FMDB *Transactions on Sustainable Health Science Letters*

Exploring the Efficacy of LightGBM, AdaBoost, and CatBoost in Alzheimer's Disease Classification

G. Hariprasath[1, *], P. Dhinakaran[2], V. Harish Kumar[3]

[1,2,3]Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.
hg8694@srmist.edu.in[1], pp4417@srmist.edu.in[2], hk0837@srmist.edu.in[3]

Abstract: Alzheimer's disease (AD) is a major public health concern that requires prompt and accurate diagnosis to intervene effectively. We present a comprehensive machine learning system in our work that is specifically designed to classify AD. It incorporates various neuroimaging and clinical features, including air time1, disp index1, gmrt in air1, max x extension1, and max y extension1. We can decipher complex data correlations suggesting AD pathology using rigorous preprocessing and visualization methods, such as correlation heatmaps and 3D scatter plots. We can distinguish minute changes between AD, moderate cognitive impairment, and healthy controls using CatBoost, LightGBM, and AdaBoost classifiers. A thorough assessment of the model's performance is provided by the rigorous evaluation metrics of accuracy, precision, recall, and F1-score, which are supplemented by detailed classification reports and confusion matrices. Learning curves also provide information about the generalization and flexibility of models. Our findings support integrated analysis approaches across many data modalities and highlight the revolutionary potential of machine learning in AD diagnosis. Developing customized treatment plans and cutting-edge clinical decision support systems is expected to improve patient outcomes and the standard of care for neurodegenerative diseases.

Keywords: Alzheimer's Disease; Machine Learning; Clinical Features; CatBoost and LightGBM; AdaBoost Classification; Evaluation Metrics; Personalized Treatment; Scatter and Bar Plot.

Received on: 19/12/2023, **Revised on:** 26/01/2024, **Accepted on:** 25/02/2024, **Published on:** 05/03/2024

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSHSL

DOI: https://doi.org/10.69888/FTSHSL.2024.000171

Cite as: G. Hariprasath, P. Dhinakaran, and V. H. Kumar, "Exploring the Efficacy of LightGBM, AdaBoost, and CatBoost in Alzheimer's Disease Classification," *FMDB Transactions on Sustainable Health Science Letters*., vol. 2, no.1, pp. 41–55, 2024.

1. Introduction

Alzheimer's disease is a degenerative neurological condition affecting the brain, resulting in cognitive decline and memory loss. It is the most prevalent kind of dementia, accounting for between 60 and 70% of all cases. Alzheimer's disease gradually weakens memory, reasoning, and behaviour, eventually affecting the individual's capacity to carry out everyday chores. Language, decision-making, and personality can be impaired as the condition advances. Alzheimer's disease is characterized by the buildup of aberrant protein deposits in the brain, such as beta-amyloid plaques and tau tangles, which cause brain cell death and neuronal connection breakage. The specific etiology of Alzheimer's disease is unknown, although it is thought to be a complex interaction of hereditary, environmental, and lifestyle factors. There is currently no cure for Alzheimer's disease, and present therapies mostly focus on symptom management and delaying disease progression. Given the growing incidence of Alzheimer's disease, particularly among the elderly, researchers are working to understand its underlying causes better and

[*] Corresponding author.

create viable treatments. The precise etiology of Alzheimer's disease is unknown; however, it is thought to be impacted by a mix of genetic, environmental, and lifestyle factors. In this paper, we want to use machine learning to tackle the complexity of Alzheimer's disease, a severe neurological disorder that affects millions of people worldwide. Our objective is to create prediction models that can help with the early detection and management of Alzheimer's disease, ultimately leading to better patient outcomes and quality of life. To achieve this goal, we used three different machine learning models: [CatBoost], [LightGBM], and [AdaBoost]. Each of these models has distinct strengths and capabilities, allowing us to test several approaches to Alzheimer's disease prediction and categorization. The study's findings might substantially influence Alzheimer's disease research and therapeutic practice.

By discovering precise and reliable indicators of Alzheimer's disease start and development, we can enable healthcare practitioners to intervene early and execute tailored therapies, eventually improving patient outcomes and advancing our understanding of this dreadful illness. Our comprehensive examination and comparison of machine learning models aims to provide useful insights that will pave the way for more effective diagnostic and treatment techniques in the battle against Alzheimer's disease. Throughout this study, we will compare and analyze the performance of these three models using a variety of measures, including accuracy, precision, recall, and F1-score. We hope to determine the most effective technique for classifying Alzheimer's disease and differentiating it from other cognitive diseases or healthy states by thoroughly analyzing these models.

In our efforts to treat Alzheimer's disease, we are using machine learning (ML) models to categorize this terrible neurological disorder. Alzheimer's disease presents considerable hurdles owing to its progressive nature and the intricacy of the underlying processes. Early detection and precise diagnosis are critical to effective treatment and intervention. We hope to construct strong models to differentiate between those with Alzheimer's disease and those without. We begin the training phase, in which we use a variety of machine learning algorithms specialized to classify Alzheimer's disease. We use ensemble algorithms like AdaBoost, which iteratively combine weak learners to create a powerful classifier. In addition, we use gradient boosting methods like CatBoost and LightGBM, which are optimized for effectively handling category features and huge datasets. Our ultimate objective is to learn more about the underlying patterns and biomarkers linked with Alzheimer's disease and develop accurate classifiers.

By identifying these aspects, we want to contribute to a more comprehensive scientific knowledge of the disease and open the path for more focused therapies and personalized treatment plans. We want to achieve significant progress in the battle against Alzheimer's disease by using an interdisciplinary strategy that combines machine learning, neurobiology, and clinical knowledge. Our research holds the prospect of earlier identification, better patient outcomes, and, eventually, a world in which Alzheimer's disease is no longer a threat to individuals and families. Machine learning (ML) is critical in categorizing Alzheimer's disease because it uses several data sources to discover illness-related patterns and signs. ML is a potent tool for classifying Alzheimer's disease, allowing for integrating diverse data sources, discovering predictive indicators, and building reliable diagnostic and prognostic models to enhance patient treatment and outcomes. ML models need high-quality data to learn from. Datasets for Alzheimer's disease categorization often comprise demographic information, medical history, cognitive evaluations, genetic markers, and neuroimaging results. Preprocessing procedures include resolving missing data, encoding categorical variables, and standardizing or normalizing numerical characteristics to guarantee consistency and compatibility with ML algorithms. Machine learning models use feature selection and engineering approaches to discover the most important Alzheimer's disease predictions. This method includes determining the relevance of various characteristics, lowering dimensionality, and developing new features to represent complicated connections within the data.

For example, MRI images may generate neuroimaging parameters such as hippocampal volume or cortical thickness, which can give insights into brain structural alterations linked with Alzheimer's. Various ML algorithms are used for Alzheimer's disease categorization. The data type, interpretability requirements, processing resources, and performance measures all influence the method selection. Models are trained on labelled datasets with the presence or absence of Alzheimer's disease as the goal variable. Trained models are assessed using performance indicators such as accuracy, precision, recall, and F1-score. These metrics evaluate the model's ability to accurately categorize Alzheimer's patients and separate them from healthy controls or those with other cognitive disorders. ML models give information on the variables and biomarkers related to Alzheimer's disease, assisting doctors in early detection and diagnosis. Predictive models can identify people who are at risk of developing Alzheimer's disease, enabling early intervention and personalized treatment plans. Furthermore, ML approaches help uncover novel biomarkers and therapeutic targets by analyzing big datasets and integrating multimodal data sources.

Early detection of Alzheimer's disease allows for more timely treatments and treatment techniques. While there is no cure for Alzheimer's disease, early identification allows healthcare practitioners to introduce therapies that delay disease development, manage symptoms, and improve patients' quality of life. Early detection can improve patient outcomes by allowing people with Alzheimer's disease to obtain necessary care and support services sooner. This can help postpone the onset of severe cognitive deficits and functional decline, allowing patients to preserve their independence and quality of life for longer.

We compared the performance of three machine learning models for Alzheimer’s disease classification, AdaBoost, CatBoost, and LightGBM, utilizing a variety of measures and factors. AdaBoost, famed for its ensemble technique, provides high interpretability by progressively merging weak learners to produce a strong classifier, albeit pretreatment steps may be required to handle categorical information successfully. CatBoost, optimized for categorical variables, has shown quick training and resistance to overfitting and built-in techniques for dealing with big datasets and parallel processing. Similarly, LightGBM demonstrated rapid training durations, good handling of huge datasets, and choices for feature significance analysis, although it may have less interpretability than AdaBoost. This project presents a comprehensive evaluation of machine learning models for Alzheimer’s disease categorization, shedding light on their performance and applicability for the job at hand. Such comparisons are critical for determining the best model and directing future research and clinical applications in Alzheimer’s disease diagnosis and therapy.

2. Objective

- Optimize model performance by tuning parameters for each algorithm (LightGBM, AdaBoost, and CatBoost) to achieve high accuracy, precision, recall, and F1 score.
- Enhance personalized treatment planning by classifying individuals accurately and supporting the development of tailored care strategies.
- Improve overall understanding of Alzheimer’s disease by leveraging machine learning models to identify and analyze patterns in cognitive health data.
- Provide interpretability and transparency to offer insights into the features contributing most to classification decisions, aiding clinical decision-making.
- The goal is to demonstrate how machine learning may revolutionize AD diagnosis, promote integrated analysis methodologies, and open the door for creating cutting-edge clinical decision support systems and individualized treatment plans.

3. Literature review

Basher et al. [1] discuss the study of using Convolutional Neural Networks (CNN) and Deep Neural Networks (DNN) for Alzheimer’s Disease (AD) diagnosis from Structural Magnetic Resonance Imaging (SMRI) data. The study focuses on volumetric features extracted from the hippocampus to classify AD versus normal control (NC) subjects. The results show high accuracy in AD classification, outperforming other methods. The approach is automatically efficient and yields superior precision, recall, and F1 scores compared to alternative techniques. The study emphasizes the importance of accurate diagnosis for AD and highlights the potential of CNN and DNN models in improving diagnostic outcomes.

Qu et al. [2] study focuses on utilizing a Graph Convolutional Network (GCN) based on Univariate Neurodegeneration Biomarkers (UNBs) for the early diagnosis of Alzheimer’s disease. The research aims to extract hidden information from UNBs using an attention module within the GCN framework. By incorporating phenotypic measures like age, gender, and genetic information, the model achieves improved classification accuracy. The results demonstrate the effectiveness of the proposed method in characterizing the influence of Alzheimer’s disease on individual morphology, with potential implications for enhancing early detection and understanding of neurodegenerative diseases.

Eke et al. [3] discuss the use of machine learning techniques, particularly support vector machines, in the early detection of Alzheimer’s Disease through the analysis of blood plasma proteins. It highlights the limitations of amyloid-based biomarkers and the importance of exploring non-amyloid biomarkers for early detection. The study emphasizes further research to identify the most important protein panels for accurate diagnosis.

Martinez-Murcia et al. [4] discuss a study that utilizes deep convolutional autoencoders to analyze MRI images in Alzheimer’s Disease research. The main contributions include directly applying deep convolutional feature representation to MRI images without prior feature extraction, achieving high performance in AD prognosis, and providing a model for visualizing brain regions affected by the disease. The study explores the relationship between cognitive symptoms, neurodegeneration, and imaging features, shedding light on the classification and understanding of Alzheimer’s Disease progression.

Dong et al. [5] discuss the use of curcumin-conjugated magnetic nanoparticles (Cur-MNPs) to target Aβ pathologies in Alzheimer’s disease (AD). The study demonstrates that Cur-MNPs can target both Aβ plaques and Aβ oligomers at early and late stages of AD progression. The technology allows for the visualization of Aβ pathologies using molecular MRI, providing a potential tool for early AD diagnosis and drug development. Grants supported the research and hold promise for detecting and monitoring AD progression in patients.

Seifallahi et al. [6] focus on utilizing the Timed Up and Go (TUG) test with a Kinect V.2 camera and machine learning to detect Alzheimer's Disease (AD) in its early stages. By analyzing joint position data from TUG performances of AD patients and healthy controls, the study identified significant features related to balance and gait. The machine learning approach, particularly the support vector machine classifier, accurately distinguished AD and healthy individuals.

Li et al. [7] discuss a study on Feature Extraction and Identification of Alzheimer's Disease using Multi-Channel EEG data. The research focuses on decoding latent factors from brain activity to understand cognitive impairment in Alzheimer's disease. Key findings include power spectrum characteristics and dominant frequency differences between Alzheimer's disease and normal groups. Analysis of latent factor distribution in the theta frequency band reveals distinct patterns between the two groups, providing insights for disease identification.

Zhang et al. [8] proposed a novel tensor multi-task ensemble learning method for predicting Alzheimer's disease (AD) progression using brain biomarkers. The approach outperforms benchmarks and state-of-the-art methods in predicting cognitive scores, demonstrating improved stability and accuracy. The model enhances prediction accuracy for future time points by aggregating MRI data over time. The study's findings have significant implications for early detection and intervention in AD, potentially influencing future research and clinical practices.

Ahmed et al. [9] focus on structural magnetic resonance imaging (SMRI) data from the Gwangju Alzheimer's and Related Dementia (GARD) cohort dataset. Due to data scarcity, the approach involves using ensembles of simple convolutional neural networks (CNNs) as feature extractors, SoftMax cross-entropy as the classifier, and a patch-based approach. The final decision-making procedure involves a weighted voting strategy, where the decision scores of individual models contribute to the final decision, leading to a significant increase in accuracy.

Chen et al. [10] proposed a novel model for Alzheimer's disease (AD) multi-classification that combines latent space learning and feature learning. The model uses MRI data from the ADNI dataset and sets up two multi-classification tasks: AD3 (NC vs. MCI vs. AD) and AD4 (NC vs. sMCI vs. pMCI vs. AD). Multiple ROI templates extract features, and the model selects the most discriminative features for classification. Comparative experiments show that the proposed model outperforms other methods in terms of evaluation metrics. The study emphasizes the importance of using multiple ROI templates and exploring their inter-relationship to enhance AD diagnosis.

4. Proposed methodology

4.1. AdaBoost algorithm

AdaBoost and Adaptive Boosting are mainly used for classification tasks. This algorithm is a powerful ensemble learning algorithm that combines the multiple weak predictions of multiple weak learners to create a strong classifier. This algorithm works by assigning equal weights to each training sample in the dataset. It trains the weak learners iteratively. Each iteration fits a weak learner in the training data. The algorithm pays more attention to training samples at each iteration, which previous weak learners misclassify. This combines the prediction of weak learners by weighted voting, where the weight assigned to each weak learner is based on its accuracy on the training data. After each iteration, the weights of the training samples are updated based on the performance of the weak learner. The training process continues until a predefined number of times is determined by a hyperparameter called the number of estimators. AdaBoost is particularly effective in improving the classification accuracy of weak learners and combining the predictions of multiple models.

4.2. LightGBM Algorithm

LightGBM Algorithm is a gradient boosting framework, an algorithm developed by Microsoft that is widely used for classification and provides high performance. This algorithm builds a strong model predictive model by sequentially combining the predictions of multiple weak learners. LightGBM follows a depth-first method. This algorithm uses gradient optimization techniques to improve training speed and efficiency. It uses histograms to approximate the loss function's gradients, reducing its memory usage and other associated computational costs. It also supports feature parallelism, which enables us to split the data into multiple subsets and compute histograms for each subset in parallel. It uses regularization techniques, including L1 L2 regularization, to prevent overfitting and improve generalization performance. This powerful and efficient algorithm handles large-scale and complex datasets. It has unique features such as life-wise growth, gradient-based optimization, and histogram-based splitting, making it well-suited for handling complex datasets.

4.3. CatBoost Algorithm

CatBoost is a powerful gradient-boosting algorithm designed to handle categorical features effectively. CatBoost algorithm is also based on the gradient boosting framework, which sequentially builds on ensemble weak learners. It reduces the loss function by adding new trees to correct the errors made by the existing ensemble. The main strength of the boosting algorithm is the ability to handle categorical variables without using encoding techniques. This algorithm encodes categorical variables internally, such as target and one-hot encoding, allowing it to incorporate categorical information into the model efficiently. It is an ordered boosting technique in which trees are built in a particular order based on the values of categorical features, which helps reduce overfitting and improve generalization performance by ensuring that each tree learns from the mistakes of the previous trees. This algorithm uses L1 L2 regularization techniques to prevent the model from memorizing the noise in the training data and improve its ability to generalize to unseen data. This algorithm generates a relatively easier-to-understand model, as well as how the model makes predictions and identifies important features.

4.4. Architecture diagram

Figure 1 illustrates a sequence of activities and steps involved in classifying Alzheimer’s disease. The initial step is to process the data using the panda’s library to gain insights about the data trained by the machine learning model. After processing the input data, we check for null values. If null values are present in the dataset, we can use the fill () method to fill the values in the particular column. These are the steps involved in preprocessing. After completing the preprocessing technique, visualizing data is very important, as it helps gain insights about the data and how data is distributed in the dataset. It also helps determine how different features correlate with each other. Data visualization allows us to identify outliers, anomalies, and potential patterns relevant to the classification task. Visualizing plays a crucial role in understanding the underlying structure of data.

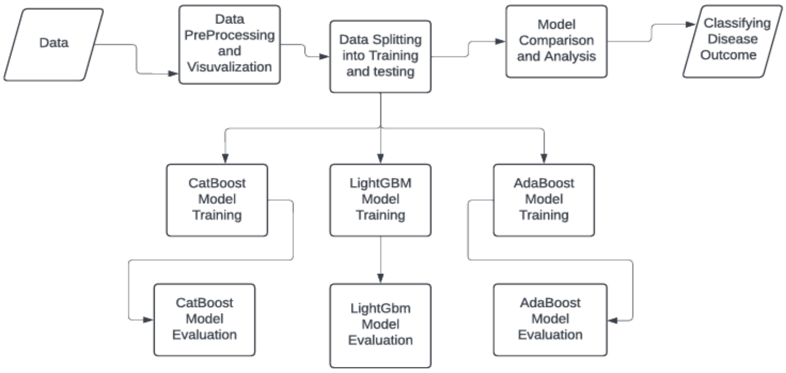
Data
Data PreProcessing and Visuvalization
Data Splitting into Training and testing
Model Comparison and Analysis
Classifying Disease Outcome
CatBoost Model Training
LightGBM Model Training
AdaBoost Model Training
CatBoost Model Evaluation
LightGbm Model Evaluation
AdaBoost Model Evaluation

Figure 1: Architecture Diagram

After visualizing the data, we need to split the data into training and testing data. The training data consists of 80% of the data, and the testing data consists of 20%. The model learns efficiently from the amount of data given in the training datasets to capture underlying relationships and patterns. The testing data is reserved for testing the trained model’s performance on unseen data. This testing data allows us to assess how well the model generalizes to new and unseen data and provides an estimate of how it performs. The data is trained with the Catboost, LightGBM, and AdaBoost algorithms, and their metrics are evaluated to assess the effectiveness of each algorithm in classifying Alzheimer’s disease. After evaluating the all-model metrics, these model metrics are compared to find the best-performing algorithm. The comparison is based on metrics like accuracy, precision, recall, and F1 score of each algorithm across the testing dataset. By comparing these metrics, we can determine which algorithm achieves the overall performance and is most effective at accurately classifying Alzheimer’s disease.

4.5. Algorithm

```
Step 1: Loading the dataset
Step 2: Preprocess and visualize the data
        2.1 Handling missing values
        2.2 Plotting scatter, bar plot to gain insights about the data
        2.3 Encode categorical variables
        2.4 Split data into features and target variables
        2.5 Split data into train and test data
Step 3: Training CatBoost model
```

```
    3.1 Initializing CatBoostClassifier with desired parameters
    3.2 Fitting the model into the training data
    3.3 Evaluate model performance on the testing data
    3.4 Store performance metrics
Step 4: Training AdaBoost model
    4.1 Initializing AdaBoostClassifier with desired hyperparameters
    4.2 Fit the model into the training data
    4.3 Evaluate model performance on the testing data
    4.4 Store performance metrics
Step 5: Training LightGBM model
    5.1 Initializing LightGBM Classifier with desired hyperparameters
    5.2 Fit the model into the training data
    5.3 Evaluate model performance on the testing data
    5.4 Store performance metrics
Step 6: Compare model performance
    6.1 Comparing performance metrics
    6.2 Identifying the best-performing algorithm based on the evaluation results.
Step 7: Plot graph for results
    7.1 Plot bar graph for comparing performance metrics of algorithms.
    7.2 Plotting graph for Precision, Recall, F1 -score, and accuracy metrics.
```

4.6. Formulas

Accuracy: Accuracy measures the proportion of correctly classified instances out of the total number of instances.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Precision: Precision measures the proportion of true positive prediction out of all the positive instances in the dataset

$$\text{Precision} = \frac{TP}{TP+FP}$$

Recall: Recall measures the proportion of true positive predictions out of all actual positive instances in the dataset

$$\text{Recall} = \frac{TP}{TP+FN}$$

F1 – score: The F1 – score is the harmonic mean of precision and recall. It provides a balance between precision and recall.

$$\text{F1 – score} = \frac{Two*Precision*Recall}{Precision+Recall}$$

Where:

TP (True Positives): Number of correctly classified positive instances
TN (True Negatives): Number of correctly classified negative instances
FP (False positives): Number of incorrectly classified positive instances
FN (False Negatives): Number of incorrectly classified negative instances

4.7. Execution

CatBoost algorithm: To implement the catboost model for Alzheimer’s disease prediction, we need to install the catboost library. To install catboost, the following command must be executed in the terminal.

```
pip install catboost
```

Once the catboost library is installed, we can import the catboost library into Google colab and use it to train and test our model for disease classification.

AdaBoost algorithm: To implement the adaboost algorithm, we can use the scikit-learn library in Python. AdaBoost is a part of the scikit-learn library. If the sci-kit-learn library is not installed, we can use the following command in the terminal.

```
Pip install sci-kit-learn
```

Then, the AdaBoost classifier algorithm can be imported from the sci-kit-learn library in Google Colab to train and evaluate the model for Alzheimer's disease prediction.

```
from sklearn. ensemble import AdaBoostClassifier
```

LightGBM algorithm: To implement the LightGBM algorithm, we need to install the LightGBM library. To install LightGBM, the following command must be executed in the terminal.

```
Pip install lightgbm
```

Once the LightGBM library has been installed in Google Colab, we can use it to train and evaluate our model for Alzheimer's classification.

5. Implementation

5.1. Data and Preprocessing

The dataset is obtained from the UCI machine learning repository and includes characteristics such as "air_time1," "disp_index1," "gmrt_in_air1," "max_x_extension1," and "max_y_extension1," among others. There is also a target variable labelled "class," which most likely represents the class or category to which each observation belongs, indicating that it is a categorical variable. A series of preprocessing activities are carried out to prepare the dataset for analysis and model construction. First, the data is put into a Data Frame for fast manipulation and analysis. The characteristics and the target variable are then split, with the former saved in a Data Frame labelled X and the latter as y.

Any missing or null values in the dataset are handled using suitable methods such as imputation or deletion. Categorical variables are generally encoded in a numerical manner suited for machine learning methods, such as Label Encoding or One-Hot Encoding. Feature scaling approaches are used to guarantee that features are of comparable size and that no one feature dominates during model training. The dataset is divided into training and testing sets to allow for unbiased model assessment, with one subset utilized for model training and the other for evaluation.

5.2. Data Visualization

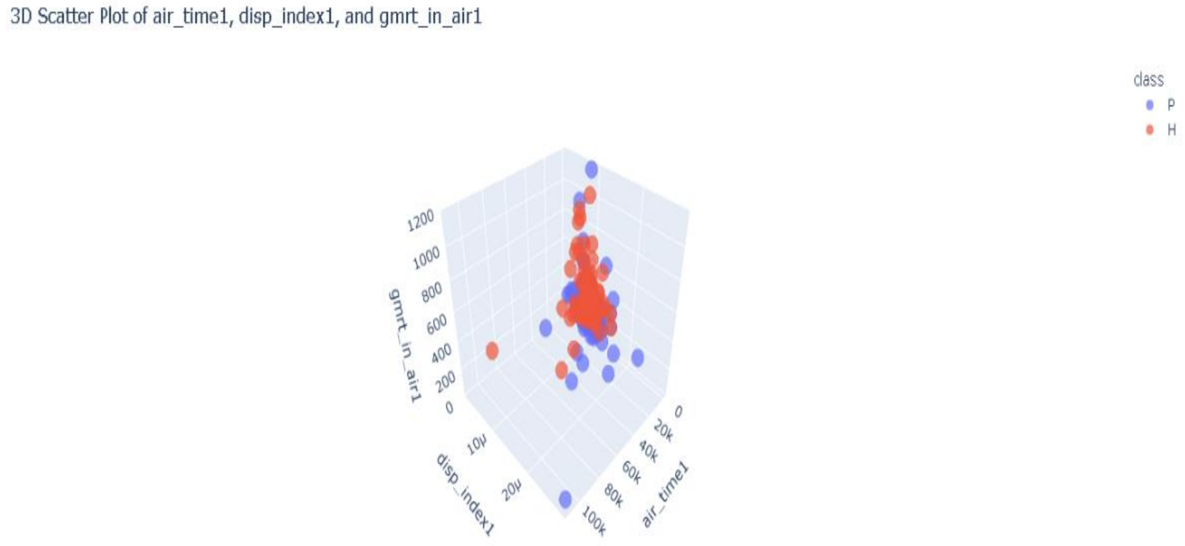

Figure 2 :3D Scatter Plot of air_time1, disp_index1, and gmrt_in_air1

Figure 2 depicts the association between three variables: “air_time1,” “disp_index1,” and “gmrt_in_air1.” Each point on the plot represents an observation from the dataset, and the values of these three variables determine its location. Each point’s colour correlates to the class or category the observation belongs to, providing easy visual discrimination across classes. This visualization provides a thorough perspective of the distribution and clustering of data points in three dimensions, allowing for the detection of observable patterns or trends across classes. Overall, this 3D scatter plot is an effective exploration tool for learning about the connections between various factors and their predictive value for the target variable.

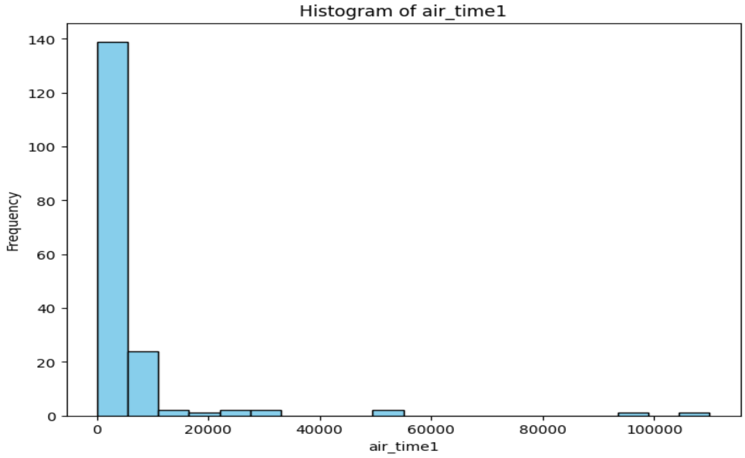

Figure 3: Histogram of air_time1

Figure 3 is a Histogram of the dataset’s ‘air_time1’ feature. Histograms are graphical representations of numerical data distributions that show the frequency of observations falling inside specific intervals or bins. In this figure, the ‘air_time1’ feature is the variable of interest, with its values divided into 20 evenly spaced bins along the x-axis. The height of each bar in the histogram indicates the frequency or count of data that falls into the specified bin. This histogram provides insight into the distribution of values inside the ‘air_time1’ feature, including the central tendency, dispersion, and any potential outliers. Examining this figure provides insight into the usual duration of airtime occurrences and the dataset’s general distribution pattern.

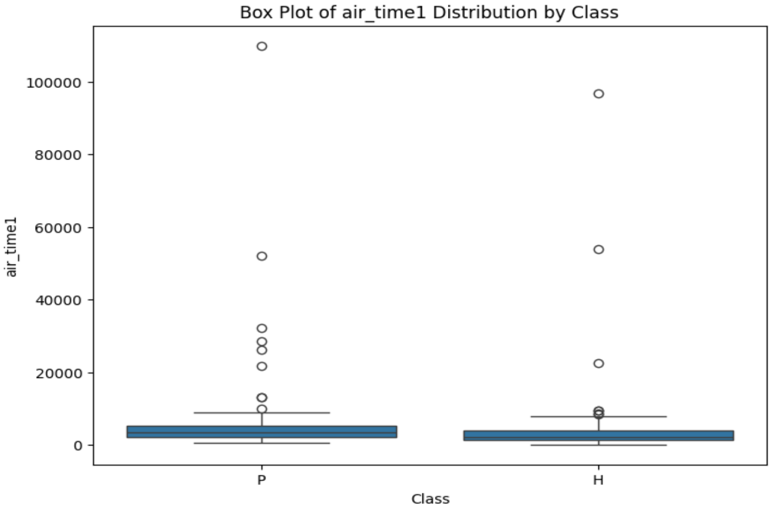

Figure 4: Box Plot of air_time1 Distribution by Class

Figure 4 is a box plot that depicts the distribution of the ‘air_time1’ feature across the dataset’s classes. A box plot, also known as a box-and-whisker plot, depicts the distribution of numerical data using quartiles, offering information on the central tendency, spread, and presence of outliers within each category. In this graphic, the x-axis indicates the classes or categories, while the y-axis shows the ‘air_time1’ feature values. Each box in the graphic shows the interquartile range (IQR), with a horizontal line indicating the median value. The “whiskers” extend from the box to highlight the data’s range, omitting outliers represented by individual points beyond the whiskers. This visualization enables a simple comparison of the distribution of airtime durations across classes, revealing possible variations or parallels in distribution patterns.

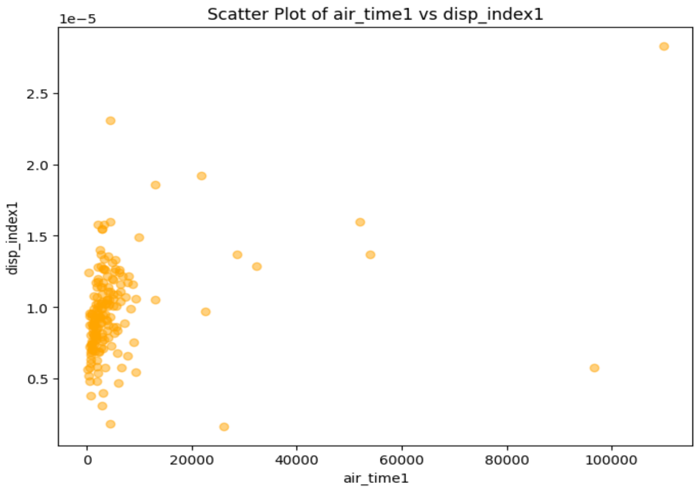

Figure 5: Scatter Plot of air_time1 vs disp_index1

Figure 5 is a scatter plot showing the connection between the variables' air_time1' and 'disp_index1'. Each data point in a scatter plot indicates a combination of two variables' values, with 'air_time1' values drawn on the x-axis and 'disp_index1' values shown on the y-axis. Scatter plots are widely used to show the correlation or relationship between two continuous data. In this scenario, the graphic shows the relationship between the duration of airtime ('air_time1') and the displacement index ('disp_index1'). The x-axis labels "air_time1" and the y-axis "disp_index1" clearly identify the variables displayed. Overall, this scatter plot makes it easier to visually grasp the relationship between the two variables and spot any patterns or trends in the data.

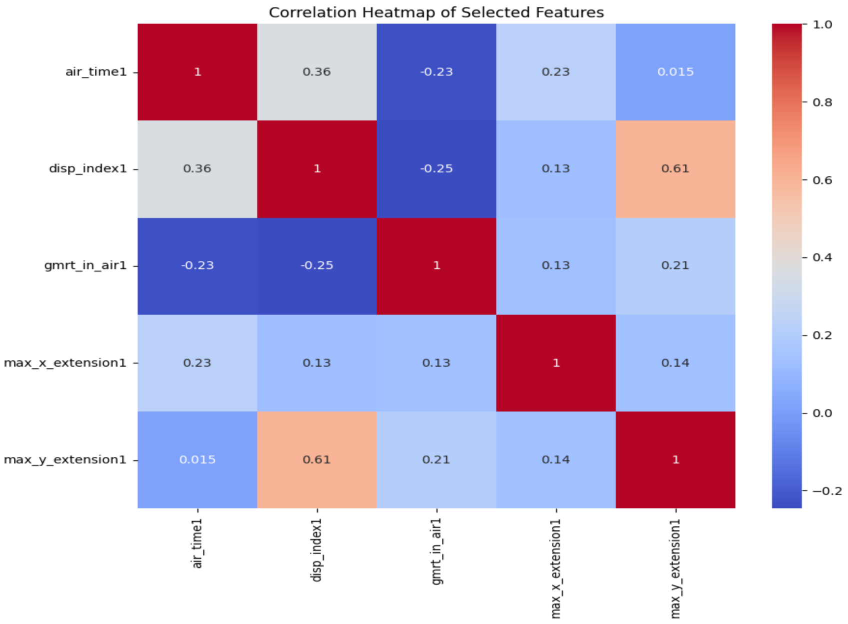

Figure 6: Correlation Heatmap

Figure 6 depicts the heatmap of five chosen features: 'air_time1', 'disp_index1', 'gmrt_in_air1', 'max_x_extension1', and'max_y_extension1'. It then computes the correlation coefficients between these attributes, resulting in a correlation matrix. This matrix depicts the pairwise correlations of the specified attributes, with values ranging from -1 to 1. Each cell in the heatmap represents the correlation coefficient between two characteristics, with colour intensity indicating the strength and direction of the link. Warm colours (e.g., red) denote positive correlations, whereas cold colours (e.g., blue) indicate negative correlations. This visualization may help you spot patterns and interdependence between variables and comprehend how changes in one feature may affect changes in another. High correlation coefficients imply strong associations, whereas low or near-zero values indicate weak or no correlations. Overall, the correlation heatmap is an effective tool for exploratory data analysis and feature selection within the dataset.

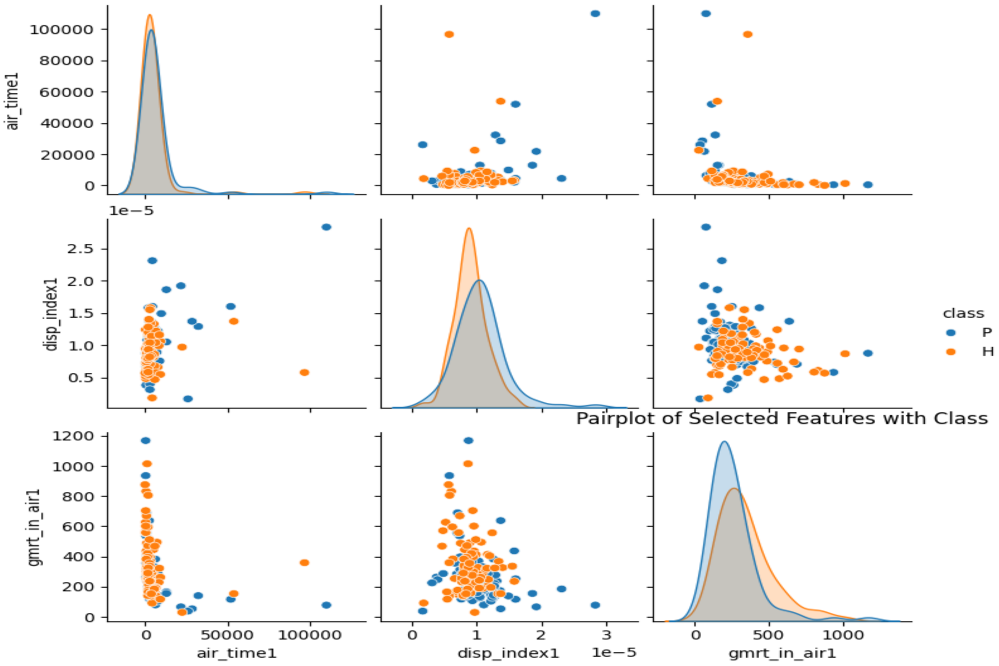

Figure 7: Pair Plot

Figure 7 is a Pair plot focusing on four selected features (‘air_time1’, ‘disp_index1’, ‘gmrt_in_air1’) and the target variable ‘class.’ This pair plot depicts pairwise correlations between these characteristics, allowing for the simultaneous investigation of univariate distributions and bivariate connections. Each scatter plot in the pair plot matrix depicts the correlation between two characteristics, with points coloured according to the various classes in the ‘class’ variable. Pair plots are useful tools for exploratory data analysis because they reveal data distributions, correlations, and class separations, which may be used to drive future modeling decisions.

5.3. Training

Three distinct models are trained: CatBoost, LightGBM, and AdaBoost classifiers. CatBoost and LightGBM are gradient-boosting methods, whereas AdaBoost is an ensemble approach. The training consists of dividing the data into training and testing sets, imputing missing values, scaling features, and fitting the models to the training data.

CatBoost: The CatBoost library is included in the Python environment. A CatBoostClassifier object is created using hyperparameters such as iterations, learning rate, and depth. These hyperparameters regulate the number of boosting iterations, the learning rate at each iteration, and the depth of the individual trees. The CatBoostClassifier is learned using the training data. CatBoost handles category characteristics automatically during training, eliminating the need for explicit coding. After training, the learned CatBoost model predicts the test set.

LightGBM: An LGBMClassifier object is created using the following hyperparameters: boosting type, num_leaves, max_depth, learning_rate, and n_estimators. These hyperparameters determine the boosting method, the maximum number of leaves in each tree, the maximum depth of the trees, the learning rate at each iteration, and the total number of boosting iterations. The instantiated LGBMClassifier is trained using the training data. LightGBM automatically supports category characteristics, eliminating the need for explicit encoding.

AdaBoost: An AdaBoostClassifier object is created with certain hyperparameters, such as n_estimators. This hyperparameter determines the maximum number of weak learners employed in the ensemble. Missing values are imputed using the Simple Imputer technique, and features are scaled with the StandardScaler.The instantiated AdaBoostClassifier is trained using the preprocessed training data. Following training, the AdaBoost model generates predictions on the preprocessed test set.

5.4. Evaluation

Performance Metrics Calculation: After making predictions on the test set with each model, performance measures like accuracy, precision, recall, and F1-score are determined.

Accuracy: Accuracy is the percentage of properly categorized cases among all instances. Precision is the ratio of genuine positive predictions to all positive predictions, demonstrating the model's ability to prevent false positives.

Recall: Recall measures the proportion of true positive predictions among all real positive cases, demonstrating the model's ability to detect positive instances.

F1-score: The F1-score is the harmonic mean of accuracy and recall, fairly assessing a model's overall performance.

Classification Report: A classification report is created for each model, which includes a full breakdown of precision, recall, and F1-score for each class in the dataset. This report provides information on how well each model performs across different classes, which can aid in the identification of possible class-specific performance concerns.

Confusion Matrix Visualization: Confusion matrices are created for each model, displaying the number of true positive, true negative, false positive, and false negative predictions made by the model. These matrices thoroughly depict the model's classification performance, showing any misclassification patterns.

Learning Curve Visualization: Each model's learning curve shows how its performance varies as the amount of training data increases. These curves aid in diagnosing problems such as overfitting or underfitting by visualizing the model's training and validation scores as a function of the training set size.

6. Results and Discussion

We chose Python to develop our Alzheimer classification model for this experiment. The proposed model was trained and evaluated on Windows 11 with an Intel core i5 12450H processor,16 GB RAM, and GTX 1650 GPU, and the experiment was on the Google platform. The dataset has been trained and tested using the CatBoost, AdaBoost, and LightGBM models. The dataset was split into 80% and 20 % for training and testing. The model was evaluated by the following metrics: Accuracy, precision, Recall, and F1-score to evaluate the effectiveness of the model.

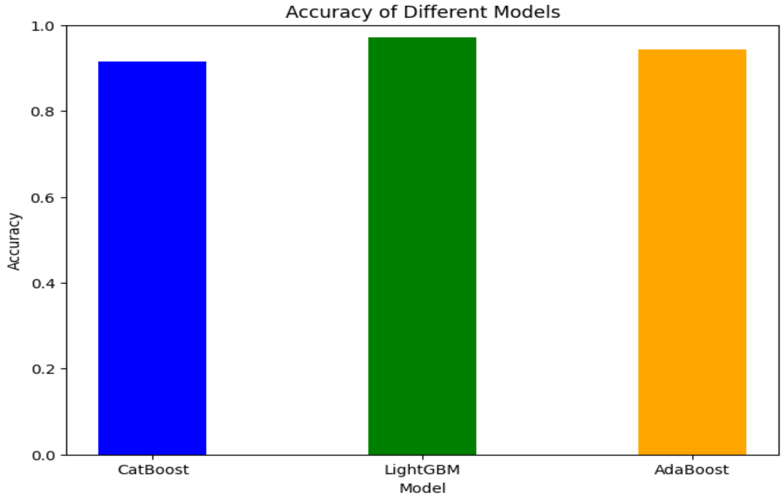

Figure 8: Accuracy of different models

Figure 8 illustrates the accuracy of different machine-learning models. The graph indicates that the LightGBM model has an accuracy of 0.9714, followed by the AdaBoost algorithm, which has an accuracy of 0.9428, followed by the catboost algorithm of 0.9142. Table 1 illustrates the accuracy of all the models. The highest accuracy is achieved by the LightGBM model, followed by the Adaboost model and then CatBoost Model. These values provide insights into the performance of each model in classifying Alzheimer's disease.

Table 1: Accuracy metrics

Model	Accuracy (%)
LightGBM	97.14
CatBoost	91.42
AdaBoost	94.28

Table 2: Precision and Recall Metrics

Model	Precision (%)	Recall (%)
LightGBM	97.32	97.14
CatBoost	92.85	91.42
AdaBoost	94.28	94.28

Table 2 illustrates the precision and recall metrics of the machine learning model used in Alzheimer's disease classification. Precision is the proportion of true positive prediction made by the model. The highest precision is achieved by the LightGBM model, followed by the AdaBoost model and then the Catboost model. Recall measures the ability of classification models to identify all relevant instances in the dataset. It is the proportion of true positive predictions from all positive instances in the dataset.

Table 3: F1 – Score

Model	F1-Score (%)
LightGBM	97.15
CatBoost	91.47
AdaBoost	94.28

Table 3 illustrates the f1-scores of the machine learning model used in the experiment. F1 -score is the harmonic mean of precision and recall. The LightGBM model has achieved the highest f1-score of 97.15%, followed by the adaboost and catboost algorithms, respectively.

6.1. Confusion matrices for algorithms

A confusion matrix is a table used in classification problems to evaluate the performance of the classification machine learning models. It provides a detailed overview of the classification results and shows the relationship between actual and predicted classes. It helps assess how well the model correctly predicts classes and where the machine learning algorithm is making mistakes.

6.1.1. Confusion matrix for CatBoost algorithm

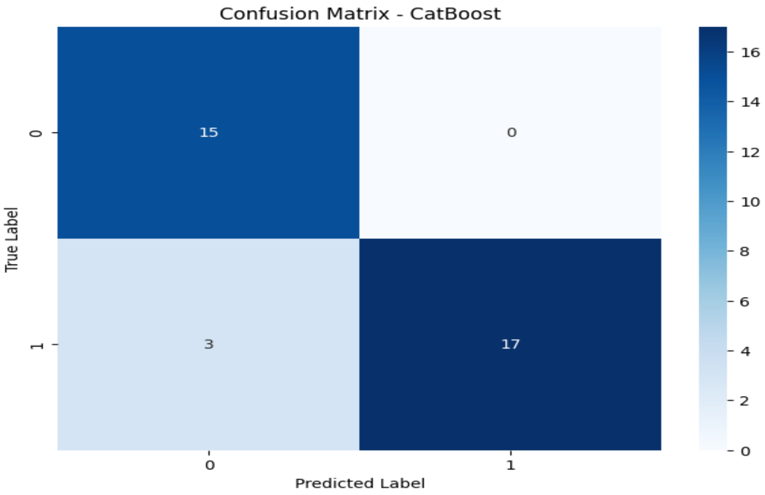

Figure 9: Confusion matrix for the catboost algorithm

Figure 9 illustrates that the model performs well in identifying the positive instances, which indicates that it successfully predicts the all-true positive cases without errors. The model also performs well in identifying negative instances, with a small number of false positives (3) but a larger number of true negatives (17). From the figure, we can see the absence of false positives, which indicates that the model seems more cautious in classifying positives while being highly accurate in predicting the negative class.

6.1.2. Confusion matrix for LightGBM algorithm

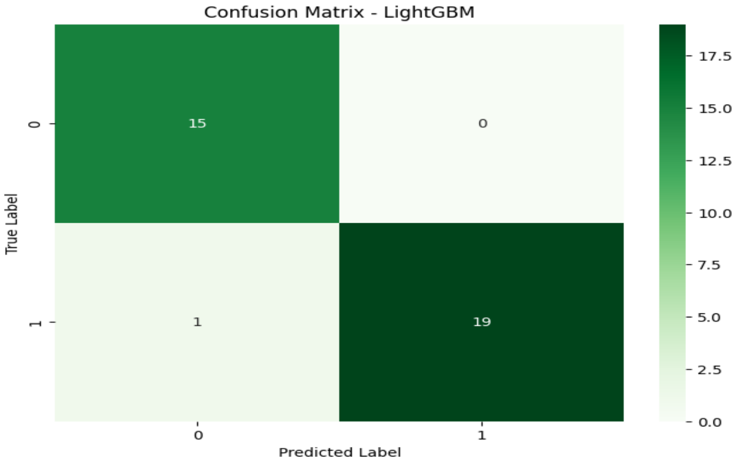

Figure 10: Confusion matrix for LightGBM algorithm

Figure 10 depicts that the model consists of 15 True positive values, indicating it has classified individuals with Alzheimer’s disease. It also has classified 19 True negative values, which indicates the model has classified 19 individuals who do not have the disease. One false negative value indicates that the model has misclassified one individual as not having the disease when they have it. There are no false positive values, which indicates that the model did not incorrectly classify an individual as having a disease when they do not.

6.1.3. Confusion matrix for AdaBoost algorithm

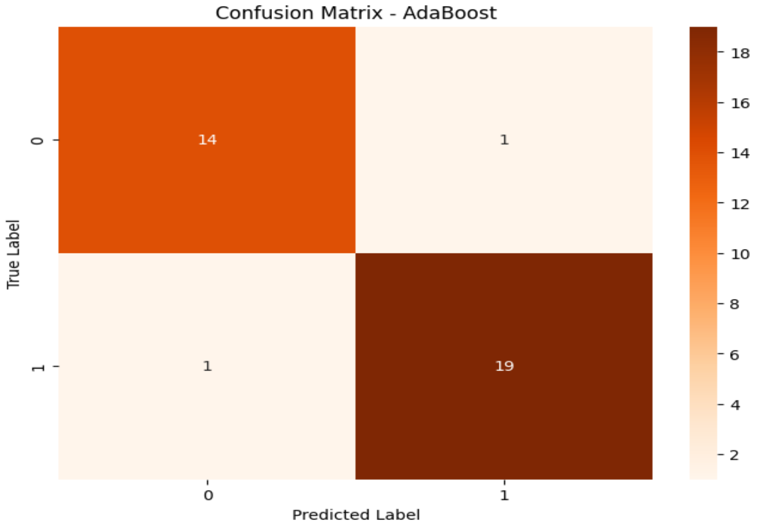

Figure 11: Confusion matrix for AdaBoost algorithm

Figure 11 illustrates that the model has classified 14 true positives, indicating that it has accurately classified 14 individuals with the disease. There are 19 true negatives, indicating that the model has correctly classified 19 individuals who do not have the disease. The model has one false positive, indicating it has misclassified one individual as having the disease when the individual does not. There is also a presence of one false negative, indicating the model incorrectly classified one individual as not having the disease when they have the disease.

6.2. Learning Curves

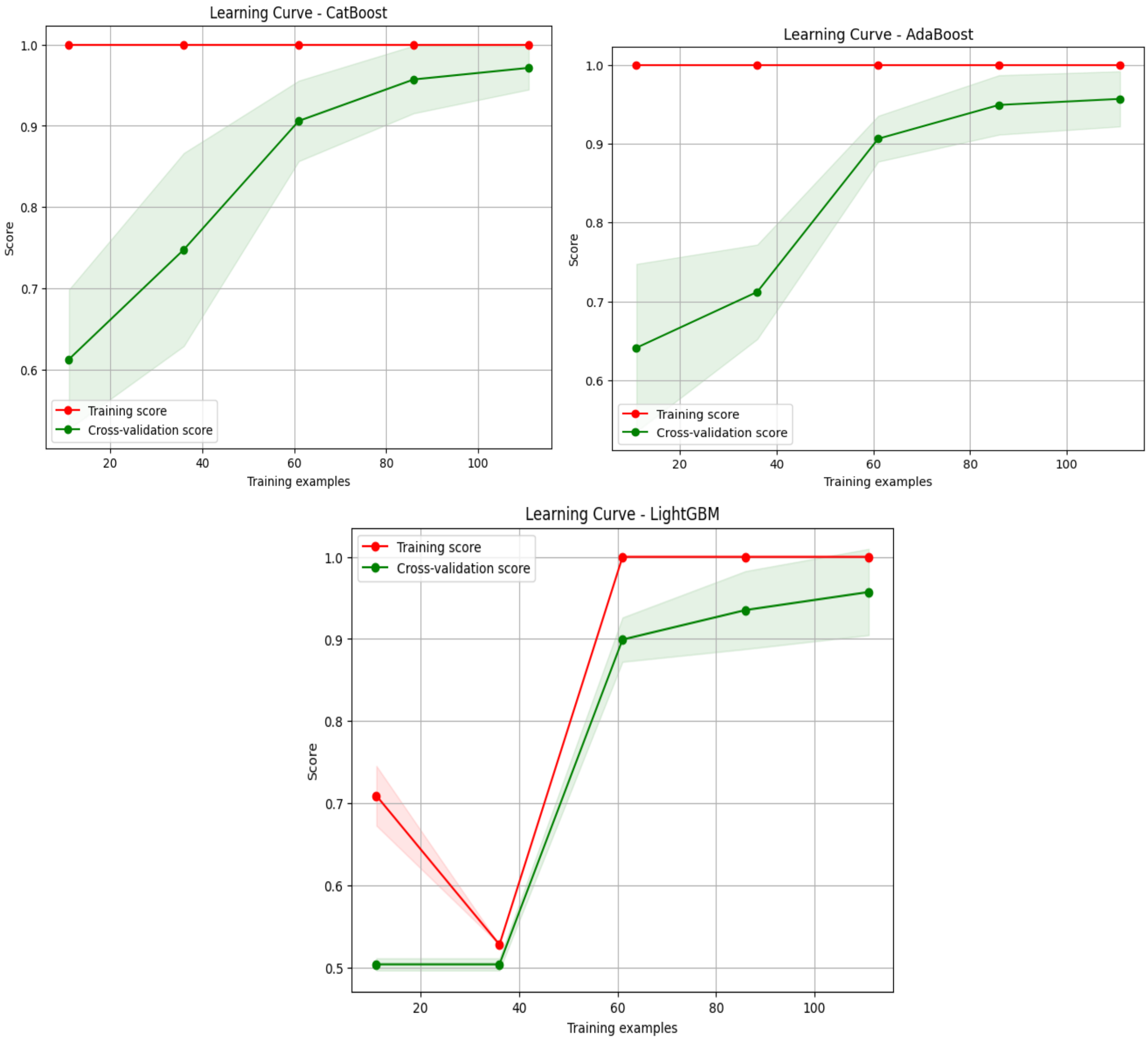

Figure 12: The learning curve of all algorithms

Figure 12 depicts a learning curve, a graphical representation of the model's performance over time as it learns from the training data. It is a visualization of training data size on the model's performance, and the model's ability to generalize improves as it learns from the dataset given as input to the model. A learning curve is usually visualized using training and cross-validation score curves to check whether the model is overfitting, underfitting, or balanced. The training score curve represents the model's performance on the training dataset as the training data increases. It indicates how well the model is learning the patterns in the training data. The cross-validation curve represents the model's performance on a validation set as training data increases. It describes how well the model generalizes to new and unseen data. This learning curve is useful for adjustments to the model or data for better performance.

7. Conclusion

In the classification of Alzheimer's disease using LightGBM, AdaBoost, and CatBoost machine learning algorithms, all three models have exhibited promising results. Each algorithm showed strengths in different aspects, contributing to the classification task's overall effectiveness. LightGBM demonstrated accuracy, a value of 97.14%, making it particularly effective in correctly classifying instances of Alzheimer’s disease. Its training speed and scalability make it suitable for handling large datasets. Adaboost demonstrated an accuracy of 94.28% and proved effectively strong in generalization capabilities and effectively learned from training data, performing well on unseen data. Its ensemble approach of combining multiple weak learners helped to overcome overfitting and improved the model’s robustness. Catboost demonstrated an accuracy of 91.42% and excelled in handling categorical variables. Its built-in handling of categorical variables and automatic feature scaling streamlined the modelling process. The combination of LightGBM, AdaBoost and CatBoost algorithms offers a comprehensive approach to Alzheimer’s disease classification, leveraging the strength of each algorithm to achieve high accuracy, precision, and generalization performance. Overall, this study highlights the potential of machine learning algorithms in improving the accuracy and reliability of Alzheimer’s disease classification.

Acknowledgment: The support of all my co-authors is highly appreciated.

Data Availability Statement: The research contains data related to Alzheimer’s Disease and associated metrics obtained from the UCI machine learning repository.

Funding Statement: No funding has been obtained to help prepare this manuscript and research work.

Conflicts of Interest Statement: No conflicts of interest have been declared by the author(s). Citations and references are mentioned in the information used.

Ethics and Consent Statement: The consent was obtained from the organization and individual participants during data collection, and ethical approval and participant consent were received

References

1. A. Basher, B. C. Kim, K. H. Lee, and H. Y. Jung, “Volumetric feature-based Alzheimer’s disease diagnosis from sMRI data using a convolutional neural network and a deep neural network,” IEEE Access, vol. 9, no.2, pp. 29870–29882, 2021.
2. Z. Qu, T. Yao, X. Liu, and G. Wang, “A graph convolutional network based on univariate neurodegeneration biomarker for Alzheimer’s disease diagnosis,” IEEE J. Transl. Eng. Health Med., vol. 11, no.6, pp. 405–416, 2023.
3. C. S. Eke, E. Jammeh, X. Li, C. Carroll, S. Pearson, and E. Ifeachor, “Early detection of Alzheimer’s disease with blood plasma proteins using support vector machines,” IEEE J. Biomed. Health Inform., vol. 25, no. 1, pp. 218–226, 2021.
4. F. J. Martinez-Murcia, A. Ortiz, J.-M. Gorriz, J. Ramirez, and D. Castillo-Barnes, “Studying the manifold structure of Alzheimer’s disease: A deep learning approach using convolutional autoencoders,” IEEE J. Biomed. Health Inform., vol. 24, no. 1, pp. 17–26, 2020.
5. C. M. Dong et al., “Early detection of amyloid β pathology in Alzheimer’s disease by molecular MRI,” in 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 2020.
6. M. Seifallahi, A. H. Mehraban, J. E. Galvin, and B. Ghoraani, “Alzheimer’s disease detection using comprehensive analysis of timed up and go test via Kinect V.2 camera and machine learning,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 30, no.,6, pp. 1589–1600, 2022.
7. K. Li et al., “Feature extraction and identification of Alzheimer’s disease based on latent factor of multi-channel EEG,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 29, no.8, pp. 1557–1567, 2021.
8. Y. Zhang, T. Liu, V. Lanfranchi, and P. Yang, “Explainable tensor multi-task ensemble learning based on brain structure variation for Alzheimer’s disease dynamic prediction,” IEEE J. Transl. Eng. Health Med., vol. 11, no.11, pp. 1–12, 2023.
9. S. Ahmed et al., “Ensembles of patch-based classifiers for diagnosis of Alzheimer diseases,” IEEE Access, vol. 7, no.5, pp. 73373–73383, 2019.
10. Z. Chen, H. Lei, Z. Huang, and B. Lei, “Latent space learning and feature learning using multi-template for multi-classification of Alzheimer’s disease,” in 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Mexico, 2021.