## **Name : Fardin Alam 2031795642**

## 

## **1.Problem Statement -**

Alzheimer's Disease (AD) is a progressive neurological disorder that leads to memory loss, cognitive decline, and ultimately, loss of ability to carry out simple tasks. Early detection of Alzheimer's is critical for effective treatment planning and quality of life improvement. However, manual diagnosis through clinical evaluation and imaging techniques is time-consuming, expensive, and prone to human error.  
 This project aims to automate the detection and classification of Alzheimer’s Disease stages using machine learning techniques on patient data, thereby providing a faster, more accurate, and scalable solution for early diagnosis

## **2. Objectives**

The primary objectives of this project are as follows:

* To develop a machine learning model for Alzheimer's Disease detection that is both faster and more accurate than traditional models used in previous studies.
* To leverage ensemble learning techniques to enhance the predictive performance and robustness of the model.
* To optimize preprocessing, feature selection, and model hyperparameters to achieve better generalization with minimal computation time.
* To conduct a comparative evaluation against baseline methods reported in the literature, highlighting improvements in efficiency and accuracy.
* To propose a lightweight, scalable solution that can eventually be deployed in real-world clinical settings.

## **3. Literature Review**

Over the past decade, researchers have explored various machine learning techniques to detect and classify Alzheimer's Disease (AD) based on clinical and neuroimaging data. Traditional approaches largely relied on Support Vector Machines (SVMs), Decision Trees, and K-Nearest Neighbors (KNN), which provided moderate accuracy but often suffered from overfitting and high computational costs.

For instance, **Dyrba et al. (2015)** utilized SVMs on MRI and PET scans, achieving around 82% accuracy but with significant processing time and dependency on imaging data. **Moradi et al. (2015)** introduced a feature reduction method combined with a Random Forest classifier, improving model interpretability but still requiring complex preprocessing pipelines.  
 More recent works have shifted towards **ensemble methods** and **deep learning**, with **Suk et al. (2017)** applying deep sparse autoencoders combined with Random Forests, reporting accuracies above 85% at the cost of increased training time.

Despite these advancements, many existing models are either computationally expensive or lack robustness on unseen data. Moreover, deep learning models often require large datasets and significant hardware resources, making them less accessible in practical clinical environments.

In this project, we aim to bridge this gap by proposing an efficient ensemble-based approach that not only achieves competitive or superior accuracy but also ensures faster training and prediction times compared to more resource-intensive models

## **Network Description**

In this project, we adopted an ensemble learning approach to build a predictive model for Alzheimer’s Disease classification. Rather than relying on a single classifier, we combined the strengths of multiple models to achieve higher accuracy, robustness, and generalization. Ensemble methods are well known for their ability to reduce variance and bias, thereby often outperforming individual base models, especially on real-world, noisy datasets like medical records.

### **4.1 Base Models**

We selected the following algorithms as base models for the ensemble:

* **Random Forest Classifier** Random Forest (RF) is an ensemble of decision trees trained on different subsets of the data. Each tree votes for a class, and the final prediction is made based on the majority vote. RF is particularly effective in handling high-dimensional datasets and can capture complex feature interactions without heavy preprocessing. Key hyperparameters such as the number of trees (n\_estimators) and maximum tree depth (max\_depth) were carefully tuned to balance between bias and variance.
* **Gradient Boosting Classifier** Gradient Boosting (GB) builds trees sequentially, where each new tree focuses on correcting the errors made by the previous ones. Unlike Random Forest, which trains trees independently, Gradient Boosting improves upon the model iteratively. We employed learning rate tuning and limited the maximum depth of trees to prevent overfitting.
* **Extreme Gradient Boosting (XGBoost)** XGBoost is an optimized version of Gradient Boosting that introduces regularization to prevent overfitting and incorporates techniques like column subsampling and efficient tree pruning. It is highly efficient for both computation and memory, making it suitable for achieving our project's goal of building a fast, lightweight model.

### **4.2 Ensemble Strategy: Voting Classifier**

After training the three base models, we combined them using a **Voting Classifier**. Two ensemble strategies were considered:

* **Hard Voting**: Each base classifier predicts a class label, and the majority vote determines the final prediction.
* **Soft Voting**: The predicted probabilities of each class are averaged, and the class with the highest average probability is chosen.

Given that our base classifiers are all probability-calibrated, we opted for **Soft Voting**, as it often provides smoother and more reliable decision boundaries, especially when the individual models have varying strengths.

The final ensemble model leverages the diversity among Random Forest, Gradient Boosting, and XGBoost — models that each approach learning from different perspectives (bagging vs. boosting) — to produce more robust predictions.

### **4.3 Network Architecture Overview**

Although this project primarily relies on classical machine learning models rather than deep learning architectures (CNNs or RNNs), the ensemble can still be represented as a modular "network" in a broader sense:

* **Input Layer**: Patient data features such as age, gender, brain volume, MMSE score, etc.
* **Parallel Model Layers**: Three parallel "paths" — one through Random Forest, one through Gradient Boosting, one through XGBoost.
* **Aggregation Layer**: Soft Voting layer that aggregates the predicted probabilities from each model.
* **Output Layer**: Final class prediction — Non-Demented, Very Mild Demented, Mild Demented, or Moderate Demented.

This structure ensures that decision-making is not dominated by any single model but is a collective agreement among specialized predictors.

### **4.4 Hyperparameter Tuning**

To further optimize the model performance, we performed limited grid search and manual tuning for key hyperparameters:

* **Random Forest**: n\_estimators=300, max\_depth=20
* **Gradient Boosting**: learning\_rate=0.05, n\_estimators=250
* **XGBoost**: learning\_rate=0.03, n\_estimators=200, max\_depth=5, with regularization parameters alpha and lambda tuned based on validation performance.

These choices helped balance between underfitting and overfitting while keeping the model training time reasonable.

### **4.5 Model Training and Evaluation Summary**

The ensemble was trained on an 80% split of the dataset, with the remaining 20% used for testing. Five-fold cross-validation was employed during model selection to ensure the reliability of the results.  
 Performance was evaluated based on accuracy, precision, recall, F1-score, and ROC-AUC, providing a holistic view of how well the model generalized across different Alzheimer’s Disease stages.

## **Challenges Faced**

Throughout the development of this project, several practical and technical challenges were encountered:

### **5.1 Limited System Memory**

One major constraint was limited RAM availability during model training. The dataset itself was moderately sized, but ensemble methods, especially Random Forests and Gradient Boosting models, can be memory-intensive because they store multiple decision trees simultaneously in memory.  
 Training large ensembles or running grid searches across many hyperparameter combinations sometimes caused memory overflows or significant system slowdowns. To mitigate this, careful tuning of parameters like max\_depth and n\_estimators was performed, ensuring that models remained compact without severely sacrificing performance.

### **5.2 Batch Training and Memory Management**

Unlike deep learning models that naturally operate in mini-batches, classical machine learning models typically load the entire dataset into memory. This posed a challenge for efficient memory management, especially when applying oversampling techniques like SMOTE, which further increased dataset size.  
 We manually split the dataset during experimentation phases and used partial fitting where available (e.g., with XGBoost) to simulate batch learning behavior and manage memory more effectively.

### **5.3 Accelerating Training Time**

Initially, training epochs for models like Gradient Boosting were noticeably slow, especially when the number of estimators was high. In order to accelerate training:

* A lower learning rate with early stopping was applied where possible (especially in XGBoost).
* The number of boosting rounds was reduced after observing marginal gains beyond a certain point.
* Parallel processing (n\_jobs=-1) was enabled in models like Random Forest and XGBoost to fully utilize available CPU cores.

Through these optimizations, training time per model was significantly reduced without negatively impacting accuracy.

## **6. Conclusion**

In this project, we successfully developed an ensemble-based machine learning model for Alzheimer's Disease detection, focusing on both speed and accuracy. By combining Random Forest, Gradient Boosting, and XGBoost classifiers through soft voting, we achieved strong predictive performance while keeping training times manageable.  
 Despite facing challenges such as limited memory, training bottlenecks, and tuning complexities, careful optimization and resource management allowed us to meet our objectives.  
 Overall, the project demonstrates that with thoughtful design, classical ensemble models can still provide a fast, reliable alternative to heavier deep learning approaches, making them highly suitable for practical clinical applications where resources may be limited.