International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Analysis of State of Art Machine Learning Models for Classification Prediction of Alzheimer's Disease

Dhwani Modi[1], Seema Mahajan[2]

Submitted:10/03/2024 **Revised**: 25/04/2024 **Accepted**: 02/05/2024

Abstract: Alzheimer's disease (AD) is a progressive neurodegenerative disorder that leads to dementia and cognitive impairment, significantly contributing to age-related cognitive decline and widespread memory loss. As the population ages, AD will increasingly impact individuals, their families, and healthcare systems, resulting in substantial social, financial, and economic challenges for aging societies. Early detection of AD is crucial, as it allows for more effective intervention compared to treatment at later stages of the disease. Machine Learning (ML) techniques, applied to Magnetic Resonance Imaging (MRI) data, offer a promising approach for the early detection of AD. This study evaluates twenty different ML models to classify dementia patients as either AD or non-AD. The performance of these ML models is assessed using metrics such as Precision, Recall, Accuracy, and F1-score.

Keywords*: Alzheimer's disease (AD), Dementia, Machine Learning, Classification, Prediction, Performance*

1. Introduction

Alzheimer's disease (AD) is a progressive neurodegenerative disorder that represents the most common form of dementia (Goyal, P., Rani, R., & Singh, K., 2022). Characterized by a gradual decline in cognitive functions, AD leads to impairments in memory, reasoning, and daily functioning. The disease's hallmark features include the accumulation of amyloid plaques and tau tangles in the brain, which are believed to disrupt neural communication and lead to neuronal death. Globally, approximately 50 million individuals are affected by AD, a number that is projected to rise significantly as the population ages (Goyal, P., Rani, R., & Singh, K., 2022).

The progression of AD is insidious, often beginning with mild cognitive impairment (MCI) that gradually worsens over time (Arafa. Et. all, 2022). Early symptoms, such as subtle memory lapses and difficulties with complex tasks, may be mistakenly attributed to normal aging or other benign conditions. By the time a definitive diagnosis is made, the disease may have already reached an advanced stage, making treatment options less effective and limiting opportunities for early intervention (Arafa. Et. all, 2022).

This delay in diagnosis presents substantial challenges. Effective management of AD is heavily dependent on early detection and timely intervention (Tang, X., & Liu, J., 2021). Early diagnosis not only allows for the initiation of treatments that can slow disease progression but also provides individuals and their families with valuable time to plan for the future. Current diagnostic methods, including clinical assessments, neuropsychological testing, and neuroimaging techniques such as MRI, are essential tools in the diagnostic process. However, these methods often require specialist expertise and may not always capture the subtle changes associated with the early stages of the disease.

Given these limitations, there is an urgent need for innovative approaches that can enhance early detection and improve diagnostic accuracy (Jo, T. et all, 2019). Machine learning (ML) offers a promising avenue for addressing these challenges. By analyzing large volumes of complex data, ML algorithms can uncover patterns and relationships that may be imperceptible through traditional methods. In particular, longitudinal brain MRI data presents a rich source of information that, when coupled with advanced ML techniques, has the potential to significantly improve our ability to detect Alzheimer's disease at an earlier stage (Jo, T. et all, 2019).

Recent advances in machine learning and data analytics have demonstrated the potential to transform the field of AD diagnosis. Algorithms can be trained to recognize subtle changes in brain structure and function that precede clinical symptoms. This ability to detect early biomarkers of the disease could lead to more precise and timely diagnoses, ultimately enhancing patient outcomes and optimizing treatment strategies (Ramzan et. all. 2022).

In summary, Alzheimer's disease poses a major global health challenge, with current diagnostic methods often failing to detect the disease until it has progressed significantly. Machine learning presents a novel approach

[1]*Computer Engineering Department, Indus University, Rancharda, Via, Shilaj, Ahmedabad,*
Email ID: dhwanimodi.21.rs@indusuni.ac.in
[2] *Computer Engineering Department, Indus University, Rancharda, Via, Shilaj, Ahmedabad,*
Email ID: ce.hod@indusuni.ac.in

to improving early detection by analyzing complex datasets such as longitudinal MRI scans. This study aims to explore the effectiveness of various machine learning algorithms in classifying Alzheimer's disease, with the goal of identifying the most accurate and clinically relevant models for early diagnosis (Ramzan et. all. 2022).

1.1. *Problem Statement*

Current diagnostic approaches for Alzheimer's disease include clinical assessments, neuropsychological tests, and imaging techniques such as MRI. While these methods can provide valuable insights, they often require expert interpretation and are not always accessible or cost-effective (Ramzan et. all. 2022). Traditional diagnostic processes may also struggle to detect the disease in its earliest stages, where intervention could be most beneficial. Machine learning has emerged as a promising tool to enhance diagnostic accuracy by analyzing large-scale data sets to uncover patterns and relationships that may not be immediately evident to clinicians.

1.2. *Objective*

The objective of this study is to leverage machine learning algorithms to improve the classification of Alzheimer's disease using longitudinal brain MRI data. Specifically, we aim to evaluate the performance of various machine learning models in distinguishing between demented and non-demented individuals. By systematically comparing twenty different algorithms, we seek to identify the most effective models for early AD detection and provide insights into their applicability in clinical settings.

1.3. *Contributions*

This paper makes several key contributions to the field of Alzheimer's disease detection:

1. Model Comparison: We conduct a comprehensive comparison of twenty machine learning models, including both traditional and advanced techniques, to assess their effectiveness in classifying Alzheimer's disease.

2. Performance Evaluation: We present detailed performance metrics, including Precision, Recall, F1-score, and Accuracy, for each model, highlighting their strengths and weaknesses.

3. Practical Insights: We identify the ExtraTree classifier as the most accurate model for this task, offering valuable insights into its potential application for early diagnosis of Alzheimer's disease.

By exploring these aspects, this study aims to advance the use of machine learning in the early detection of Alzheimer's disease and contribute to the development of more effective diagnostic tools.

2. Literature Review

This section summarizes previous research in Alzheimer's disease (AD) prediction using various machine learning (ML) techniques applied to longitudinal datasets. Techniques such as Decision Trees, Random Forests, Support Vector Machines, Gradient Boosting, and Voting classifiers have been employed to extract the most relevant features for AD prediction (Kavitha, C et. all, 2022). A study reported an improved average validation precision of 83% on the test data for AD using these methods . Another study utilized longitudinal brain MRI characteristics to classify dementia subjects as AD or non-AD using six supervised classifiers—Gradient Boosting, SVM, LR, RF, AdaBoost, and Naive Bayes—demonstrating the superior performance of the Gradient Boosting method with a classification accuracy of 97.58% .

The effectiveness of different ML methods in predicting the possible conversion from MCI to AD was also analyzed, with LADTree, ADTree, and FT classifiers selected as the best based on AUC for Month 12, Month 24, and Month 36, respectively (Battineni, G. et all, 2021). Another study explored the role of longitudinal MRI in exploratory data analysis (EDA) to estimate the scores of MMSE, CDR, and ASF, which were used to identify AD patients. The significance of the results was demonstrated using a 1.5-Tesla Vision (Siemens) scanner for image acquisition (Battineni, G. et all, 2021).

A novel ML model based on longitudinal MRI data, incorporating features such as the Mini-Mental State Examination (MMSE) score and years of education, was proposed. This model achieved 5-fold cross-validation with 94.64% accuracy and an AUC of 89.93% . Additionally, a hybrid classification system for distinguishing NC, MCI, and AD based on structural MRI images was suggested, with the kSVM-DT model achieving 80% accuracy in classification using 22 features, outperforming the 74% accuracy obtained without using a kernel (Khan, A., & Zubair, S. 2020).

Further research developed a model that performed 5-fold cross-validation to determine the best parameters for each ML model, achieving an accuracy, recall, and AUC of 84%, 80%, and 84%, respectively (Tripathi, S., et all, 2023). Another study proposed an unorganized framework reflecting Alzheimer's rate and characteristics using several ML algorithms, with SVM employing a linear kernel achieving a classification accuracy of 95%. A classifier system built using an ML pipeline, data transformation, and feature selection methods showed that the Random Forest (RF) classifier had the highest accuracy, recall, and AUC, with values of 85%, 80%, and 84%, respectively (Tripathi, S., et all, 2023) .

A study focusing on dementia diagnosis assessed the effectiveness of a dimensionality reduction method using cross-sectional MRI data, based on CDR scores and various ML models, achieving an overall accuracy of 87% (Zhang, Y. D. et. all, 2014). Another investigation proposed feature extraction methods based on Principal Component Analysis (PCA), enhanced by Linear Discriminant Analysis (LDA) or the Fisher Discriminant Ratio (FDR) for feature selection, with combined strategies generating accuracy rates between 89.52% and 96.7% using SPECT and PET images.

Various ML models have been proposed for classifying different stages of Alzheimer's disease by considering cognitive tests, physical examinations, age, mental status examinations, and laboratory investigations (Gopagoni, D., et. all, 2020). The classification accuracy for the proposed CANFIS model was estimated at 99.55%, outperforming other classification methods. A local/regional computer-aided diagnosis (CAD) system developed for the early diagnosis of AD went through preprocessing, brain labeling, feature extraction, statistical analysis, and diagnosis, with Linear-SVM showing an accuracy of 96%, specificity of 100%, and sensitivity of 93.1%.

In addition to healthy individuals, research has applied a deep learning-based method using fMRI and PET scans of Alzheimer's patients, involving image resizing and 3D to 2D conversion, followed by feature extraction using the VGG-16 architecture (Kishore, P., et all, 2021). This approach achieved an average accuracy of 99.95% for the fMRI dataset and 73.46% for the PET dataset . Six deep learning-based models were used to identify distinctive signs of cerebral amyloid angiopathy (CAA) and AD, achieving precision, sensitivity, and F1-scores of 89.08%, 87.87%, and 92.41%, respectively (Kishore, P., et all, 2021).

A deep convolutional neural network was proposed for the early diagnosis of AD, achieving accuracies of 75.00%, 82.50%, 73.75%, and 93.18% for inception-v4, ResNet, ADNet, and the proposed model, respectively (Khan, A., & Zubair, S. 2020). A hybrid model using ResNet50 as a baseline, along with AlexNet, ResNet50, DenseNet201, and VGG16, achieved an accuracy of 90%. SegNet was proposed for deep learning-based segmentation to detect AD-related brain features from structural MRI, achieving a prediction accuracy of 96.35%, sensitivity of 96.7%, and specificity of 93.9% with the ResNet-101 model.

A multimodal diagnosis approach for AD was proposed using a Principal Component Analysis Network (PCANet) and three-dimensional ShuffleNet (3DShuffleNet), combining sMRI and fMRI data for feature extraction and fusing these characteristics using kernel canonical correlation analysis (KCCA)

(López, M. et all, 2011). A unique algorithm was developed by modifying the capsule network's architecture, resulting in an accuracy of 92.39%, the highest among compared techniques. Another deep model was proposed to analyze imaging (MRI), single nucleotide polymorphisms (SNPs), genetic data, and electronic health records (EHR) for classifying patients into AD, MCI, and controls (CN), achieving an accuracy, precision, recall, and F1-score of 78.00%, 77.00%, 78.00%, and 78.99%, respectively (López, M. et all, 2011).

A new voxel-based hierarchical feature extraction (VHFE) technique was suggested for early AD detection, obtaining classification results of 97.8% (AD vs. MCI), 99.7% (AD vs. NC), and 97.7% (NC vs. MCI) using Kendall's rank correlation. A layer-wise transfer learning model using the VGG architecture family with pre-trained weights was proposed, achieving classification accuracy of 98.73% for AD/NC and 83.72% for EMCI/LMCI. The Iterative Sparse and Deep Learning (ISDL) model, designed for identifying cortical regions and extracting deep features, achieved accuracy, sensitivity, and specificity of 95.32%, 91.18%, and 93.94%, respectively. A novel methodology using mobility data and deep learning models correlated the daily activities of Alzheimer's patients with disease stages, achieving an accuracy of 90.91% (Joshi, S., et all, 2010).

3. Materials and Methods

3.1. *Dataset: Description of the OASIS Dataset*

For this study, we utilized the Open Access Series of Imaging Studies (OASIS) dataset, a comprehensive resource designed for research in neuroimaging and cognitive disorders. The OASIS dataset provides a rich collection of longitudinal brain MRI scans and associated clinical data, making it a valuable resource for studying Alzheimer's disease (AD) and other neurodegenerative conditions.

Dataset Overview

The OASIS dataset consists of MRI scans from a diverse cohort of participants, including both cognitively normal individuals and those diagnosed with various stages of dementia. The dataset is publicly available and widely used in the research community for its high-quality imaging data and detailed clinical annotations. It is managed by the Washington University School of Medicine and is available for research through the OASIS website.

Table 1 Dataset Attribute description

Attribute Name	Attribute Description
Subject ID	Subject identification number
MRI ID	Image identification number of an individual subject
Group	Demented/Nondemented/Converted
Visit	Number of subjects visit
MR Delay	Magnetic resonance delay is the delay time that is before the image Procurement
Gender	Male/Female
Hand	Right/Left-Handed
Age	Subject age while scanning
EDUC	Subject educational level
SES	Socioeconomic status
MMSE	Mini-Mental state examination score
CDR	Clinical dementia rating score
eTIV	Estimated total intracranial volume
nWBV	Normalized whole brain volume result
ASF	Atlas scaling factor

Participant Groups

The dataset includes longitudinal data collected over multiple visits for each participant. The primary groups in the dataset are:

- Cognitively Normal (Non-demented): Participants with no significant cognitive impairment, serving as a control group.

- Mild Cognitive Impairment (MCI): Participants with cognitive decline that is greater than expected for their age but not severe enough to meet the criteria for dementia.

- Alzheimer's Disease (AD): Participants diagnosed with Alzheimer's disease, representing different stages of the disease.

Imaging Data

The MRI scans provided in the OASIS dataset are acquired using high-resolution imaging protocols. The dataset includes:

- Structural MRI: Detailed images of brain anatomy, which are critical for assessing changes in brain structures associated with AD.

- T1-Weighted MRI: Commonly used to evaluate brain morphology and identify structural abnormalities.

- Longitudinal MRI Scans: Multiple scans per participant, allowing for the analysis of changes in brain structure over time.

Clinical Data

In addition to imaging data, the OASIS dataset includes a wealth of clinical information:

- Cognitive Assessments: Scores from various neuropsychological tests that measure different aspects of cognitive function, such as memory, executive

function, and language.

- Demographic Information: Data including age, sex, education level, and other relevant variables.

- Clinical Diagnoses: Information on the clinical diagnosis of each participant, including details on the severity of cognitive impairment.

Preprocessing

Before analysis, the MRI images were preprocessed to ensure consistency and quality. Preprocessing steps included:

- Motion Correction: To minimize artifacts due to participant movement during scans.

- Normalization: To standardize the images to a common anatomical template, allowing for accurate comparisons across subjects.

- Segmentation: To identify and quantify different brain structures and regions of interest.

Data Splitting

For the purpose of model training and evaluation, the dataset was split into training and testing subsets. The training set was used to train various machine learning algorithms, while the testing set was reserved for evaluating the performance of the models.

The OASIS dataset's comprehensive nature, including both longitudinal imaging and clinical data, provides a robust foundation for developing and validating machine learning models aimed at early detection of Alzheimer's disease. By leveraging this rich dataset, the study aims to enhance our ability to classify and predict the presence of AD, ultimately contributing to improved diagnostic capabilities and patient outcomes.

3.2. *Data Preprocessing*

The raw MRI data underwent a comprehensive preprocessing pipeline consisting of four main stages, each contributing to the overall quality and usability of the data for analysis. This process is linear and systematic, ensuring that the data is adequately prepared for subsequent analytical tasks. The preprocessing stages are outlined below:

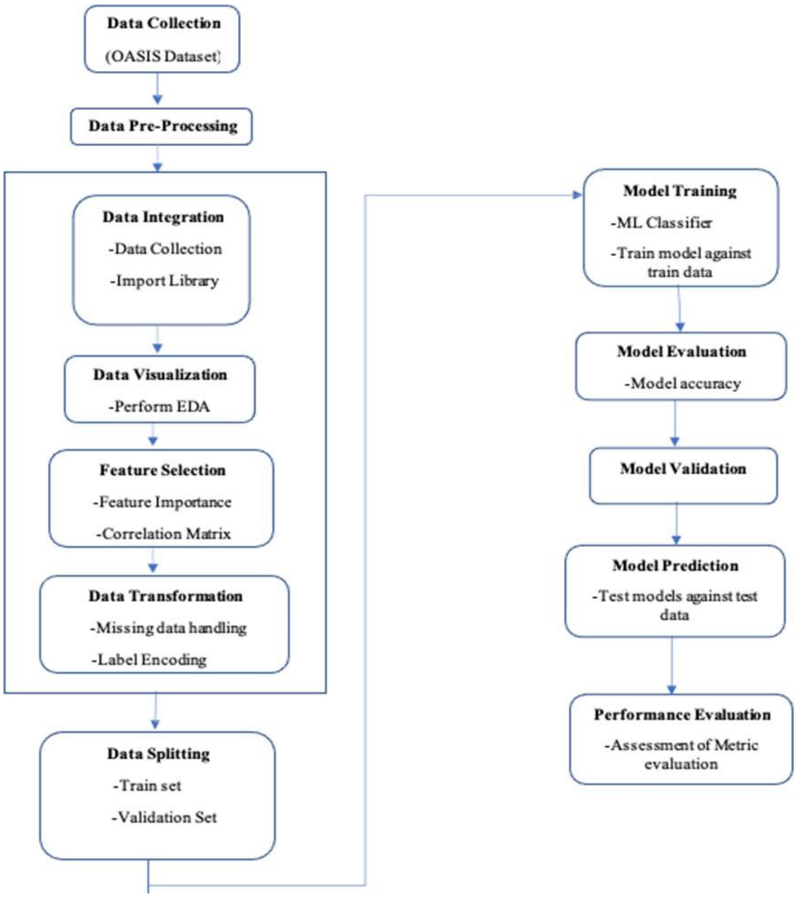


Fig.1 Experimental setup for proposed work

Data Integration:

- ETL Process:

- Extract: The initial phase involves gathering raw data from its various sources. This data collection step ensures that all relevant information is consolidated.

- Transform: In this phase, the extracted data undergoes cleaning and normalization. Data cleaning addresses issues such as missing values and inconsistencies, while normalization standardizes the data to a common scale to ensure comparability.

- Load: The cleaned and normalized data is then loaded into a database or data

warehouse. This step makes the data readily accessible for further analysis and modeling.

Data Visualization:

- Exploratory Data Analysis (EDA): This step employs a range of techniques and tools to gain insights from the data. EDA combines statistical methods and graphical representations to explore data patterns, identify trends, and understand the relationships between variables.

Feature Selection:

- Automated Feature Selection: The dataset is analyzed to identify and select features that are most predictive of the outcome variable. This process helps in reducing dimensionality and enhancing model performance by focusing on the most relevant features.

Data Deduplication and Handling Missing Values:

- De-Duplication: To ensure data integrity and avoid redundancy, the dataset is cleaned of duplicate entries. This step prevents the use of repeated or redundant information that could skew results.

- Handling Missing Values: Both removal and imputation techniques are applied to address missing data. Missing values are either filled using statistical methods or removed if they are deemed excessive or non-representative. This ensures that the dataset is complete and reliable for analysis.

By following these preprocessing steps, the dataset is refined to eliminate errors and inconsistencies, standardize features, and enhance the quality of the data. This thorough preparation is crucial for accurate analysis and effective modeling in predicting Alzheimer's disease outcomes.

Data Splitting

For effective model evaluation and to ensure robust performance, the dataset was partitioned into three distinct subsets for cross-validation purposes:

1. Training Set: This subset is used to train the model, allowing it to learn patterns and relationships within the data.

2. Validation Set: This subset is used to assess the model's performance during training. It helps in fine-tuning the model by evaluating its performance on unseen data and adjusting hyperparameters accordingly.

3. Test Set: This subset is reserved for final evaluation. It provides an unbiased assessment of the model's performance by simulating real-world scenarios where the model is tested on data it has never encountered before.

Procedure:

- Initial Splitting: The entire dataset was randomly divided into training and test subsets in an 80:20 ratio. This means 80% of the data was allocated for training the model, while 20% was set aside for testing.

- Cross-Validation: To further evaluate the model's performance and ensure it generalizes well, the training data was further split into two subsets: one for training and one for validation. This enables the model to be trained and validated on different portions of the training data, providing insights into its performance on unseen data within the training phase.

- Early Stopping: During the training process, the model's performance was monitored on the validation subset. If the model's performance on the validation set did not improve beyond a certain threshold, or if it started to degrade, training was halted to prevent overfitting and to avoid unnecessary computation.

By using this systematic approach to data splitting, we ensure that the model is trained effectively, validated thoroughly, and tested rigorously, which aids in achieving accurate and reliable predictions.

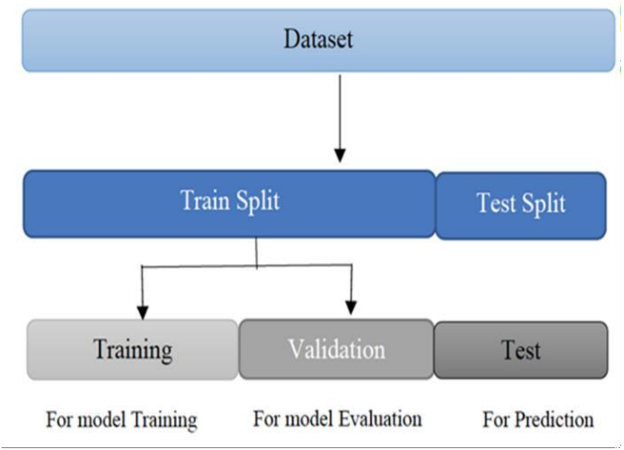


Fig. 1. Blockchain Characteristics

3.3. *Training ML Classifiers*

To effectively classify patients as demented or non-demented, we applied and evaluated a range of machine learning classifiers. These included Multi-Layer Perceptron (MLP), Logistic Regression (LR), Random Forest (RF), Bagging, Boosting, AdaBoost, XGBoost, Gradient Boosting Classifier, Histogram-based Gradient Boosting Classifier, Support Vector Machine (SVM), Decision Tree (DT), ExtraTree Classifier, Naive Bayes (NB) Classifier, Voting Classifier, Linear Discriminant

Analysis (LDA), Quadratic Discriminant Analysis (QDA), K-Nearest Neighbors (KNN), and Stochastic Gradient Descent (SGD) Classifier. Each model was trained and validated to assess its performance and accuracy in predicting Alzheimer's disease status based on the provided features.

Multi-Layer Perceptron (MLP)

Multi-Layer Perceptron (MLP) is a type of artificial neural network that extends the concept of a simple perceptron by adding multiple layers of neurons (Jyotiyana, M., & Kesswani, N. 2020). It is a fundamental architecture in deep learning and is used for a variety of tasks including classification, regression, and pattern recognition.

Architecture:

An MLP consists of three main types of layers:

1. Input Layer: This layer receives the raw input features. Each neuron in this layer represents one feature in the input data.

2. Hidden Layers: These are intermediate layers between the input and output layers. MLPs can have one or more hidden layers, and each layer contains multiple neurons. Each neuron in a hidden layer applies a weighted sum of its inputs followed by a non-linear activation function. The presence of multiple hidden layers allows the network to model complex relationships and interactions between features.

3. Output Layer: This layer produces the final output of the network. For classification tasks, the output layer typically uses a softmax or sigmoid activation function to generate class probabilities.

Working Mechanism:

- Forward Propagation: In forward propagation, input data is passed through the network layer by layer. Each neuron computes a weighted sum of its inputs, applies an activation function, and passes the result to the neurons in the next layer. This process continues until the output layer is reached.

- Activation Function: Common activation functions used in MLPs include ReLU (Rectified Linear Unit), Sigmoid, and Tanh. These functions introduce non-linearity into the network, allowing it to learn complex patterns.

- Backpropagation: During training, MLPs use backpropagation to update the weights

of the network. This involves calculating the gradient of the loss function with respect to each weight using the chain rule and adjusting the weights to minimize the error. The loss function could be Mean Squared Error for regression tasks or Cross-Entropy Loss for classification tasks.

Applications:

MLPs are versatile and can be applied to a wide range of problems, including image and speech recognition, natural language processing, and medical diagnosis. They are particularly effective when dealing with structured data and can model complex, non-linear relationships.

Strengths and Limitations:

- Strengths: MLPs are capable of learning complex patterns and interactions in data. They are flexible and can be adapted to various types of problems with appropriate modifications.

- Limitations: Training MLPs can be computationally intensive, especially with large datasets and deep architectures. They are also prone to overfitting if not properly regularized or if there is insufficient training data.

Logistic Regression (LR)

Architecture of Logistic Regression

Logistic Regression is a linear model used for binary classification, predicting the probability that a given input belongs to one of two classes (Fukunishi et. all, 2020). The architecture of Logistic Regression can be summarized as follows:

1. Input Layer:

- Features: The model accepts a set of input. Each feature represents a variable used for prediction.

2. Linear Combination:

- Weighted Sum: The features are combined linearly with associated weights (coefficients) and a bias term. This is expressed as:

3. Logistic Function (Sigmoid Function):

- Transformation: The linear combination is passed through the sigmoid function to map it to a probability value between 0 and 1.

This function transforms the output of the linear combination into a probability that the input belongs to the positive class.

4. Output Layer:

- Probability Output: The output of the sigmoid function represents the predicted probability of the input belonging to the positive class.

5. Decision Boundary:

- Classification: The decision boundary is the threshold at which the probability output changes classification. For binary classification, this is typically set at 0.5. The boundary is defined by the set of points where the predicted probability is equal to 0.5.

6. Training:

- Optimization: During training, the model learns the optimal values of the weights and the bias by minimizing the cost function. The cost function, often the log-loss or binary cross-entropy, measures the difference between the predicted probabilities and the actual class labels.

- Gradient Descent: Optimization algorithms like gradient descent are used to adjust the weights and bias to minimize the cost function.

The architecture of Logistic Regression involves a linear combination of input features, transformed by the sigmoid function to produce a probability estimate. The model's simplicity allows for efficient training and interpretation, making it suitable for many binary classification tasks.

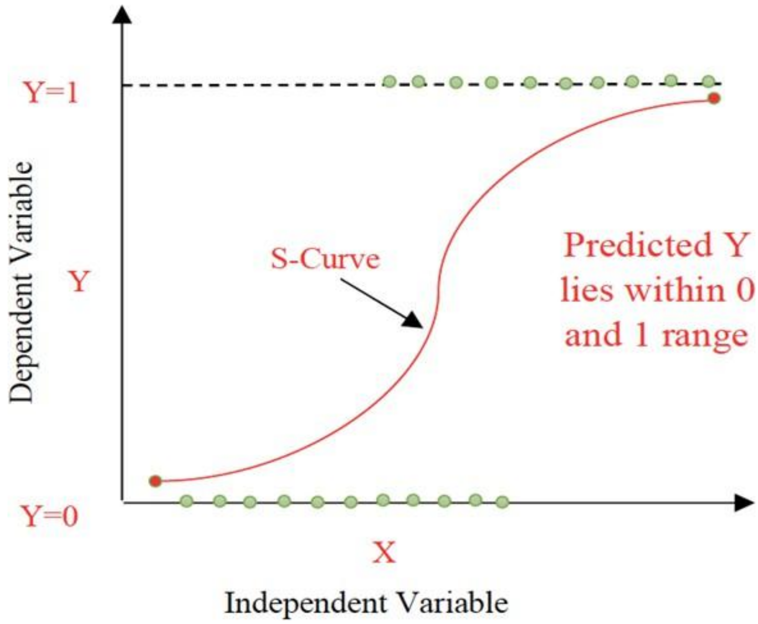


Fig. 3. Representation of LR Classifier

Random Forest (RF) Architecture

Architecture:

1. Bootstrap Sampling:

- Generates multiple training subsets from the original dataset by randomly sampling with replacement (Ali, M. S. et. all, 2021)

2. Decision Trees:

- Constructs numerous decision trees, each trained on a different bootstrap sample.

- At each node, only a random subset of features is considered for splitting, adding diversity among trees.

3. Aggregation:

- Aggregates predictions from all decision trees.

- Uses majority voting for classification tasks and averaging for regression tasks to produce the final prediction.

Bagging (Bootstrap Aggregating):

Architecture:

- Base Models: Bagging uses multiple

instances of the same model type, often decision trees. Each model is trained on a different bootstrap sample of the training data, where each sample is generated by randomly drawing with replacement from the original dataset (Logan, R. et. all, 2021).

- Aggregation: After training, the predictions from each base model are combined. For regression tasks, this is typically done by averaging the predictions. For classification tasks, a majority voting scheme is used where the most common class predicted by the base models is chosen as the final prediction.

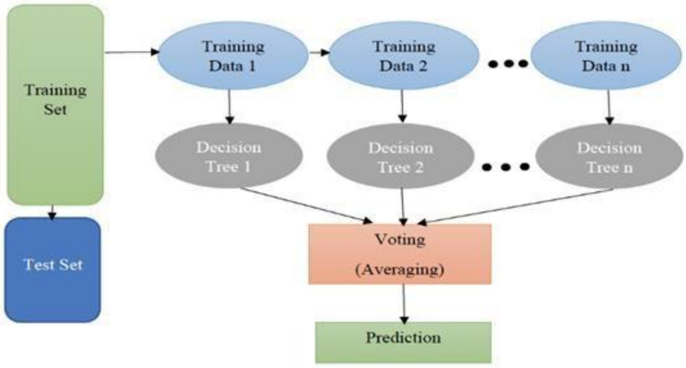


Fig. 4. Working of RF algorithm

Boosting:

Architecture:

- Sequential Learning: Boosting trains models sequentially, where each new model focuses on the errors made by the previous models (Buyrukoğlu, S., 2021). Initially, a base model is trained on the entire dataset. Subsequent models are trained to correct the errors of the previous ones, typically by adjusting the weights of incorrectly predicted instances.

- Aggregation: The final prediction is obtained by combining the weighted

predictions of all models. Each model's contribution is weighted based on its accuracy, often using techniques like weighted voting for classification or weighted averaging for regression.

AdaBoost:

Architecture:

- Sequential Boosting: AdaBoost (Adaptive Boosting) builds a series of weak learners (often decision trees) sequentially. Each new model is trained to correct the errors of the previous ones by focusing more on the misclassified instances from earlier models.

- Weight Adjustment: Initially, all training examples have equal weights. After each iteration, the weights of misclassified instances are increased, making them more influential in the next model's training. Conversely, correctly classified examples have their weights decreased.

- Model Combination: The final model prediction is a weighted vote or average of the predictions from all weak learners, where each learner's contribution is proportional to its accuracy.

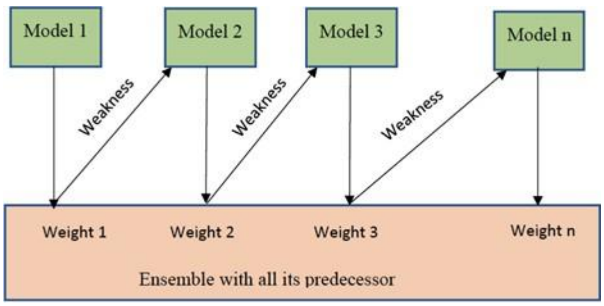


Fig. 5 AdaBoost (Boosting) Classifier

XGBoost:

Architecture:

- Gradient Boosting Framework: XGBoost builds an ensemble of decision trees in a sequential manner, where each tree aims to correct the errors made by the previous trees.

- Regularization: Incorporates L1 (Lasso)

and L2 (Ridge) regularization to prevent overfitting and enhance generalization.

- Tree Pruning: Utilizes a depth-wise splitting approach and prunes trees based on a pre-defined threshold to optimize performance and complexity.

- Handling Sparsity: Efficiently handles sparse data and missing values through automatic learning and adjustment mechanisms.

Gradient Boosting Classifier:

Architecture:

- Sequential Learning: Constructs an ensemble of weak learners (typically decision trees) sequentially. Each new model is trained to predict the residuals or errors of the existing ensemble.

- Gradient Descent Optimization: Uses gradient descent to minimize the loss function, adjusting the model iteratively by fitting new trees to the negative gradient of the loss function.

- Boosting Strategy: Combines the predictions of all weak learners using weighted averaging or voting, with each learner's weight reflecting its performance in reducing errors.

Histogram-based Gradient Boosting Classifier:

- Histogram-Based Splitting: This variant of gradient boosting uses histograms to bin continuous feature values into discrete intervals, which accelerates the computation and reduces memory usage. This approach speeds up training and makes it efficient for large datasets.

- Gradient Boosting Framework: Builds trees sequentially to correct the errors of the previous trees, with each tree focusing on the residuals of the combined predictions of all previous trees. Uses gradient descent to minimize the loss function.

Support Vector Machine (SVM):

- Hyperplane Separation: SVM constructs a hyperplane in a high-dimensional space that maximally separates data points of different classes. It aims to find the optimal hyperplane that maximizes the margin between classes (Sharma, A et. all, 2021).

- Kernel Trick: Utilizes kernel functions to transform non-linearly separable data into a higher-dimensional space where a linear separation is possible. Common kernels include linear, polynomial, and radial basis function (RBF) kernels.

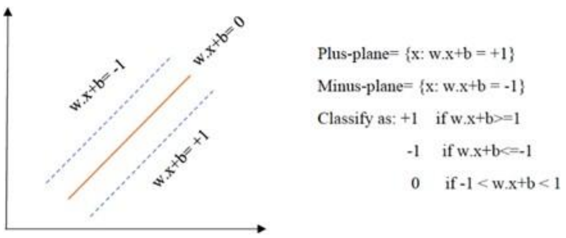


Fig. 6. Support vector machine classifier

Decision Tree (DT):

- Tree Structure: Decision Tree algorithms create a model in the form of a tree structure where internal nodes represent feature tests, branches represent the outcome of the tests, and leaf nodes represent the final decision or classification (Saputra et. all, 2020). The tree is built by recursively splitting the data based on feature values that result in the maximum information gain or impurity reduction.

- Greedy Splitting: Decision Trees use a greedy approach to find the best feature and threshold for splitting the data at each node. Common criteria for splitting include Gini impurity, entropy (for classification), and mean squared error (for regression).

ExtraTree Classifier:

- Extremely Randomized Trees: ExtraTree Classifier is similar to Decision Trees but introduces additional randomness. It selects random splits at each node without searching for the optimal split. This randomness improves model diversity and reduces overfitting.

- Randomized Splitting: Unlike traditional Decision Trees, which search for the best split, ExtraTrees use a random subset of features and possible splits at each node, creating an ensemble of highly diverse trees that are combined to make predictions.

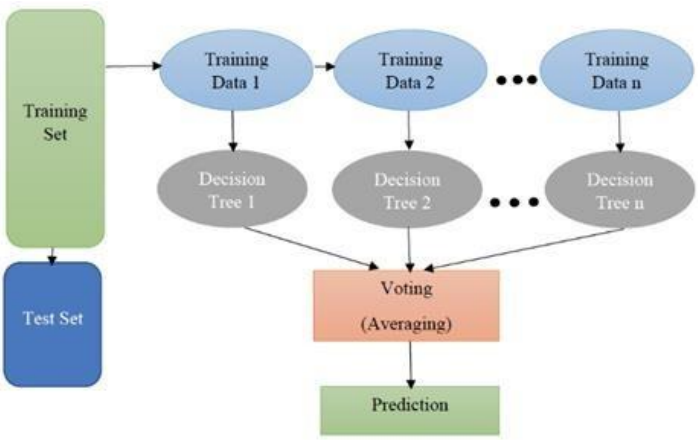


Fig. 7 Decision Tree classifier

Naive Bayes (NB) Classifier:

- Probabilistic Model: The Naive Bayes classifier is based on Bayes' Theorem and assumes that the features are conditionally independent given the class label. This assumption simplifies the computation of class probabilities by multiplying the individual probabilities of each feature (Toshkhujaev et. all, 2020).

- Model Structure: It uses the formula posterior probability of class given the features is the likelihood of features given the class.

Voting Classifier:

- Ensemble Method: The Voting Classifier combines the predictions from multiple base classifiers to make a final prediction. It aggregates the outputs of the individual models, often using a majority voting mechanism for classification or averaging for regression.

- Types of Voting: There are typically two types of voting: hard voting, where the class with the majority vote is selected, and soft voting, where the class probabilities from each classifier are averaged to determine the final prediction (Roy, S., & Chandra, A. 2022).

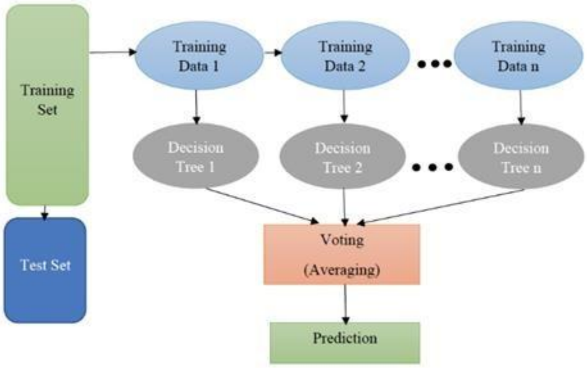


Fig. 8. Hard Voting Classifier.

Linear Discriminant Analysis (LDA):

- Architecture: LDA seeks to find a linear combination of features that best separates two or more classes. It calculates a linear

decision boundary by maximizing the ratio of between-class variance to within-class variance. This is done by projecting data onto a lower-dimensional space where the classes are maximally separated (Lin, W., Gao. et. all, 2021)

- Model Structure: LDA assumes that the features are normally distributed with the same covariance matrix for all classes. The resulting linear discriminant function is used to classify new data points based on their projection onto this space.

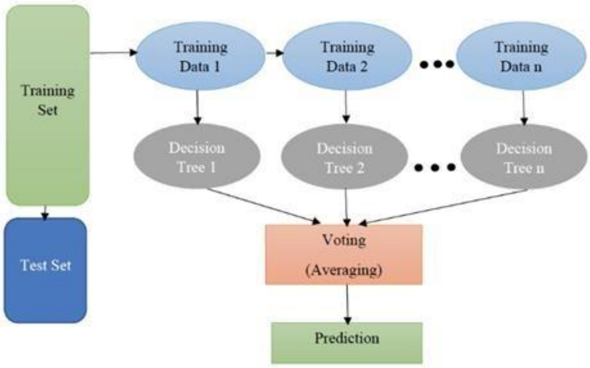


Fig. 9. LDA Classifier

Quadratic Discriminant Analysis (QDA):

- Architecture: QDA extends LDA by allowing each class to have its own covariance matrix, making it capable of capturing more complex boundaries between classes. It fits a quadratic decision boundary by modeling the likelihood of each class as a multivariate normal distribution with its own covariance matrix.

- Model Structure: QDA computes the posterior probabilities using Bayes' Theorem with class-specific covariance matrices. The decision boundary is therefore quadratic rather than linear, which can better handle situations where the classes are not linearly separable.

K-Nearest Neighbors (KNN):

- Architecture: KNN is a non-parametric, instance-based learning algorithm that classifies new data points based on the majority class among their K nearest neighbors in the feature space. It uses distance metrics (e.g., Euclidean, Manhattan) to determine the proximity of data points.

- Model Structure: The algorithm stores all training data and, during prediction, identifies the K closest training examples to a given test point. The most frequent class among these neighbors is assigned to the test point.

Stochastic Gradient Descent (SGD) Classifier:

- Architecture: SGD is an optimization algorithm used for training various machine learning models, including linear classifiers. It updates model parameters iteratively using a small, randomly selected subset of the data (mini-batch) to compute the gradient of the loss function.

- Model Structure: The classifier's parameters are updated in the direction that reduces the loss function, based on the gradient computed from the mini-batch. This process continues until convergence or for a set number of iterations, allowing the model to handle large datasets and adapt to complex patterns.

4. Experimental Results

4.1 Model Performance

To assess the effectiveness of the machine learning models, several performance metrics were computed: Precision, Recall, F1-score, and Accuracy, all based on the training data. Additionally, the study included the AUC-ROC curve for each model to evaluate their discriminative ability. The performance of each model was further analyzed using a confusion matrix, which comprises four key attributes: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

4.2 Classification Metrics

As shown in the table, we applied various machine learning algorithms to predict dementia outcomes using longitudinal MRI data. Our primary focus was on utilizing ML techniques to accurately forecast dementia status.

Precision: Precision measures the accuracy of positive predictions, indicating how often the model correctly identifies true positive cases (TP) while minimizing false positives (FP). It is calculated as:

$$\text{Precision} = TP/(TP+FP)$$

Recall: Also known as sensitivity or the true positive rate, recall quantifies the model's ability to identify all positive instances within the dataset. It is calculated by dividing the true positives by the sum of true positives and false negatives (FN):

$$\text{Recall} = TP/(TP+FN)$$

F1 Score: The F1 Score is the harmonic mean of precision and recall, providing a balance between the two. It evaluates the accuracy of the testing process, with a range between 0 and 1:

$$\text{F1 Score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision}*\text{Recall})$$

In this study, multiple ML classifiers were evaluated based on these metrics. Among them, the ExtraTree classifier outperformed all others, achieving the highest precision accuracy of 91%. This indicates that the ExtraTree classifier was particularly effective at correctly identifying positive dementia cases while minimizing false positives.

Table 2 Performance result of binary classification

Sr. No	ML Models	Model	Classification	Precision (%)	Recall (%)	f1-score (%)	Support (%)	Accuracy (%)
1.	Extra Tree Classifier	ET	Nondemented(0)	84	90	87	52	87
			Demented(1)	91	85	88	60	
2.	Hard voting (Voting Classifier)	HV	Nondemented(0)	82	87	84	52	85
			Demented(1)	88	83	85	60	
3.	Soft voting (Voting Classifier)	SV	Nondemented(0)	81	85	83	52	84
			Demented(1)	86	83	85	60	
4.	XGBOOST	XGB	Nondemented(0)	81	85	83	52	84
			Demented(1)	86	83	85	60	
5.	Random Forest Classifier (RF)	RF	Nondemented(0)	80	87	83	52	83
			Demented(1)	88	82	84	60	

6.	AdaBoost classifier	ADB	Nondemented(0)	78	87	82	52	82
			Demented(1)	87	78	82	60	
7.	Decision Tree Classifier (DT)	DT	Nondemented(0)	75	83	79	52	79
			Demented(1)	84	77	80	60	
8.	Quadratic Discriminant Analysis (QDA)	QDA	Nondemented(0)	72	90	80	52	79
			Demented(1)	89	70	79	60	
9.	Histogram-based Gradient Boosting Classifier (Ensemble Method)	Hist-GBC	Nondemented(0)	75	85	79	52	79
			Demented(1)	85	75	80	60	
10.	Multilayer Perceptron	MLP	Nondemented(0)	82	69	75	52	78
			Demented(1)	76	87	81	60	
11.	Support Vector Machine (SVM)	SVM	Nondemented(0)	71	88	79	52	78
			Demented(1)	87	68	77	60	
12.	Gradient Boosting Classifier (Ensemble Method)	GBC	Nondemented(0)	76	79	77	52	78
			Demented(1)	81	78	80	60	
13.	Linear Discriminant Analysis (LDA)	LDA	Nondemented(0)	68	88	77	52	75
			Demented(1)	86	63	73	60	
14.	Logistic Regression (LR)	LR	Nondemented(0)	69	77	73	52	73
			Demented(1)	78	70	74	60	
15.	GaussianNB (Naive_bayes)	GNB	Nondemented(0)	67	85	75	52	73
			Demented(1)	83	63	72	60	
16.	KNeighbors Classifier	KNN	Nondemented(0)	62	75	68	52	67
			Demented(1)	73	60	66	60	
17.	SGDClassifier	SGD	Nondemented(0)	56	98	71	52	63
			Demented(1)	95	33	49	60	
18.	Multinomial NB (Naive Bayes)	MNB	Nondemented(0)	58	67	63	52	62
			Demented(1)	67	58	63	60	
19.	Bagging Classifier (Ensemble Method)	Bagging	Nondemented(0)	56	69	62	52	61
			Demented(1)	67	53	59	60	
20.	BernoulliNB (Naive_bayes)	BNB	Nondemented(0)	52	65	58	52	56
			Demented(1)	62	48	54	60	

Fig. 10. Classification Metrics Performance of (a) Multilayer Perceptron (b) Logistic Regression (c) Random Forest (d) Support Vector Machine (e) Decision Tree (f) ExtraTree (g) GaussianNB (h) BernoulliNB (i) MultinomialNB (j) XGBoost (k) AdaBoost (l) Linear Discriminant Analysis (m) Quadratic Discriminant Analysis (n) Hard Voting (o) Soft Voting (p) Bagging (q) Gradient Boosting Classifier (r) Histogram-based gradient boosting (s) KNeighbors (t) Stochastic Gradient Descent classifiers

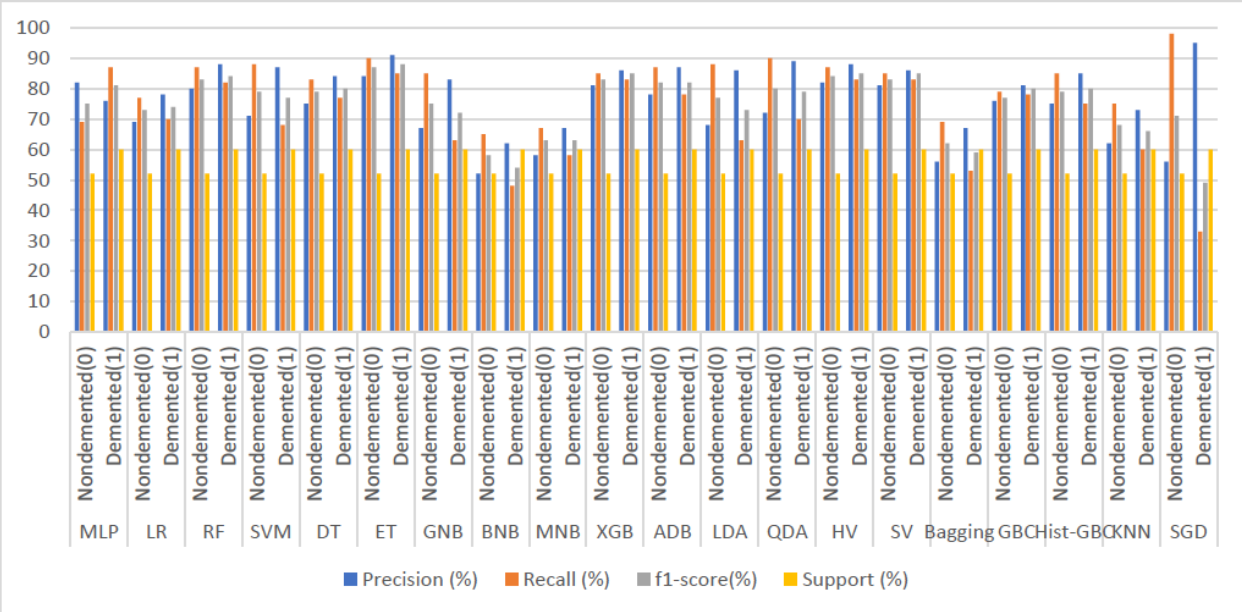


Fig. 11. Comparison of classification metrics of different ML algorithms.

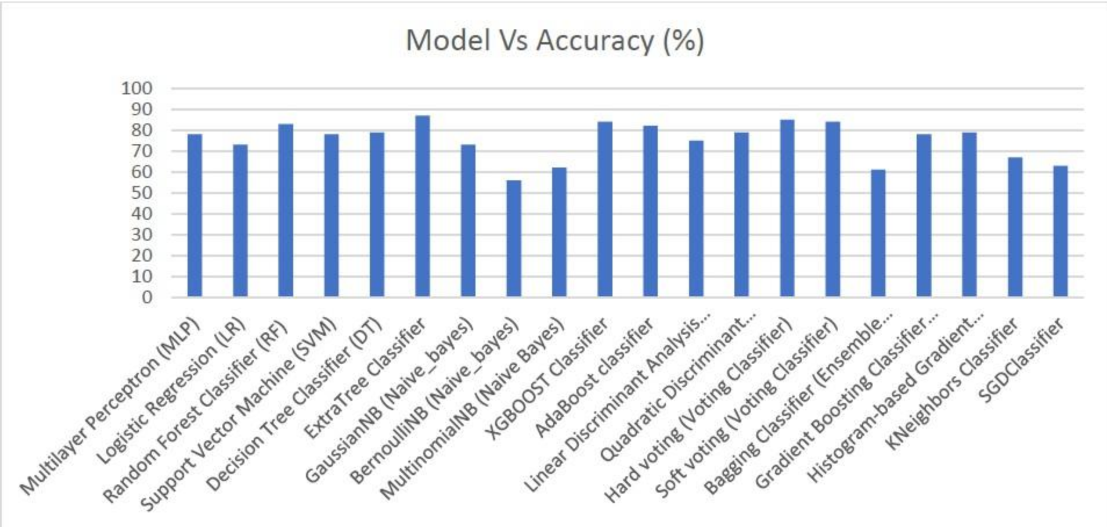


Fig. 12. Accuracy calculated on OASIS dataset

4.3. Confusion Matrix

A confusion matrix is an N × N matrix that represents the performance of a classification model, detailing the outcomes of predictions by comparing them to the actual values. It is composed of four key components: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These components provide insights into the model's accuracy and error distribution. In the context of classification, the confusion matrix is used to record and visualize the results, as illustrated in Fig. 20.

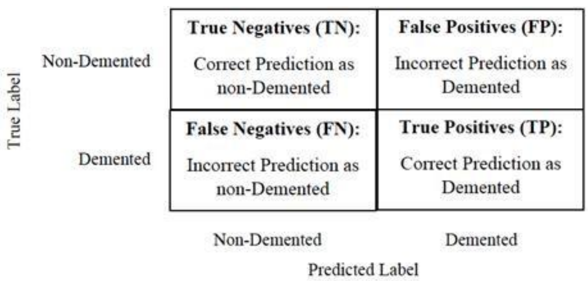
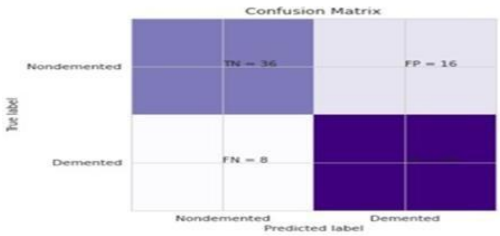
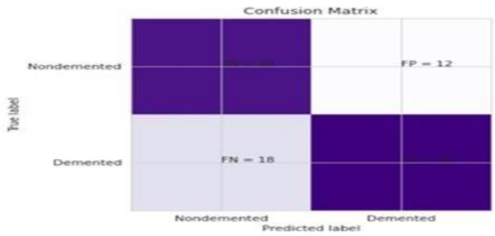


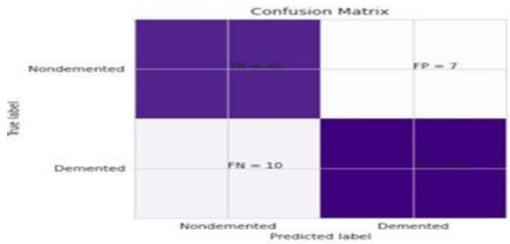
Fig. 13. Confusion matrix distribution describing the performance of a classification model



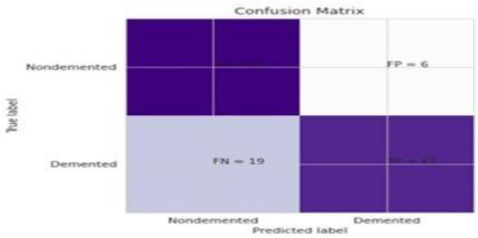
(a) Multilayer Perceptron



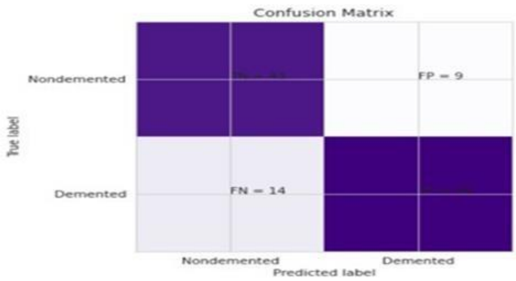
(b) Logistic Regression



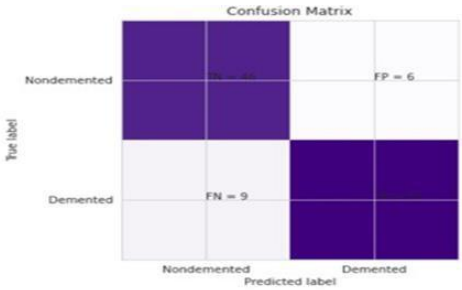
(c) Random Forest



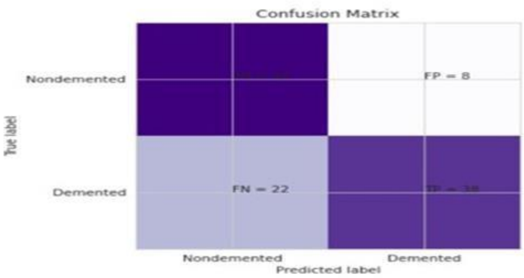
(d) Support Vector Machine



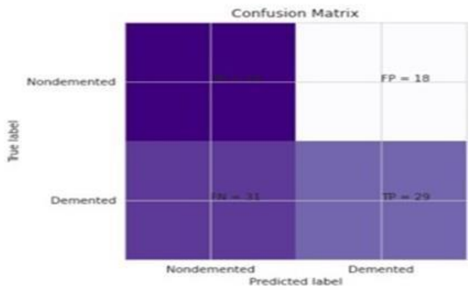
(e) Decision Tree



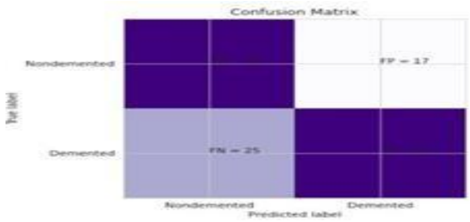
(f) ExtraTree



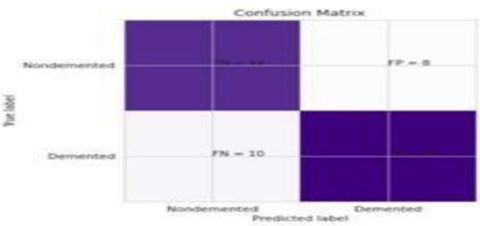
(g) GaussianNB



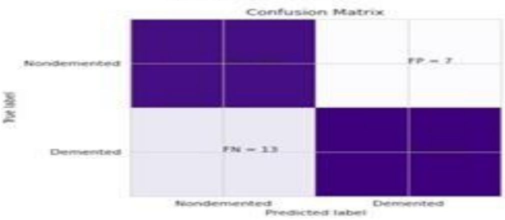
(h) BernoulliNB



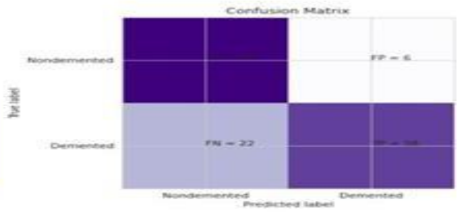
(i) MultinoimialNB



(j) XGBoost



(k) AdaBoost



(l) Linear Discriminant Analysis

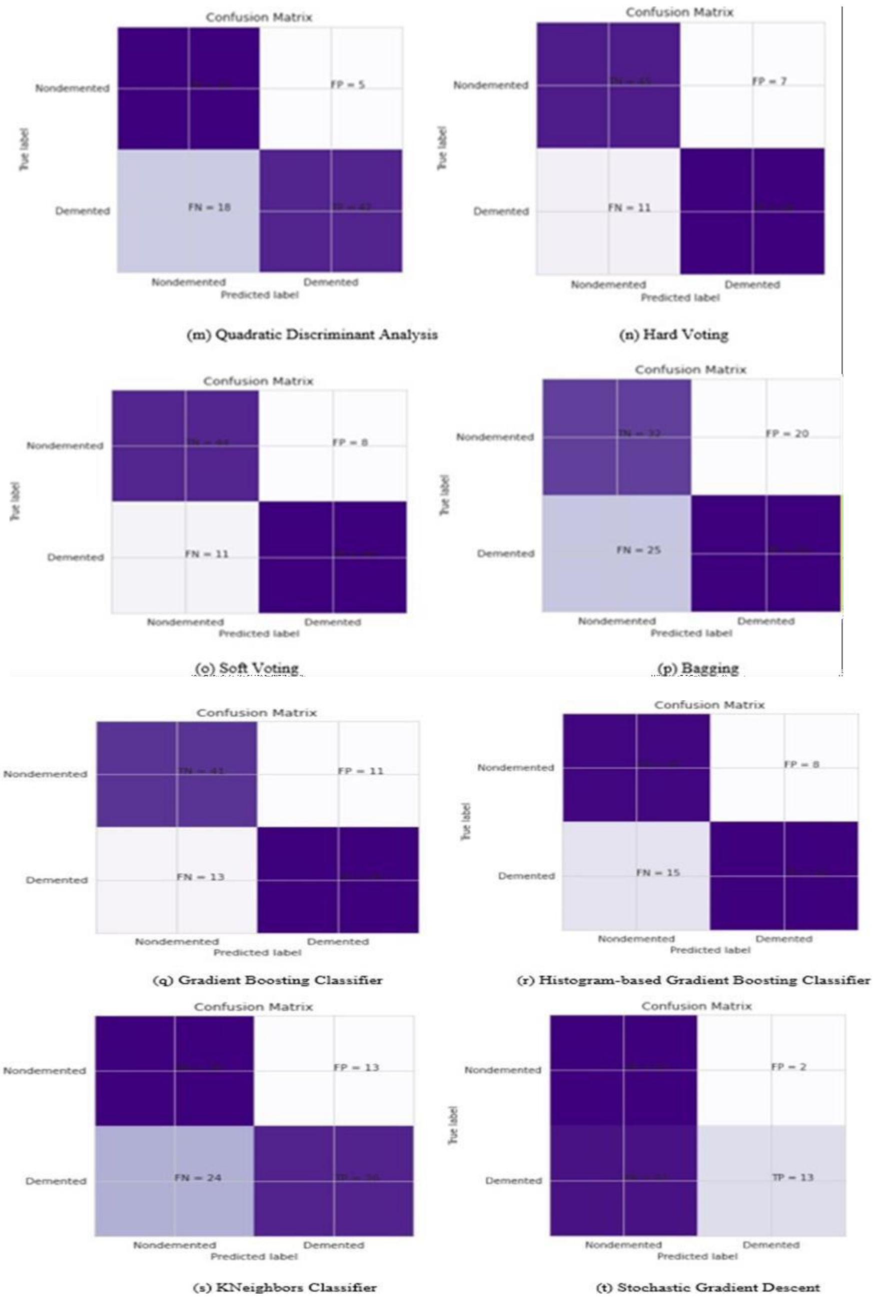
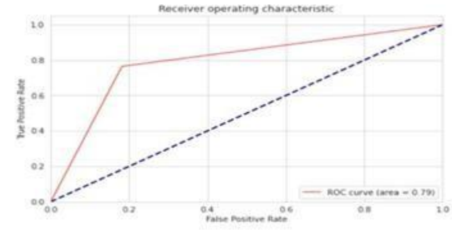


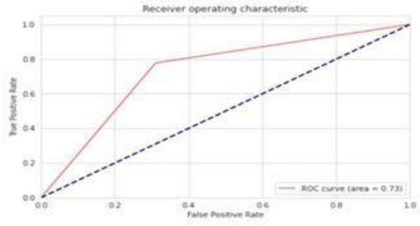
Fig. 14. Confusion matrix Performance of (a) Multilayer Perceptron (b) Logistic Regression (c) Random Forest (d) Support Vector Machine (e) Decision Tree (f) ExtraTree (g) GaussianNB (h) BernoulliNB (i) MultinomialNB (j) XGBoost (k) AdaBoost (l) Linear Discriminant Analysis (m) Quadratic Discriminant Analysis (n) Hard Voting (o) Soft Voting (p) Bagging (q) Gradient Boosting Classifier (r) Histogram-based gradient boosting (s) KNeighbors (t) Stochastic Gradient Descent classifiers

4.4. Receiver Operator Characteristic (ROC) Curve

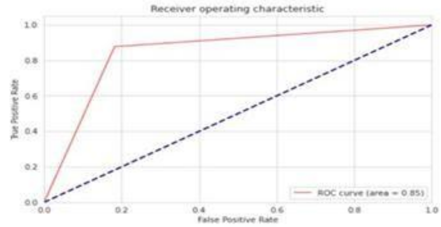
The Receiver Operator Characteristic (ROC) curve is a graphical representation that illustrates the diagnostic ability of a binary classifier system. It plots the True Positive Rate (TPR), also known as sensitivity, against the False Positive Rate (FPR), also known as 1-specificity, at various threshold settings. Each point on the ROC curve corresponds to a different sensitivity-specificity pair, representing how well the model distinguishes between classes.



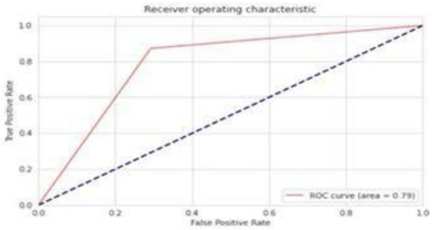
(a) Multilayer Perceptron



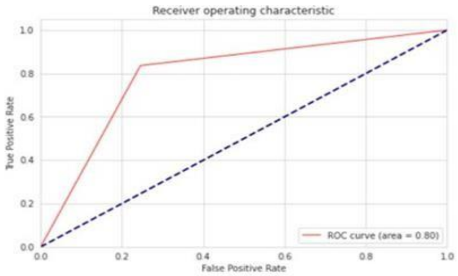
(b) Logistic Regression



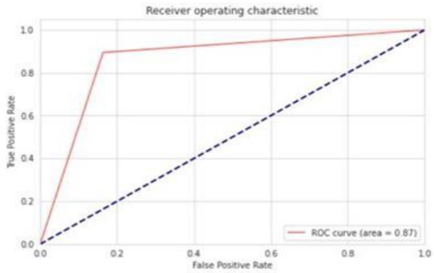
(c) Random Forest



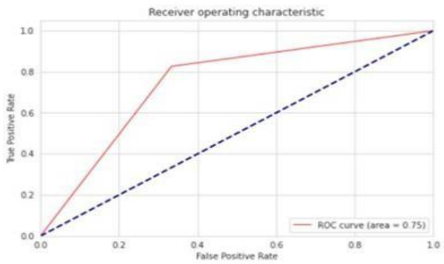
(d) Support Vector Machine



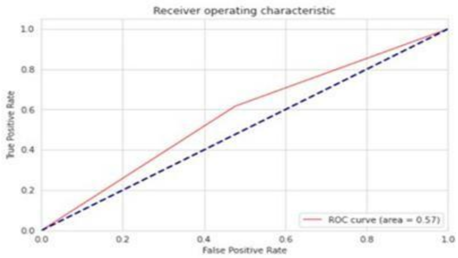
(e) Decision Tree



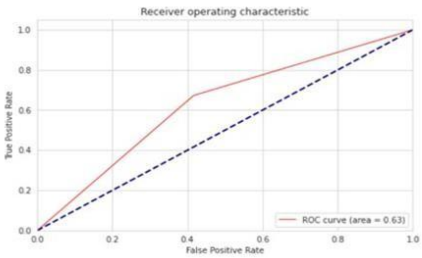
(f) ExtraTree



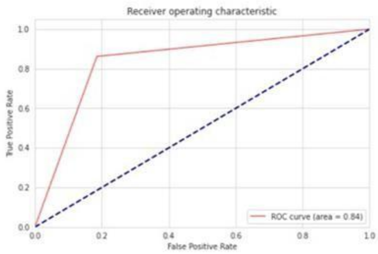
(g) GaussianNB



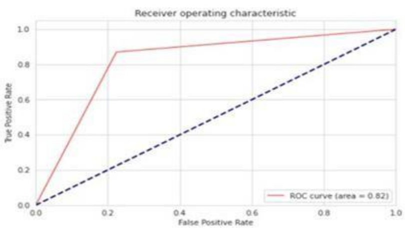
(h) BernoulliNB



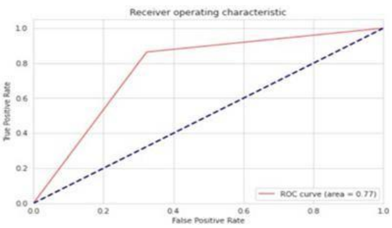
(i) MultinoimialNB



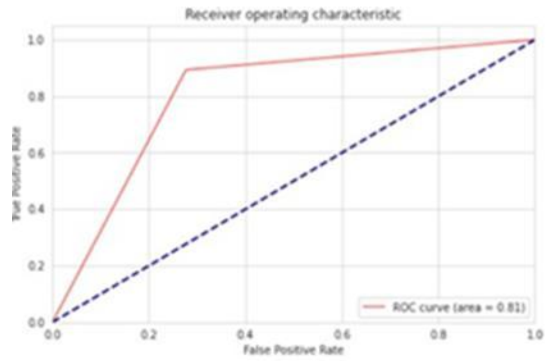
(j) XGBoost



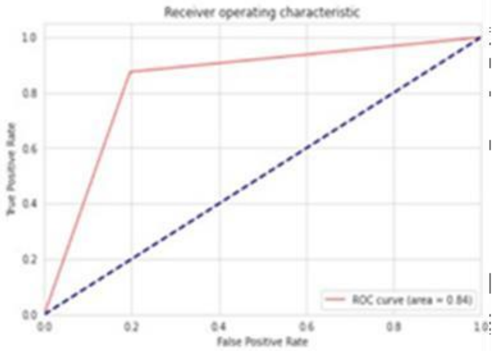
(k) AdaBoost



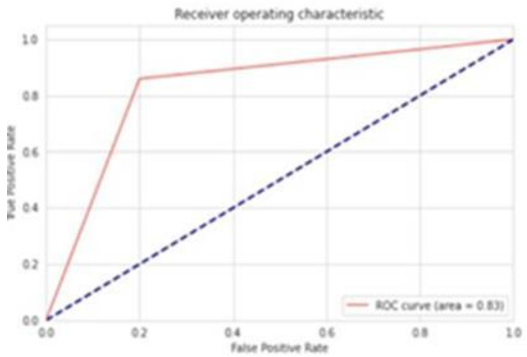
(l) Linear Discriminant Analysis



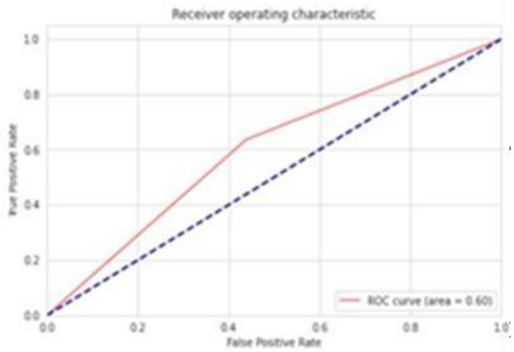
(m) Quadratic Discriminant Analysis



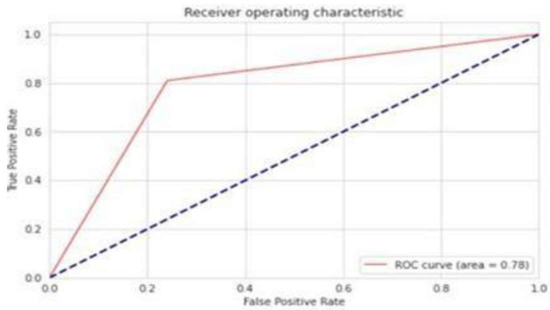
(n) Hard Voting



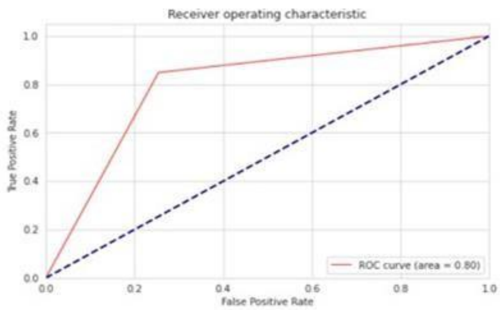
(o) Soft Voting



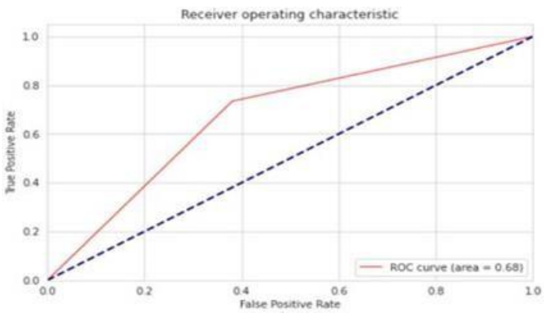
(p) Bagging



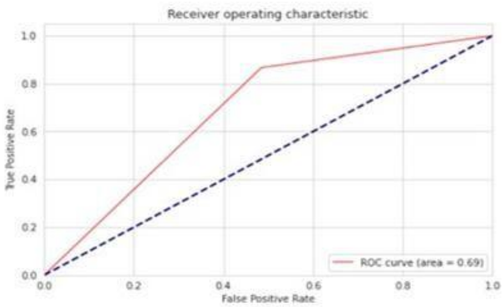
(q) Gradient Boosting Classifier



(r) Histogram-based Gradient Boosting Classifier



(s) KNeighbors Classifier



(t) Stochastic Gradient Descent

Fig. 15. ROC curve of (a) Multilayer Perceptron (b) Logistic Regression (c) Random Forest (d) Support Vector Machine (e) Decision Tree (f) ExtraTree (g) GaussianNB (h) BernoulliNB (i) MultinomialNB (j) XGBoost (k) AdaBoost (l) Linear Discriminant Analysis (m) Quadratic Discriminant Analysis (n) Hard Voting (o) Soft Voting (p) Bagging (q) Gradient Boosting Classifier (r) Histogram-based gradient boosting (s) KNeighbors (t) Stochastic Gradient Descent classifiers

The ROC curve is particularly useful for assessing the overall performance of the model in distinguishing between positive (demented) and negative (non-demented) cases. The closer the ROC curve approaches the upper left corner of the plot, the higher the model's accuracy. This proximity indicates a better trade-off between sensitivity and specificity, reflecting the model's effectiveness in accurately classifying Alzheimer's disease (AD) cases.

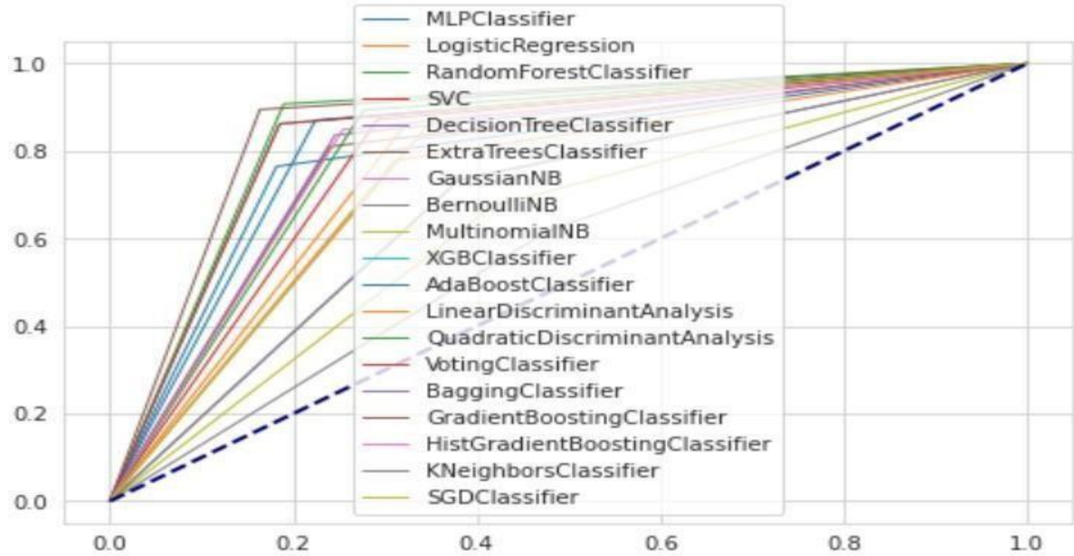


Fig. 16. Comparison of ROC curve of different ML classifiers

5. Conclusions

This study presents a straightforward approach to detecting Alzheimer's Disease (AD) by classifying longitudinal brain MRI data into two categories: Demented and Non-demented, using supervised learning techniques. Given that Alzheimer's disease currently has no cure, early detection and accurate diagnosis are crucial for reducing risk and managing symptoms effectively. By analyzing datasets of medical records using machine learning algorithms, it is possible to identify AD more swiftly, which can assist clinicians in taking proactive steps to support patients.

The application of prediction algorithms simplifies the process of determining the stage of AD, allowing for timely intervention and potentially improving patient outcomes. Should a treatment for Alzheimer's be discovered in the near future, these predictive models could play a vital role in determining the disease's progression and the potential benefits of early treatment. Our literature review indicates that various machine learning algorithms have been employed in the effort to detect Alzheimer's disease. By utilizing measures such as MMSE, CDR, ASF, and Education, we were able to train our model to differentiate between individuals with and without Alzheimer's disease. The primary goal of this study was to leverage machine learning techniques to predict AD classification outcomes with greater accuracy and enhanced performance.

References

[1] Goyal, P., Rani, R., & Singh, K. (2022). State-of-the-art machine learning techniques for diagnosis of Alzheimer's disease from MR-images: A systematic review. *Archives of Computational Methods in Engineering*, *29*(5), 2737-2780.

[2] Arafa, D. A., Moustafa, H. E. D., Ali-Eldin, A. M., & Ali, H. A. (2022). Early detection of Alzheimer's disease based on the state-of-the-art deep learning approach: a comprehensive survey. *Multimedia Tools and Applications*, *81*(17), 23735-23776.

[3] Tang, X., & Liu, J. (2021). Comparing different algorithms for the course of Alzheimer's disease using machine learning. *Annals of Palliative Medicine*, *10*(9), 9715724-9719724.

[4] Jo, T., Nho, K., & Saykin, A. J. (2019). Deep learning in Alzheimer's disease: diagnostic classification and prognostic prediction using neuroimaging data. *Frontiers in aging neuroscience*, *11*, 220.

[5] Ramzan, F., Khan, M. U. G., Rehmat, A., Iqbal, S., Saba, T., Rehman, A., & Mehmood, Z. (2020). A deep learning approach for automated diagnosis and multi-class classification of Alzheimer's disease stages using resting-state fMRI and residual neural networks. *Journal of medical systems*, *44*, 1-16.

[6] Kavitha, C., Mani, V., Srividhya, S. R., Khalaf, O. I., & Tavera Romero, C. A. (2022). Early-stage Alzheimer's disease prediction using machine learning models. *Frontiers in public health*, *10*, 853294.

[7] Battineni, G., Hossain, M. A., Chintalapudi, N., Traini, E., Dhulipalla, V. R., Ramasamy, M., & Amenta, F. (2021). Improved Alzheimer's disease detection by MRI using multimodal machine learning algorithms. *Diagnostics*, *11*(11), 2103.

[8] Khan, A., & Zubair, S. (2020). Longitudinal magnetic resonance imaging as a potential correlate in the diagnosis of Alzheimer disease: exploratory

data analysis. *JMIR Biomedical Engineering*, *5*(1), e14389.

[9] Tripathi, S., Augustin, A., & Kim, E. (2023). Longitudinal Neuroimaging Data Classification for Early Detection of Alzheimer's Disease using Ensemble Learning Models. *Authorea Preprints*.

[10] Zhang, Y. D., Wang, S., & Dong, Z. (2014). Classification of Alzheimer disease based on structural magnetic resonance imaging by kernel support vector machine decision tree. *Progress In Electromagnetics Research*, *144*, 171-184.

[11] Gopagoni, D., Lakshmi, P. V., & Ananthula, R. S. (2020). An application of machine learning strategies to predict alzheimer's illness progression in patients. *International Journal of Advanced Research in Engineering and Technology*, *11*(6).

[12] Kishore, P., Kumari, C. U., Kumar, M. N. V. S. S., & Pavani, T. (2021). Detection and analysis of Alzheimer's disease using various machine learning algorithms. *Materials today: proceedings*, *45*, 1502-1508.

[13] Khan, A., & Zubair, S. (2020, November). A Machine Learning-based robust approach to identify Dementia progression employing Dimensionality Reduction in Cross-Sectional MRI data. In *2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH)* (pp. 237-242). IEEE.

[14] López, M., Ramírez, J., Górriz, J. M., Álvarez, I., Salas-Gonzalez, D., Segovia, F., ... & Alzheimer's Disease Neuroimaging Initiative. (2011). Principal component analysis-based techniques and supervised classification schemes for the early detection of Alzheimer's disease. *Neurocomputing*, *74*(8), 1260-1271.

[15] Joshi, S., Simha, V., Shenoy, D., Venugopal, K. R., & Patnaik, L. M. (2010). Classification and treatment of different stages of alzheimer's disease using various machine learning methods. *International Journal of Bioinformatics Research*, *2*(1), 44-52.

[16] Jyotiyana, M., & Kesswani, N. (2020). Classification and prediction of Alzheimer's disease using multi-layer perceptron. *International Journal of Reasoning-based Intelligent Systems*, *12*(4), 255-263.

[17] Fukunishi, H., Nishiyama, M., Luo, Y., Kubo, M., & Kobayashi, Y. (2020). Alzheimer-type dementia prediction by sparse logistic regression using claim data. *Computer Methods and Programs in Biomedicine*, *196*, 105582.

[18] Ali, M. S., Islam, M. K., Haque, J., Das, A. A., Duranta, D. S., & Islam, M. A. (2021, April). Alzheimer's disease detection using m-random forest algorithm with optimum features extraction. In *2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA)* (pp. 1-6). IEEE.

[19] Logan, R., Williams, B. G., Ferreira da Silva, M., Indani, A., Schcolnicov, N., Ganguly, A., & Miller, S. J. (2021). Deep convolutional neural networks with ensemble learning and generative adversarial networks for Alzheimer's disease image data classification. *Frontiers in aging neuroscience*, *13*, 720226.

[20] Buyrukoğlu, S. (2021, September). Improvement of machine learning models' performances based on ensemble learning for the detection of Alzheimer disease. In *2021 6th International Conference on Computer Science and Engineering (UBMK)* (pp. 102-106). IEEE.

[21] Sharma, A., Kaur, S., Memon, N., Fathima, A. J., Ray, S., & Bhatt, M. W. (2021). Alzheimer's patients detection using support vector machine (SVM) with quantitative analysis. *Neuroscience Informatics*, *1*(3), 100012.

[22] Saputra, R. A., Agustina, C., Puspitasari, D., Ramanda, R., Pribadi, D., & Indriani, K. (2020, November). Detecting Alzheimer's disease by the decision tree methods based on particle swarm optimization. In *Journal of Physics: Conference Series* (Vol. 1641, No. 1, p. 012025). IOP Publishing.

[23] Toshkhujaev, S., Lee, K. H., Choi, K. Y., Lee, J. J., Kwon, G. R., Gupta, Y., & Lama, R. K. (2020). Classification of Alzheimer's disease and mild cognitive impairment based on cortical and subcortical features from MRI T1 brain images utilizing four different types of datasets. *Journal of Healthcare Engineering*, *2020*(1), 3743171.

[24] Roy, S., & Chandra, A. (2022). On the detection of Alzheimer's disease using fuzzy logic based majority voter classifier. *Multimedia Tools and Applications*, *81*(30), 43145-43161.

[25] Lin, W., Gao, Q., Du, M., Chen, W., & Tong, T. (2021). Multiclass diagnosis of stages of Alzheimer's disease using linear discriminant analysis scoring for multimodal data. *Computers in biology and medicine*, *134*, 104478.