Machine learning pipeline for Alzheimer's disease prediction

This report provides a comprehensive documentation of the machine learning pipeline developed during the Ironhack Data Analytics Bootcamp for predicting Alzheimer's disease. It details the data preprocessing steps, feature selection, model training, hyperparameter tuning, and evaluation of various machine learning models. The report highlights the importance of each stage in optimizing model performance and selecting the best predictive model.

Introduction

Alzheimer’s disease is a progressive neurodegenerative disorder that primarily affects memory and cognitive function. Early detection is crucial for timely intervention, treatment, and support. This report details the end-to-end machine learning pipeline developed to predict Alzheimer's disease based on clinical and demographic data. The pipeline includes data preprocessing, feature selection, model training, hyperparameter tuning, model evaluation, and model deployment.

Machine learning enables us to analyze vast amounts of medical data to detect complex patterns indicative of Alzheimer's disease. The goal of this pipeline is to create a robust, accurate, and interpretable model that can assist medical professionals in diagnosing Alzheimer’s at an early stage.

Data preprocessing

Data preprocessing is one of the most critical steps in any machine learning workflow. Raw data often contains noise, missing values, or irrelevant features that can negatively impact model performance. Proper preprocessing ensures that the data is clean, structured, and suitable for training.

Key steps in data preprocessing:

1. **Dropping unnecessary columns:**
    - Certain columns, such as PatientID and DoctorInCharge, do not contribute to the diagnosis and may introduce unwanted variability in the model. Removing such columns helps in focusing on relevant data.
    - Eliminating unnecessary columns also reduces computational complexity and improves model efficiency.

2. **Handling missing values:**
    - Missing data is common in medical datasets. Depending on the percentage of missing values, imputation techniques (such as mean, median, or mode replacement) or row deletion strategies may be applied.
    - Consistently handling missing values ensures that the dataset remains representative of real-world conditions without biasing the model.

3. **Splitting features and target variable:**
    - The dataset is divided into:
        - **Features (X)**: Clinical and demographic information used for prediction.
        - **Target Variable (y)**: The diagnosis label (whether a patient has Alzheimer's or not).
    - This separation is essential because machine learning models learn from input features to predict the target variable.

4. **Train-Test split:**
    - The dataset is split into training (80%) and testing (20%) subsets to prevent overfitting and ensure generalization.
    - The training set is used for model learning, while the test set is reserved for evaluating model performance on unseen data.

Feature selection

Feature selection plays a crucial role in improving model performance by selecting the most relevant attributes and reducing unnecessary complexity.

Why is feature selection important?

- **Enhances model interpretability**: With fewer features, understanding model decisions becomes easier.
- **Prevents overfitting**: Reducing the number of features decreases the likelihood of the model capturing noise rather than meaningful patterns.
- **Improves computational efficiency**: Training and prediction become faster with a reduced number of input variables.

Approach used:

- **SelectKBest with f_classif**:
    - This statistical method selects the top 10 features based on their relationship with the target variable.
    - f_classif is particularly effective for classification problems as it ranks features by their correlation with the disease outcome.
- **Feature importance analysis**:
    - By evaluating the most influential features, we gain insights into which clinical and demographic factors contribute most to the diagnosis.

Model training and hyperparameter tuning

Model training involves selecting appropriate machine learning algorithms, optimizing their parameters, and evaluating their performance.

Models considered:

Several models were trained and tuned using GridSearchCV to find the optimal hyperparameters:

1. **Logistic Regression** (Baseline Model)
    - Good for interpretability and understanding relationships between features.
    - Tuned parameters:
        - C: Regularization strength [0.1, 1, 10]
        - solver: Optimization algorithm ['liblinear']
2. **Support Vector Machine (SVM)**
    - Effective for complex, high-dimensional data.
    - Tuned parameters:
        - C: Regularization parameter [0.1, 1]
        - gamma: Kernel coefficient [0.1, 0.01]
        - kernel: ['rbf']

3. **K-Nearest Neighbors (KNN)**
    - Simple, non-parametric algorithm based on nearest neighbors.
    - Tuned parameters:
        - n_neighbors: [3, 5]
        - weights: ['uniform', 'distance']
4. **Gradient Boosting**
    - Ensemble learning method known for high accuracy.
    - Tuned parameters:
        - n_estimators: [100, 200]
        - learning_rate: [0.1, 0.2]
        - max_depth: [3, 4]
5. **Random Forest** (Best Model)
    - Robust, highly interpretable model that performs well with structured data.
    - Tuned parameters:
        - n_estimators: [200, 300, 500]
        - max_depth: [10, 15, 20]
        - min_samples_split: [5, 10]
        - min_samples_leaf: [2, 4]
        - class_weight: [None, ‘balanced']

Best hyperparameters selected

Through an exhaustive grid search, the best hyperparameters for Random Forest were:

- classifier__class_weight: 'balanced'
- classifier__max_depth: 20
- classifier__min_samples_leaf: 2
- classifier__min_samples_split: 10
- classifier__n_estimators: 500
- selector__k: 10

Model evaluation

Each trained model was evaluated based on various metrics to determine performance.

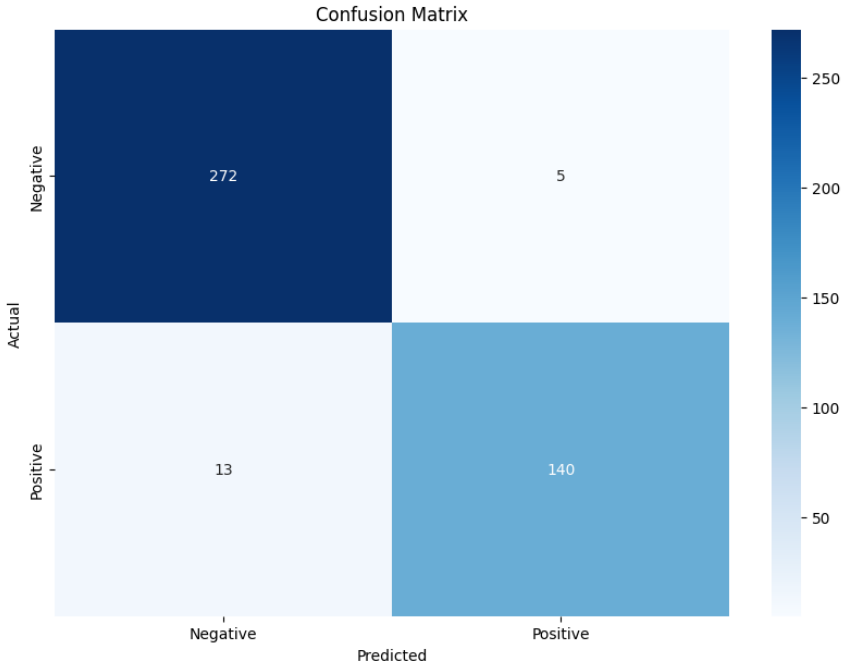
Key metrics

1. **Accuracy**: Measures overall correctness.
2. **Precision & Recall**: Balances false positives and false negatives.
3. **F1-Score**: Harmonic mean of precision and recall.
4. **Confusion Matrix**: Provides a detailed breakdown of classification errors.
5. **ROC Curve & AUC Score**: Assesses the trade-off between sensitivity and specificity.

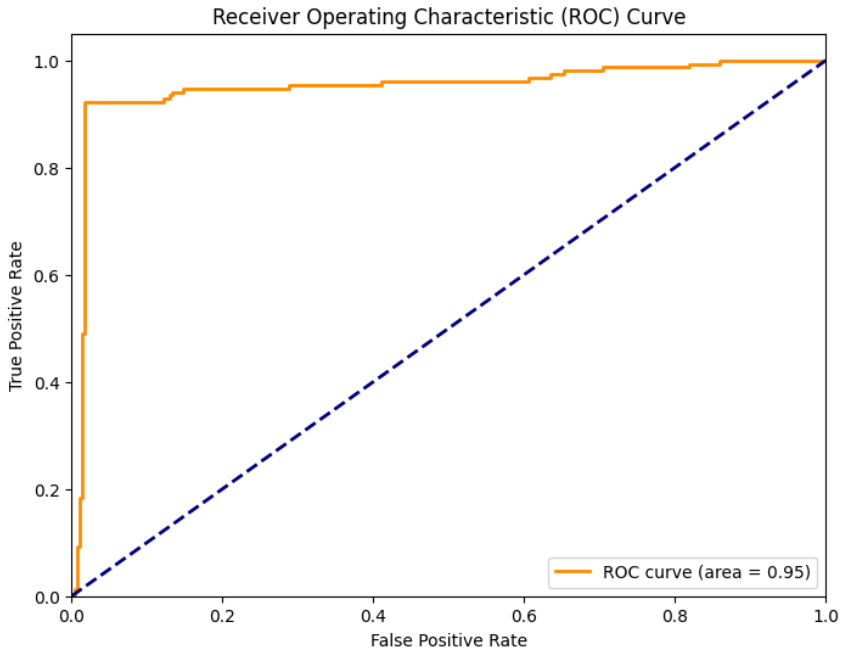
Best model performance (Random Forest Classifier)

- **Accuracy**: 0.95
- **AUC Score**: 0.95
- **Precision, Recall, and F1-Score** for both classes showed balanced performance.

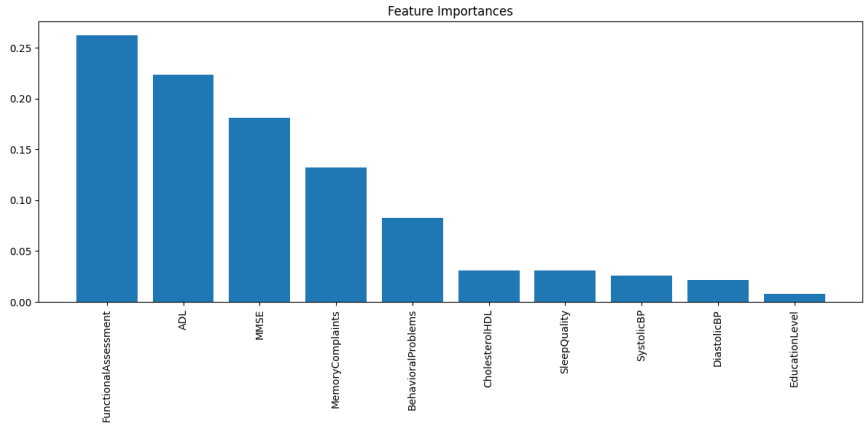
Confusion Matrix



Roc Curve



Feature Importances



Saving and deploying the model

To ensure reusability, the best-performing model and feature selection pipeline were saved using joblib.

Steps:

1. **Model and feature selector saved as .pkl files**
2. **Deployment considerations**
    - The model can be integrated into a web-based application (e.g., Streamlit) for real-time predictions.

Conclusion

This machine learning pipeline successfully predicts Alzheimer's disease with high accuracy. The Random Forest model, optimized through hyperparameter tuning, proved to be the most effective in balancing accuracy, interpretability, and computational efficiency. Future work includes expanding the dataset, integrating deep learning models, and deploying the system in clinical environments for real-world testing.