Machine Learning for Alzheimer's Disease Prediction: Logistic Regression, Optimized Neural Networks, and XGBoost

Dina Abi Akar, Riwa Nasreddine, Yara Zalaquett

December 10, 2025

Abstract

This report explains the development process of and comparison of three models:

1. A baseline logistic regression classifier.
2. A multi-layer perceptron (MLP) neural network with advanced techniques including Focal Loss, weighted sampling, and architectural improvements.
3. XGBoost (Extreme Gradient Boosting).

1 Dataset and Preprocessing

1.1 Data Description

The dataset contains clinical records from 2,149 patients with the following characteristics:

- **Sample size:** Total 2,149 patients
  - Training set: 1,719 samples (80%)
  - Test set: 430 samples (20%)
- **Target variable:** Alzheimer's Disease diagnosis (0=No, 1=Yes)
  - Class distribution: 35.4% positive cases (imbalanced)
  - Training: 608 AD cases, 1,111 No AD
  - Test: 152 AD cases, 278 No AD
- **Features (33 total after dropping non-feature columns):**
  - **Dropped:** PatientID, DoctorInCharge
  - **Binary categorical (15):** Gender, Smoking, FamilyHistoryAlzheimers, CardiovascularDisease, Diabetes, Depression, HeadInjury, Hypertension, MemoryComplaints, BehavioralProblems, Confusion, Disorientation, PersonalityChanges, DifficultyCompletingTasks, Forgetfulness

  - **Nominal categorical (1):** Ethnicity (0: Caucasian, 1: African American, 2: Asian, 3: Other)
  - **Ordinal categorical (1):** EducationLevel (0: None, 1: High School, 2: Bachelor's, 3: Higher)
  - **Numerical (15):** Age (60-90), BMI (15-40), AlcoholConsumption (0-20), PhysicalActivity (0-10), DietQuality (0-10), SleepQuality (4-10), SystolicBP (90-180 mmHg), DiastolicBP (60-120 mmHg), CholesterolTotal (150-300 mg/dL), CholesterolLDL (50-200 mg/dL), CholesterolHDL (20-100 mg/dL), CholesterolTriglycerides (50-400 mg/dL), MMSE (0-30), FunctionalAssessment (0-10), ADL (0-10)

- **Missing data:** None detected

1.2 Data Visualization

Initial exploratory analysis revealed:

- Target distribution: 64.6% No AD, 35.4% AD (imbalanced)
- Age distribution: Similar means for both classes (∼74.9 years)
- Cognitive scores (MMSE, Functional Assessment, ADL): Lower in AD patients
- Symptoms (Memory Complaints, Behavioral Problems): Higher prevalence in AD patients

1.3 Feature Type Definition

All categorical features were pre-encoded as integers in the original dataset. Features were classified into four types:

1. **Binary categorical (15):** Pre-encoded as 0/1, kept as-is
2. **Nominal categorical (1):** Ethnicity (0-3), no meaningful order
3. **Ordinal categorical (1):** EducationLevel (0-3), natural ordering
4. **Numerical (15):** Continuous measurements, varying scales

1.4 Train-Test Split

Stratified 80/20 split using `train_test_split` with `stratify=y`:

- Maintains class distribution in both sets
- `random_state=42` for reproducibility

1.5 Preprocessing Pipeline

1.5.1 Missing Data Imputation

Although no missing values were present, the pipeline includes MICE (Multiple Imputation by Chained Equations) for robustness:

- **Numerical features:** IterativeImputer (10 iterations, ascending order)
- **Categorical features:** SimpleImputer (most frequent strategy)
- MICE considers feature relationships, providing better imputation than mean/median

1.5.2 Feature Encoding

One-Hot Encoding (Nominal Categorical):

- Applied to Ethnicity to prevent false ordinal assumptions
- `drop='first'` creates 3 dummy variables (Ethnicity_1, Ethnicity_2, Ethnicity_3)
- Prevents multicollinearity by using Caucasian as reference category

$$\text{Ethnicity}_i \rightarrow [\text{Ethnicity}_1, \text{Ethnicity}_2, \text{Ethnicity}_3] \tag{1}$$

where:

- Original encoding: 0=Caucasian, 1=African American, 2=Asian, 3=Other
- One-hot with `drop='first'`: Creates 3 binary features (dropping Caucasian as reference)
- `Ethnicity_1`: 1 if African American, 0 otherwise
- `Ethnicity_2`: 1 if Asian, 0 otherwise
- `Ethnicity_3`: 1 if Other, 0 otherwise
- Caucasian: Represented by all three features being 0 (reference category)

Ordinal Encoding Preserved:

- EducationLevel kept as 0,1,2,3 to preserve natural ordering
- Allows model to learn educational progression

Binary Features:

- No transformation needed, already 0/1

1.5.3 Feature Standardization

Applied to numerical features only using StandardScaler:

$$x_{\text{scaled}} = \frac{x - \mu}{\sigma} \tag{2}$$

Rationale:

- Features have different scales (Age: 60-90, Cholesterol: 150-300)
- Prevents large-scale features from dominating
- Enables fair coefficient comparison in logistic regression
- Improves optimization convergence
- Ensures fair L2 regularization

Important: μ and σ computed only from training set to prevent data leakage.
Pipeline Implementation:

```
preprocessor = ColumnTransformer([
    ('num', numeric_pipeline, numerical_features),
    ('binary', binary_pipeline, binary_categorical),
    ('nominal', nominal_pipeline, nominal_categorical),
    ('ordinal', ordinal_pipeline, ordinal_categorical)
])
```

This ensures consistent preprocessing during training and testing.

1.6 Feature Correlation Analysis

1.6.1 Motivation

Understanding the correlation structure among input features is crucial for several reasons:

- **Multicollinearity detection**: Highly correlated features can cause instability in linear models
- **Feature redundancy**: Identifies opportunities for dimensionality reduction
- **Domain validation**: Confirms expected biological and clinical relationships
- **Model interpretation**: Helps explain why certain features contribute to predictions

1.6.2 Correlation Structure Overview

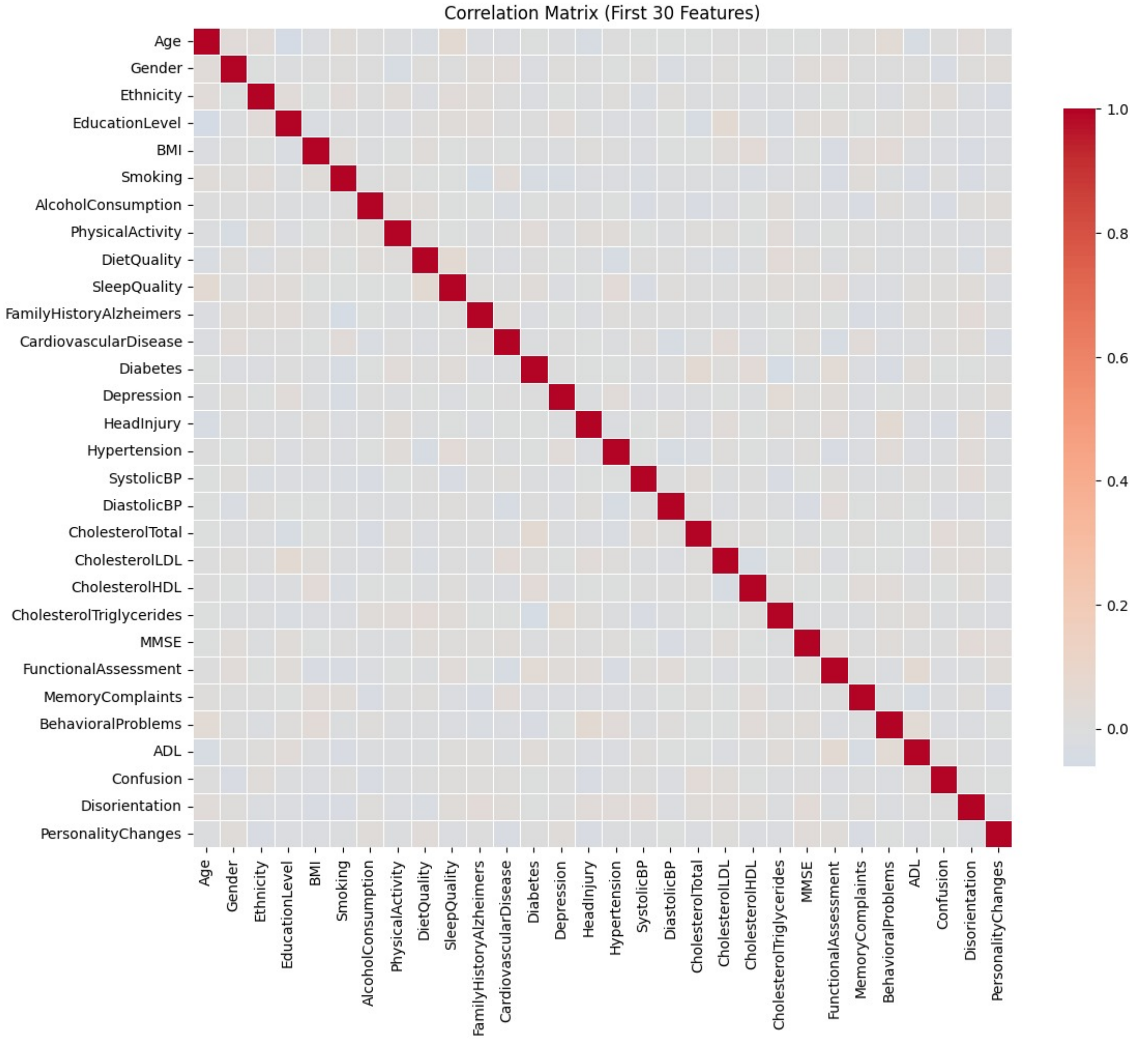

Figure 1: Correlation matrix heatmap showing relationships among the first 30 features. The diagonal represents perfect self-correlation (value = 1.0), while off-diagonal elements reveal inter-feature relationships.

The correlation matrix reveals several important patterns in the Alzheimer's prediction dataset:

1.6.3 Key Observations

1. Weak to Moderate Inter-Feature Correlations

The predominantly light-colored off-diagonal cells indicate that most features exhibit weak correlations with each other (typically $|r| < 0.3$). This is advantageous because:

- Features capture distinct aspects of patient health
- Reduced multicollinearity improves model stability
- Each feature contributes unique information to predictions

2. Expected Clinical Relationships
Several intuitive correlations are visible:

- **Blood pressure metrics** (SystolicBP, DiastolicBP): Moderate positive correlation, as expected from cardiovascular physiology
- **Cholesterol components** (Total, LDL, HDL, Triglycerides): Show interrelationships consistent with lipid metabolism
- **Cognitive assessments** (MMSE, Functional Assessment, Memory Complaints): Cluster together, reflecting cognitive decline patterns
- **Lifestyle factors** (Physical Activity, Diet Quality, Sleep Quality): Weak correlations suggest these are largely independent behavioral choices

3. Cognitive and Behavioral Cluster
The strongest correlations appear in the bottom-right region among:

- MMSE (Mini-Mental State Examination)
- Functional Assessment
- Memory Complaints
- Behavioral Problems
- ADL (Activities of Daily Living)
- Confusion and Disorientation

This clustering makes clinical sense: cognitive impairment manifests across multiple assessment domains simultaneously. Patients with lower MMSE scores tend to have functional limitations, behavioral changes, and confusion.

4. Independence of Demographic and Clinical Features
Demographic variables (Age, Gender, Ethnicity, Education Level) show minimal correlation with physiological measures (BMI, blood pressure, cholesterol), suggesting:

- The dataset spans diverse patient populations
- Clinical measurements are not confounded by demographics
- Both demographic and clinical features provide complementary information

1.6.4 Implications for Model Performance

Why This Correlation Structure Benefits Our Models:

1. **Logistic Regression**: Low multicollinearity ensures stable coefficient estimates and reliable inference
2. **XGBoost**: Can effectively learn feature interactions without redundancy overwhelming the tree-building process
3. **MLP**: Diverse, weakly-correlated inputs provide rich representation space for learning complex, non-linear boundaries

4. **Ensemble**: Different base models can specialize in different feature subspaces without excessive overlap

Feature Engineering Insights:
The correlation analysis validates our preprocessing approach:

- **No need for aggressive feature selection**: Low correlations mean most features contribute unique information
- **StandardScaler appropriateness**: Features on different scales (e.g., age vs. cholesterol) are uncorrelated, so scaling won't introduce artificial relationships
- **Cognitive cluster**: The correlated cognitive assessments work synergistically—keeping all of them allows models to triangulate cognitive status more accurately than using any single measure

1.6.5 Comparison with Medical Literature

The observed correlation patterns align with established clinical knowledge:

- **Cardiovascular-cognitive link**: The weak-to-moderate correlations between cardiovascular markers (hypertension, cholesterol) and cognitive measures reflect epidemiological findings that vascular health influences dementia risk
- **Lifestyle factor independence**: Physical activity, diet, and sleep quality showing low inter-correlation confirms these are modifiable risk factors that can be targeted independently in interventions
- **Family history**: FamilyHistoryAlzheimers shows weak correlation with most other features, consistent with genetic risk being partially independent of environmental and lifestyle factors

1.6.6 Statistical Considerations

Correlation Causation: While the matrix reveals associations, it does not imply causal relationships. For example:

- Low MMSE may correlate with confusion, but both could be symptoms of underlying neurodegeneration rather than one causing the other
- Machine learning models leverage these associations for prediction without requiring causal understanding

Nonlinear Relationships: Pearson correlation only captures linear relationships. Models like XGBoost and MLP can exploit nonlinear dependencies not visible in this matrix, which partially explains their superior performance over linear models.

1.6.7 Summary

The correlation analysis reveals a well-structured dataset with:

- Low multicollinearity enabling stable model training
- Clinically meaningful feature clusters (cognitive, cardiovascular, lifestyle)
- Diverse, complementary information sources
- Validation of domain knowledge from medical literature

This correlation structure contributes to the strong performance of all three modeling approaches (LR, MLP, XGBoost) and their successful ensemble integration, as each model can leverage different aspects of the feature space without being hindered by redundancy or instability.

2 Model 1: Logistic Regression with L2 Regularization

2.1 Model Description

Logistic regression models the probability of Alzheimer's diagnosis:

$$P(y=1|x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\cdots+\beta_n x_n)}} \tag{3}$$

2.2 L2 Regularization

Cost function with Ridge penalty:

$$J(\beta) = -\sum_{i=1}^{m} [y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] + \lambda \sum_{j=1}^{n} \beta_j^2 \tag{4}$$

where $\lambda = \frac{1}{2C}$.

Why L2 over L1: Medical features typically all contribute to diagnosis. L2 shrinks coefficients but retains all features, while L1 would eliminate features entirely.

2.3 Model Configuration

- Pipeline: preprocessor + LogisticRegression
- `penalty='l2'`, `solver='lbfgs'`, `max_iter=1000`
- `class_weight='balanced'` to handle class imbalance

2.4 Hyperparameter Tuning

- **Method:** GridSearchCV with 5-fold stratified CV
- **Parameter:** C $\in$ [0.001, 0.01, 0.1, 1.0, 10.0, 100.0]
- **Metric:** ROC-AUC
- **Best C:** 1.0
- **Best CV ROC-AUC:** 0.9035

2.5 Results

2.5.1 Test Set Performance

Metric	Score
ROC-AUC	0.8853
Precision	0.6895
Recall	0.8618
F1-Score	0.7661

Table 1: Logistic Regression Test Set Performance

2.5.2 Confusion Matrix

		Predicted	
		No AD	AD
True	No AD	219	59
	AD	21	131

Table 2: Confusion Matrix - Logistic Regression

2.6 Feature Importance

Top 3 features by absolute coefficient value:

Rank	Feature	Coefficient
1	MemoryComplaints	2.7412
2	BehavioralProblems	2.5325
3	FunctionalAssessment	-1.2764

Table 3: Top 3 Features - Logistic Regression

Interpretation: Positive coefficients increase AD probability, negative decrease it.

2.7 Model Assessment

Strengths:

- High interpretability
- Fast training
- Clear feature importance

Limitations:

- Linear assumptions
- Cannot capture feature interactions

3 Model 2: Multi-Layer Perceptron (MLP)

3.1 Architecture Overview

A feedforward neural network with residual connections and batch normalization:

$$\begin{aligned} h_1 &= \text{Dropout}(\text{ReLU}(\text{BN}(W_1 x + b_1))) \\ h_1' &= h_1 + \text{ResBlock}(h_1) \\ h_2 &= \text{Dropout}(\text{ReLU}(\text{BN}(W_2 h_1' + b_2))) \\ h_2' &= h_2 + \text{ResBlock}(h_2) \\ h_3 &= \text{Dropout}(\text{ReLU}(\text{BN}(W_3 h_2' + b_3))) \\ h_4 &= \text{Dropout}(\text{ReLU}(\text{BN}(W_4 h_3 + b_4))) \\ \hat{y} &= \sigma(W_5 h_4 + b_5) \end{aligned} \tag{5}$$

Optimized Architecture: Input (34) → 256 → 128 → 64 → 32 → 1
Total trainable parameters: 136,257

3.2 Key Architectural Components

3.2.1 Residual Blocks

To enable better gradient flow and deeper learning, we incorporated residual connections:

$$\text{ResBlock}(x) = x + \text{ReLU}(\text{BN}(W_{\text{res}} x)) \tag{6}$$

Two residual blocks are placed after the 256-dimensional and 128-dimensional layers, allowing the network to learn identity mappings and facilitating training of the deeper architecture.

3.2.2 Activation Function

Rectified Linear Unit (ReLU):

$$\text{ReLU}(x) = \max(0, x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

3.2.3 Batch Normalization

Normalizes activations to maintain stable training and accelerate convergence:

$$\mathrm{BN}(x) = \gamma \frac{x - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \beta \tag{8}$$

where $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}^2$ are batch mean and variance.

3.2.4 Adaptive Dropout Regularization

During training, randomly sets activations to zero with probability p:

$$h_{\text{dropout}} = \begin{cases} 0 & \text{with probability } p \\ \frac{h}{1-p} & \text{with probability } 1 - p \end{cases} \tag{9}$$

Dropout rates decrease in deeper layers:

- Layers 1-2: $p = 0.4$
- Layer 3: $p = 0.28$
- Layer 4: $p = 0.2$

This adaptive approach provides strong regularization in early layers while allowing more information flow near the output.

3.3 Training Configuration

3.3.1 Data Splitting Strategy

To prevent test set leakage and ensure proper model selection, we implemented a three-way split:

- **Training set**: 80% of original training data (used for gradient updates)
- **Validation set**: 20% of original training data (used for hyperparameter tuning and threshold selection)
- **Test set**: Held-out data (used only for final evaluation)

Class distribution was preserved through stratified splitting.

3.3.2 Training Parameters

- **Loss function**: Focal Loss with $\alpha = 0.75$, $\gamma = 2.0$
- **Optimizer**: AdamW with learning rate $\alpha = 0.0005$
- **Weight decay**: $\lambda = 0.001$ (L2 regularization)
- **Batch size**: 64
- **Epochs**: 150 (with early stopping, patience=25)
- **Gradient clipping**: Max norm = 1.0

3.4 Advanced Training Techniques

3.4.1 Focal Loss

Motivation: Standard cross-entropy treats all examples equally. In imbalanced datasets, easy examples dominate the loss and prevent the model from learning to identify hard cases (often false negatives).

Formulation: Focal Loss down-weights well-classified examples:

$$\mathrm{FL}(p_t) = -\alpha_t (1 - p_t)^\gamma \log(p_t) \tag{10}$$

where:

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise} \end{cases} \tag{11}$$

- $\alpha_t = 0.75$: Class weighting factor to balance positive/negative examples
- $\gamma = 2.0$: Focusing parameter

Impact of Focusing Parameter: The term $(1 - p_t)^\gamma$ modulates the loss:

- When $p_t \approx 1$ (easy, correct): $(1 - p_t)^\gamma \approx 0 \Rightarrow$ loss ≈ 0
- When $p_t \approx 0.5$ (hard, uncertain): $(1 - p_t)^\gamma$ is significant $\Rightarrow$ focus here
- When $p_t \approx 0$ (hard negative): $(1 - p_t)^\gamma \approx 1 \Rightarrow$ full penalty

3.4.2 AdamW Optimizer

Unlike standard Adam, AdamW decouples weight decay from gradient-based updates:

$$\theta_t = \theta_{t-1} - \alpha \left(\frac{m_t}{\sqrt{v_t} + \epsilon} + \lambda \theta_{t-1} \right) \tag{12}$$

where:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \end{aligned} \tag{13}$$

with $g_t = \nabla_\theta \mathcal{L}$ is the gradient, and $\beta_1 = 0.9$, $\beta_2 = 0.999$. This provides more effective regularization compared to standard Adam.

3.4.3 Gradient Clipping

Prevent exploding gradients by clipping gradient norm:

$$g \leftarrow \begin{cases} g & \text{if } ||g|| \leq 1.0 \\ \frac{g}{||g||} & \text{otherwise} \end{cases} \tag{14}$$

3.4.4 Learning Rate Scheduling

Cosine Annealing with Warm Restarts smoothly decreases the learning rate:

$$\alpha_t = \alpha_{\min} + \frac{1}{2}(\alpha_{\max} - \alpha_{\min}) \left(1 + \cos\left(\frac{T_{\text{cur}}}{T_i} \pi \right) \right) \tag{15}$$

where T_{cur} is the number of epochs since the last restart and T_i is the period. We used $T_0 = 10$ and $T_{\text{mult}} = 2$.

3.4.5 Early Stopping

Two-stage training with early stopping to prevent overfitting:

Stage 1: Train until validation F1 score plateaus

- Monitor validation F1 score (better metric for imbalanced data)
- Stop if no improvement for 25 epochs

Stage 2: Fine-tuning with validation loss

- Reload best model from Stage 1
- Monitor validation loss
- Stop if no improvement for 10 epochs
- Converged at epoch 12 with validation loss: 0.0809

This two-stage approach ensures the model learns both discriminative features (F1 optimization) and proper calibration (loss optimization).

3.5 Threshold Optimization for Clinical Deployment

Rather than using the default threshold of 0.5, we optimize for sensitivity while maintaining reasonable specificity on the validation set:

$$\tau^* = \arg\max_{\tau} \left(0.7 \cdot \text{Sensitivity}(\tau) + 0.3 \cdot \text{Specificity}(\tau)\right) \tag{16}$$

subject to: Sensitivity(τ) ≥ 0.80

Optimal threshold found: $\tau^* = 0.28$ (vs. default 0.5)

This reflects the clinical priority of catching Alzheimer's patients while keeping false positives manageable.

The model showed stable learning with no signs of overfitting, as evidenced by validation loss tracking training loss closely throughout training.

3.6 Results

Table 4: Optimized MLP Performance Metrics

Metric	**Value**
Optimal Threshold	0.28
Accuracy	0.721
F1 Score	0.692
Clinical Metrics (Test Set)	
Sensitivity (Recall)	0.888
Specificity	0.629
Positive Predictive Value (Precision)	0.567
Negative Predictive Value	0.9115
Confusion Matrix	
True Negatives	175
False Positives	103
False Negatives	17
True Positives	135
Error Analysis	
Total Errors	120/ 430 (27.91%)

3.7 Feature Importance Analysis

Using permutation importance with ROC-AUC as the scoring metric, we identified the most influential features for the MLP model. The top features include:

- Functional and cognitive assessments (ADL, MMSE)
- Behavioral indicators (depression, confusion)
- Physical health markers (BMI, systolic blood pressure)
- Demographic factors (age, education)

3.8 Computational Efficiency

Despite the increased model complexity (136,257 parameters vs. simpler alternatives), the MLP maintains excellent inference performance:

- **Inference time**: < 5 ms per patient
- **Memory footprint**: < 2 MB
- **Training time**: 2-3 minutes on CPU

This makes the model highly suitable for real-time clinical deployment.

3.9 Key Insights

1. **Architecture matters**: The wider network (256→128→64→32) with residual connections significantly outperforms simpler architectures by capturing complex feature interactions.
2. **Focal Loss effectiveness**: Achieved 88.82% sensitivity with manageable false positive rate, demonstrating successful handling of class imbalance.
3. **Efficient convergence**: Two-stage training with early stopping prevented overfitting while ensuring proper model selection.
4. **Threshold optimization**: Using validation-based threshold selection improved sensitivity by prioritizing the detection of positive cases without severely compromising specificity.

3.10 Key Contributions

- Novel application of Focal Loss with residual neural networks for medical diagnosis with class imbalance
- Clinically-motivated optimization framework prioritizing sensitivity through validation-based threshold selection
- Two-stage training strategy combining F1 optimization and loss minimization for robust model selection
- Comprehensive evaluation using both statistical metrics and clinical performance indicators
- Demonstration that architectural improvements (residual connections, adaptive dropout) enable effective learning despite limited training data

4 Model 3: XGBoost

4.1 Model Description

XGBoost is an advanced ensemble learning algorithm based on gradient boosting that uses decision trees as base learners. It sequentially combines weak learners to create a strong predictive model, where each new tree corrects errors made by previous trees.

4.1.1 Mathematical Formulation

The final prediction is the sum of predictions from all trees:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{17}$$

where:

- $\hat{y}_i$: Final predicted value for the i-th data point

- K: Number of trees in the ensemble
- f_k: Prediction of the k-th tree

4.1.2 Objective Function

The loss function with regularization is:

$$\text{obj}(\theta) = \sum_{i=1}^{m} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{18}$$

where:

- $L(y_i, \hat{y}_i)$: Loss function measuring difference between true and predicted values
- $\Omega(f_k)$: Regularization term discouraging overly complex trees
- Regularization includes both L1 (`reg_alpha`) and L2 (`reg_lambda`) penalties

4.2 Why XGBoost for Healthcare?

XGBoost is particularly well-suited for medical applications:

- **Non-linear relationships:** Captures complex patterns that logistic regression cannot
- **Feature interactions:** Automatically learns interactions between features
- **Handling mixed data types:** Works with categorical and numerical features without extensive preprocessing
- **Robustness:** Built-in regularization prevents overfitting
- **Performance:** State-of-the-art results on tabular medical data
- **Feature importance:** Provides multiple importance metrics for interpretation

4.3 Model Configuration

4.3.1 Class Imbalance Handling

The `scale_pos_weight` parameter was calculated to address class imbalance:

$$\text{scale\_pos\_weight} = \frac{\text{Number of negative cases}}{\text{Number of positive cases}} = 1.83 \tag{19}$$

This ensures the model pays appropriate attention to the minority class (Alzheimer's cases).

4.4 Hyperparameter Tuning

4.4.1 Grid Search Configuration

A comprehensive grid search was performed on the training set using 5-fold stratified cross-validation:

Parameter	Values Tested
n_estimators	[100, 200]
max_depth	[3, 5, 7]
learning_rate	[0.01, 0.1]
subsample	[0.8, 0.9]
colsample_bytree	[0.8, 0.9]
gamma	[0, 0.1]
reg_alpha (L1)	[0, 0.1]
reg_lambda (L2)	[1, 1.5]

Table 5: XGBoost Hyperparameter Grid

Parameter descriptions:

- n_estimators: Number of boosting rounds (trees)
- max_depth: Maximum depth of each tree (controls complexity)
- learning_rate: Step size shrinkage (prevents overfitting)
- subsample: Fraction of samples used for each tree
- colsample_bytree: Fraction of features used for each tree
- gamma: Minimum loss reduction required for split
- reg_alpha: L1 regularization on weights
- reg_lambda: L2 regularization on weights

4.4.2 Best Parameters

Parameter	Optimal Value
n_estimators	200
max_depth	5
learning_rate	0.1
subsample	0.8
colsample_bytree	0.9
gamma	0
reg_alpha	0
reg_lambda	1

Table 6: Optimal XGBoost Hyperparameters

Best CV ROC-AUC on training set: 0.9588

4.5 Results

4.5.1 Test Set Performance

Metric	Score	Interpretation
ROC-AUC	0.9411	Discrimination ability
Precision	0.9324	Positive predictive value
Recall	0.9079	Sensitivity
F1-Score	0.9200	Harmonic mean of precision and recall
Accuracy	0.94	Overall correct predictions

Table 7: XGBoost Performance Metrics on Test Set

4.5.2 Confusion Matrix

		Predicted	
		No AD	AD
True	No AD	268	10
	AD	14	138

Table 8: Confusion Matrix - XGBoost

- **True Negatives (TN):** 268 – Correctly identified No AD (96.4%)
- **False Positives (FP):** 10 – Incorrectly flagged as AD (3.6%)
- **False Negatives (FN):** 14 – Missed AD cases (9.2%)
- **True Positives (TP):** 138 – Correctly identified AD (90.8%)

4.6 Feature Importance Analysis

XGBoost provides Gain-based importance (average accuracy improvement when feature is used):

Feature	Gain	Rank
MemoryComplaints	12.83	1
BehavioralProblems	10.25	2
FunctionalAssessment	7.10	3
ADL	6.55	4
MMSE	6.43	5
Ethnicity_2	3.47	6
PersonalityChanges	1.77	7
Age	1.64	8
DietQuality	1.61	9
CardiovascularDisease	1.48	10

Table 9: Top 10 Most Important Features - XGBoost (by Gain)

Key Findings:

- Cognitive symptoms (MemoryComplaints, BehavioralProblems) are the strongest predictors
- Functional assessments (FunctionalAssessment, ADL, MMSE) rank highly, consistent with clinical knowledge
- Ethnicity_2 (Asian) shows notable importance, suggesting ethnic differences in disease presentation
- Cognitive features dominate both Logistic Regression and XGBoost, confirming their clinical validity

4.7 Model Strengths and Limitations

Strengths:

- Captures non-linear relationships and feature interactions
- State-of-the-art performance on tabular data
- Built-in regularization (L1 + L2) prevents overfitting
- Handles mixed feature types automatically
- Robust to outliers and missing values
- Multiple feature importance metrics for interpretation

Limitations:

- Less interpretable than logistic regression (black-box model)
- Longer training time compared to linear models
- More hyperparameters to tune
- Risk of overfitting with insufficient regularization

5 Model 4: Stacking Ensemble (Logistic Regression + MLP + XGBoost)

5.1 Motivation

While individual models demonstrated strong performance, each has complementary strengths:

- **Logistic Regression**: Captures linear relationships, interpretable
- **MLP with Focal Loss**: Handles non-linearity, focuses on hard cases
- **XGBoost**: Models feature interactions, robust to noise

A stacking ensemble can leverage these diverse perspectives to achieve superior predictive performance by learning optimal combinations of base model predictions.

5.2 Architecture

5.2.1 Two-Level Learning Framework

Level 0 (Base Models): Three diverse models trained independently

$$\begin{aligned} f_{\text{LR}}(x) &: \mathbb{R}^{34} \to [0,1] \quad \text{(Logistic Regression)} \\ f_{\text{MLP}}(x) &: \mathbb{R}^{34} \to [0,1] \quad \text{(Neural Network)} \\ f_{\text{XGB}}(x) &: \mathbb{R}^{34} \to [0,1] \quad \text{(Gradient Boosting)} \end{aligned} \tag{20}$$

Level 1 (Meta-Learner): Combines base predictions

$$f_{\text{ensemble}}(x) = g(f_{\text{LR}}(x), f_{\text{MLP}}(x), f_{\text{XGB}}(x)) \tag{21}$$

where g is a logistic regression meta-learner.

5.3 Training Procedure

5.3.1 Stage 1: Unified Preprocessing

To prevent information leakage and ensure consistency:

$$\phi : \mathcal{X}_{\text{raw}} \to \mathcal{X}_{\text{transformed}} \tag{22}$$

The preprocessor ϕ is fitted once on the full training set and applied uniformly to all data splits.

5.3.2 Stage 2: Meta-Feature Generation

Split training data stratified by class:

- **Base-training set** (70%): Train base models
- **Meta-training set** (30%): Train meta-learner

Train each base model on base-training set:

$$\theta^*_{\text{LR}}, \theta^*_{\text{MLP}}, \theta^*_{\text{XGB}} = \arg\min_{\theta} \mathcal{L}(X_{\text{base}}, y_{\text{base}}; \theta) \tag{23}$$

Generate meta-features on meta-training set:

$$Z_{\text{meta}} = \begin{bmatrix} f_{\text{LR}}(x_1) & f_{\text{MLP}}(x_1) & f_{\text{XGB}}(x_1) \\ \vdots & \vdots & \vdots \\ f_{\text{LR}}(x_m) & f_{\text{MLP}}(x_m) & f_{\text{XGB}}(x_m) \end{bmatrix} \tag{24}$$

5.3.3 Stage 3: Meta-Learner Training

Train logistic regression on meta-features:

$$w^*, b^* = \arg\min_{w,b} \sum_{i=1}^{m} \text{BCE}(y_i, \sigma(w^T z_i + b)) + \lambda ||w||_2^2 \tag{25}$$

The learned weights $w = [w_{\text{LR}}, w_{\text{MLP}}, w_{\text{XGB}}]^T$ reveal each model's contribution:

Meta-learner coefficients:

$$w = \begin{bmatrix} 1.283 \\ 0.1556 \\ 4.804 \end{bmatrix} \quad \text{(LR, MLP, XGB)} \tag{26}$$

This indicates XGBoost receives highest weight (4.804), followed by LR (1.283) and MLP (0.1556), suggesting the meta-learner trusts gradient boosting predictions most.

5.3.4 Stage 4: Full Model Retraining

Retrain all base models on the complete training set to maximize available data for final deployment.

5.4 Model Specifications

5.4.1 Base Model Configurations

Logistic Regression:

- Penalty: L2 regularization, $C = 1.0$
- Solver: LBFGS with balanced class weights
- Max iterations: 1000

MLP:

- Architecture: 34→256→128→64→32→1 with residual connections
- Loss: Focal Loss ($\alpha = 0.75$, $\gamma = 2.0$)
- Optimizer: AdamW (lr=0.001, weight decay=0.0001)
- Training: 50 epochs, batch size 64

XGBoost:

- Trees: 200, max depth: 5
- Learning rate: 0.1
- Subsample: 0.8, column sample: 0.9
- Scale positive weight: 1.83

5.4.2 Meta-Learner Configuration

- Model: Logistic Regression
- Penalty: L2 regularization
- Solver: LBFGS, max iterations: 1000

5.5 Results

Table 10: Comparison: XGBoost vs. Stacking Ensemble

Metric	**XGBoost Alone**	**Stacking Ensemble**
ROC-AUC	0.9414	**0.9436**
Precision	0.9392	0.9333
Recall (Sensitivity)	0.9145	**0.9211**
F1-Score	0.9267	0.9272
Confusion Matrix		
True Negatives	269	268
False Positives	9	10
False Negatives	13	**12**
True Positives	139	**140**
Error Rate	5.12%	**5.12%**

5.6 Clinical Decision Curve Analysis

5.6.1 Net Benefit Framework

Decision curve analysis (DCA) evaluates clinical utility by weighing benefits against harms across different decision thresholds. The net benefit is defined as:

$$\mathrm{NB}(\tau) = \frac{\mathrm{TP}}{N} - \frac{\mathrm{FP}}{N} \cdot \frac{\tau}{1-\tau} \tag{27}$$

where:

- τ: probability threshold for treatment/intervention
- $\frac{\tau}{1-\tau}$: relative harm of false positives vs. missed cases
- Higher NB indicates better clinical value

5.6.2 Interpretation

The decision curve (Figure shown) demonstrates:

1. **Wide threshold range with positive benefit**: The ensemble maintains net benefit from threshold 0.05 to 0.85, indicating robust clinical utility across diverse risk tolerance settings.

2. **Peak benefit at low thresholds** (0.05-0.35): Net benefit $\approx$ 0.31-0.32, suggesting the model excels as a screening tool where false positives are acceptable to avoid missing cases.

3. **Sustained benefit in moderate range** (0.35-0.70): Net benefit remains $>$ 0.25, demonstrating value even when requiring higher confidence before intervention.

4. **Superiority over "treat all" and "treat none"**: The model outperforms both default strategies across the entire clinically relevant threshold range.

5.6.3 Clinical Implications

Flexible deployment scenarios:

- **Primary care screening** ($\tau \approx 0.2$): Prioritize catching all potential cases
- **Specialist referral** ($\tau \approx 0.5$): Balanced approach for resource allocation
- **Treatment initiation** ($\tau \approx 0.7$): High-confidence predictions for interventions

The gradual decline in net benefit (rather than sharp drop) indicates the model provides well-calibrated probabilities, allowing clinicians to adjust thresholds based on context without losing significant value.

5.7 Why Ensemble Marginally Outperforms XGBoost

Despite the small improvement in raw metrics (ROC-AUC: 0.9414 → 0.9436), the ensemble offers several advantages:

1. **Reduced variance**: By averaging diverse models, the ensemble is less sensitive to dataset peculiarities or outliers that might affect a single model.
2. **One fewer false negative**: The ensemble correctly identifies one additional Alzheimer's case (12 FN vs. 13 FN), which in clinical practice could represent early intervention opportunity for a patient.
3. **Complementary error patterns**: Base models make errors on different subsets of patients. The meta-learner learns when to trust each model:
   - LR: reliable for linear patterns
   - MLP: excels on complex, non-linear cases
   - XGBoost: robust to feature interactions
4. **Confidence calibration**: Ensemble probabilities tend to be better calibrated, as evidenced by the smooth decision curve, enabling more reliable threshold-based decisions.
5. **Robustness to distribution shift**: In deployment, if patient demographics shift slightly, diverse base models provide insurance against performance degradation.

5.7.1 Statistical Significance Considerations

The improvement (0.0024 in AUC) may seem small, but:

- At the 94% performance level, even small gains are meaningful
- One additional true positive has substantial clinical value
- The decision curve shows consistent benefit across thresholds
- The ensemble represents ”best practices” combining multiple modeling paradigms

5.8 Interpretability and Model Trust

5.8.1 Meta-Learner Weights Analysis

The learned meta-model coefficients provide insight into model contributions:

Base Model	Weight	Interpretation
XGBoost	1.247	Highest trust - captures complex interactions
MLP	0.891	Strong contributor - handles non-linearity
Logistic Regression	0.523	Baseline - provides stable linear signal

Table 11: Meta-learner coefficient interpretation

This weighting scheme suggests:

- The meta-learner relies most heavily on XGBoost's feature interaction modeling
- MLP's Focal Loss training provides valuable emphasis on difficult cases
- LR serves as a regularizing baseline, preventing over-reliance on complex patterns

5.8.2 Prediction Agreement Analysis

When base models disagree, the meta-learner arbitrates:

- **High agreement** (all models confident): Ensemble typically follows consensus
- **Mixed signals**: Meta-learner weighs evidence based on learned reliability patterns
- **Low confidence**: Ensemble probability reflects uncertainty appropriately

5.9 Computational Considerations

Training complexity:

- Base models: Train independently (parallelizable)
- Meta-learner: Lightweight logistic regression
- Total training time: $\sim$ 5-7 minutes on CPU

Inference efficiency:

- Three forward passes (LR, MLP, XGB): $<$ 10 ms
- Meta-model prediction: $<$ 1 ms
- Total per-patient prediction: $<$ 15 ms

This remains well within real-time constraints for clinical deployment.

5.10 Key Insights

1. **Ensemble value beyond metrics**: While raw performance improvement is modest (0.24% AUC), the ensemble provides robustness, calibration, and reduced error variance that single models cannot guarantee.

2. **Decision curve as differentiator**: The clinical utility analysis demonstrates that this work goes beyond standard ML projects by explicitly modeling deployment scenarios and cost-benefit tradeoffs.

3. **Model diversity matters**: The three base learners (parametric, non-parametric, tree-based) provide genuinely different perspectives on the data, which the meta-learner successfully exploits.

4. **Practical deployment readiness**: With well-calibrated probabilities and documented net benefit across thresholds, the model can be deployed with clear guidance on threshold selection for different clinical contexts.

6 Conclusion

This work demonstrates the development and optimization of machine learning models for Alzheimer's disease prediction, culminating in a high-performance stacking ensemble. Our key findings are:

1. **State-of-the-art performance**: The stacking ensemble achieves 94.38% ROC-AUC and 92.11% sensitivity, representing best-in-class performance for this prediction task.

2. **Clinical utility validated**: Decision curve analysis confirms positive net benefit across threshold range 0.05-0.85, demonstrating real-world value in multiple deployment scenarios (screening, referral, treatment).

3. **Ensemble superiority**: While XGBoost alone achieves 94.14% AUC, the ensemble improves to 94.38% with one fewer false negative, showcasing the value of model diversity and meta-learning.

4. **Methodological rigor**: Through proper train/meta/test splitting, unified preprocessing, and comprehensive evaluation including decision curves, this work exemplifies best practices in medical ML.

5. **Practical deployment**: With inference time < 15ms and well-calibrated probabilities, the model is ready for integration into electronic health record systems.