dementia which is invasive, hence hazardous. But together with high sensitivity, this would reduce the waiting time of the actual ill patients and save money too. Secondly, we ended up at noteworthy results, as using ML methods we achieved the classification accuracy of 87%, specificity of 88% and sensitivity of 80% from the evaluation of merely few MRI features. This is due to high sensitivity, that our model gave much better results than previous methods applied by other researchers. The third remarkable result is that, only with eight MRI features, we were able to predict the correct classification of patients as healthy or unhealthy, with a maximum accuracy of the test that can be achieved. Only a handful of features from the longitudinal set of MRI data, contribute to the advanced diagnostic performance of the AD test significantly. Thus, this set of features should not be omitted from the analytical process.

The most noteworthy outcome of our study is the advancement in the predictive power of the diagnostic procedure. But at the same time, highlighting the fact that our study results were achieved on a restricted set of population and that may not be applied to the wide range of population, in general. Thus, broad studies might be required in order to validate our findings.

7. Conclusion

We propose a general framework based on supervised learning for the diagnosis of AD to categorize longitudinal brain MRI data into two classes, demented and non-demented. The proposed pipeline is fully automated and it reports the complete monitoring of AD. Our end purpose was to predict the results of the classification of AD from all the available data by employing machine learning methods in a way actual increase and come up with higher accuracy and improved performance. It gives an interpretation of the findings via the rules which are employed during the performance assessment stage i.e. the last stage. Moreover, the extraction and selection of MRI features are independent from the ML classifier being used in this study. Thus, providing the physicians with a blend of various features indicative of the detection and complete monitoring of the disorder. In contrast with the classical methodology, our pipelining approach selected the Random Forest classifier with imputation by a median, at its peak point of the ROC curve. Hence, resulting in the highest accuracy amongst all other learning classifiers. We empirically showed the efficiency of our technique in terms of accuracy, specificity and sensitivity. Even though the attained results are better than most of the other studies (87% accuracy), much advancement and broad improvements are needed in order to enhance the model for the AD diagnosis. The proposed modeling technique can discover its suitable application, likewise as a diagnostic tool. Also, in the evaluation of AD therapeutic processes.

8. Human and animals rights

This study does not contain any studies with human or animal participants performed by any of the authors.

Informed consent

Not applicable.

Author contributions

AK and SZ developed the theory, contributed to the design of the work, execution of the analysis, performed the computations and interpretation of results. SZ verified the analytical methods. AK wrote the manuscript with support from SZ. SZ contributed to supervising the work and revising the manuscript. Both the authors read and agreed to the final manuscript.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The data used in this study were obtained from the Open Access Series of Brain Imaging (OASIS) database (https://www.oasis-brains.org/), made available by the Washington University ADRC. Longitudinal MRI data was retrieved from the following published NIH grants: P50 AG05681, P01 AG03991, R01 AG021910, P50 MH071616, U24 RR021382, R01 MH56584.

References

Robinson, L., Tang, E., Taylor, J.P., 2015. Dementia: timely diagnosis and early intervention. BMJ 350 (June), 1–6.

Chapman, D.P., Williams, S.M., Strine, T.W., Anda, R.F., Moore, M.J., 2006. Dementia and its implications for public health. Prev. Chronic Dis. 3 (2), 1–13.

World Health Organization, 2008. Mental Health Gap Action Programme - scaling up care for mental, neurological, and substance use disorders. World Heal. Organ., 44

Alzheimer's Disease Facts and Figures, 2019.

Prince, M., Bryce, R., Albanese, E., Wimo, A., Ribeiro, W., Ferri, C.P., 2013. The global prevalence of dementia: a systematic review and metaanalysis. Alzheimer's Dement. 9 (1), 63–75.e2.

Nichols, E., 2019. Global, regional, and national burden of Alzheimer's disease and other dementias, 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016. Lancet Neurol. 18 (1), 88–106.

Pellegrini, E. et al., 2018. Machine learning of neuroimaging for assisted diagnosis of cognitive impairment and dementia: a systematic review. Alzheimer's Dement. Diagnosis, Assess. Dis. Monit. 10 (August), 519–535.

Hanyu, H., Sato, T., Hirao, K., Kanetaka, H., Iwamoto, T., Koizumi, K., 2010. The progression of cognitive deterioration and regional cerebral blood flow patterns in Alzheimer's disease: a longitudinal SPECT study. J. Neurol. Sci. 290 (1–2), 96–101.

Gray, K.R., Wolz, R., Heckemann, R.A., Aljabar, P., Hammers, A., Rueckert, D., 2012. Multi-region analysis of longitudinal FDG-PET for the classification of Alzheimer's disease. Neuroimage 60 (1), 221–229.

Liu, F., Zhou, L., Shen, C., Yin, J., 2014. Multiple kernel learning in the primal for multimodal alzheimer's disease classification. IEEE J. Biomed. Heal. Inf. 18 (3), 984–990.

Khajehnejad, M., Saatlou, F.H., Mohammadzade, H., 2017. Alzheimer's disease early diagnosis using manifold-based semi-supervised learning. Brain Sci. MDPI 7 (8), 1–19.

Lama, R.K., Gwak, J., Park, J.S., Lee, S.W., 2017. Diagnosis of Alzheimer's disease based on structural MRI images using a regularized extreme learning machine and PCA features. J. Healthc. Eng. 1, 2017.

Shamonin, D.P., Bron, E.E., Lelieveldt, B.P.F., Smits, M., Klein, S., Staring, M., 2014. Fast parallel image registration on CPU and GPU for diagnostic classification of Alzheimer's disease. Front. Neuroinform. 7, 1–15.

Khan, Afreen, Zubair, Swaleha, 2018. Machine Learning Tools and Toolkits in the Exploration of Big Data. International Journal of Computer Sciences and Engineering 6 (12), 570–575.

Khan, Afreen, Zubair, Swaleha, 2019. Usage Of Random Forest Ensemble Classifier Based Imputation And Its Potential In The Diagnosis Of Alzheimer's Disease. INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH 8 (12), 271–275.

Khan, Afreen, Zubair, Swaleha, 2020. Longitudinal Magnetic Resonance Imaging as a Potential Correlate in the Diagnosis of Alzheimer Disease: Exploratory Data Analysis. JMIR BIOMEDICAL ENGINEERING 5 (1). https://doi.org/10.2196/14389 https://biomedeng.jmir.org/2020/1/e14389/, .

Khan, Afreen, Zubair, Swaleha, Sabri, Muaadhabdo Al, 2019. An Improved Pre-processing Machine Learning Approach for Cross-Sectional MR Imaging of Demented Older Adults. IEEE https://ieeexplore.ieee.org/document/9035164.

Klöppel, S., Abdulkadir, A., Jack Jr., C.R., Koutsouleris, N., Mourao-Miranda, J., Vemuri, P., 2012. Diagnostic neuroimaging across diseases. Neuroimage 61 (2), 457–463.

O'Brien, J.T., 2007. Role of imaging techniques in the diagnosis of dementia. Br. J. Radiol. 80 (2), 71–77 (special issue).

Falahati, F., Westman, E., Simmons, A., 2014. Multivariate data analysis and machine learning in Alzheimer's disease with a focus on structural magnetic resonance imaging. J. Alzheimer's Dis. 41 (3), 685–708.

Shao, Y.H., Chen, W.J., Zhang, J.J., Wang, Z., Deng, N.Y., 2014. An efficient weighted Lagrangian twin support vector machine for imbalanced data classification. Pattern Recognit. 47 (9), 3158–3167.

Nasiri, J.A., Moghadam Charkari, N., Mozafari, K., 2014. Energy-based model of least squares twin Support Vector Machines for human action recognition. Signal Process. 104, 248–257.

Alam, S., Kwon, G., Kim, J., Park, C., 2017. Twin SVM-based classification of Alzheimer's disease using complex dual-tree wavelet principal coefficients and LDA. J. Healthc. Eng. 2017.

Marcus, D.S., Fotenos, A.F., Csernansky, J.G., Morris, J.C., Buckner, R.L., 2010. Open access series of imaging studies: longitudinal MRI data in nondemented and demented older adults. J. Cogn. Neurosci. 22 (November 12), 2677–2684.

Lynch, J., Kaplan, G., 2000. Socioeconomic position. In: Berkman, L., Kawachi, I. (Eds.), Social Epidemiology. Oxford University Press, New York, pp. 13–35.

Arevalo-rodriguez, I. et al., 2015. Mini-Mental State Examination (MMSE) for the detection of Alzheimer's disease and other dementias in people with mild cognitive impairment (MCI). B. J. Psych. Adv. 21 (3).

Morris, J.C., 1993. The Clinical Dementia Rating (CDR): current version and scoring rules. Am. Acad. Neurol.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J.T., Blum, M., Hutter, F., 2015. Efficient and robust automated machine learning. In: NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems, 2015, pp. 2755–2763.

Sivarajah, U., Kamal, M.M., Irani, Z., Weerakkody, V., 2017. Critical analysis of Big Data challenges and analytical methods. J. Bus. Res. 70, 263–286.

Teng, C.M., 1999. Correcting noisy data. In: Proc 16th International Conf on Machine Learning, 1999, pp. 239–248.

Kotsiantis, S.B., Kanellopoulos, D., Pintelas, P.E., 2006. Data Preprocessing for supervised leaning. Proc. World Acad. Sci. Eng. Technol. 12, 1–7.

NIST/SEMATECH e-Handbook of Statistical Methods, 2003. Available at: http://www.itl.nist.gov/div898/handbook/ (online).

Kwak, S.K., Kim, J.H., 2017. Statistical data preparation: management of missing values and outliers. Korean J. Anesthesiol. 70 (4), 407–411.

Alhaj, T.A., Siraj, M.M., Zainal, A., Elshoush, H.T., Elhaj, F., 2016. Feature selection using information gain for improved structural-based alert correlation. PLoS One 11 (11), 1–18.

Pampaka, M., Hutcheson, G., Williams, J., 2014. Handling missing data: analysis of a challenging data set using multiple imputation. Int. J. Res. Method Educ. vol, 7288.

Cao, Y., Miao, Q.G., Liu, J.C., Gao, L., 2013. Advance and prospects of AdaBoost algorithm. Acta Autom. Sin. 39 (6), 745–758.

Geurts, P., Ernst, D., Wehenkel, L., 2006. Extremely randomized trees. Mach. Learn. 63 (October 2005), 3–42.

Natekin, A., Knoll, A., 2013. Gradient boosting machines, a tutorial. Front. Neurorobot. 7 (Dec), 1–21.

Denisko, D., Hoffman, M.M., 2018. Classification and interaction in random forests. Proc. Natl. Acad. Sci. USA 115 (8), 1690–1692.

Csató, L., Fokoué, E., Opper, M., Schottky, B., Winther, O., 2000. Efficient approaches to Gaussian process classification. Adv. Neural Inf. Process. Syst., 251–257

Pedregosa, F., Varoquaux, G., et al., 2011. Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Crammer, K., Dekel, O., et al., 2006. Online passive-aggressive algorithms Koby. J. Mach. Learn. Res. 7, 551–585.

Robbins, H., Monro, S., 1951. A stochastic approximation method. Ann. Math. Stat. 22 (3), 400–407.

Ruder, S., 2016. An overview of gradient descent optimization algorithms. Available at: http://ruder.io/optimizing-gradient-descent/ (online).

Gardner, M.W., Dorling, S.R., 1998. Artificial neural networks (the multilayer perceptron) - a review of applications in the atmospheric sciences. Atmos. Environ. 32 (14–15), 2627–2636.

Zhang, Z., 2016. Introduction to machine learning: k-nearest neighbors. Ann. Transl. Med. 4 (11), 1–7.

Amancio, D.R., Comin, C.H., et al., 2014. A systematic comparison of supervised classifer - supporting information. PLoS One 9 (4), 1–13.

Kotsiantis, S.B., 2007. Supervised machine learning: a review of classification techniques. Informatica 31, 249–268.

Arlot, S., Celisse, A., 2010. A survey of cross-validation procedures for model selection. Stat. Surv. 4, 40–79.

Tharwat, A., 2018. Classification assessment methods. Appl. Comput. Inf., 1–13

Fawcett, T., 2006. An introduction to ROC analysis. Pattern Recognit. Lett. 27 (8), 861–874.

Saito, T., Rehmsmeier, M., 2015. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLoS One 10 (3), 1–21.