Predicting Alzheimer's Disease Diagnosis Using Random Forest and Hyperparameter Tuning

Title of the Project : Predicting Alzheimer's Disease Diagnosis using Random Forest and Hyper tuning

Student Name : Bhagyasri Rajarajeswari Devi Sadanala

Roll Number : 2347390700

Institution : Aditya Women’s Degree College, Kakinada

Department Name : BCA-Data Science

Submission Date : 4-02-2025

The aim of the project is to predict the diagnosis of Alzheimer's disease by using Super-Visied machine learning techniques especially RandomForest and Hypertuning. The data set consist of various features related to cognitive health such as test scores and medical history, which are used to predict whether a person is diagnosed with Alzheimer's disease or not. After Pre-Processing and Feature Engineering, I have choosen the Random Forest Model for the training of the data set and optimised using RandomizedSearchCv to identified the best parameters. Then the models performance is evaluated through cross-validation and accuracy metrics. By the result we can say that machine learning algorithms can accurately diagnose Alzheimer's disease by analysing key variables.This project demonstrates the potential of Machine learning models in healthcare, especially for the early detection of the disease Alzheimer's disease.

1.Introduction

Statement of the Project

Alzheimer's disease is a brain disorder that slowly destroys memory and thinking skills. It's the most common type of dementia in older adults. Early detection of this disease is crucial for the better treatment and management. Traditional methods of diagnosing Alzheimer's often involve manual analysis of medical data, which can be time-consuming and error-prone. The use of Machine learning models offers early detection of the disease by automating the diagnosis process. The main aim of this project is to explore the possibility of predicting Alzheimer's disease based on a set of medical and cognitive features using machine learning.

Objective of the Study

The main objective of this project is to develop a machine learning model that can predict the presence of Alzheimer's Disease based on various clinical features. The project mainly focuses on using the Random Forest classifier and applying hyperparameter optimization techniques such as RandomizedSearchCV to enhance model performance.

2.Framework and Methodology

Overview of Machine Learning Approach

This project utilizes supervised machine learning approach, where the goal is to predict a categorical outcome (Diagnosis: Alzheimer's or Not) using the input features. The process involves the data preprocessing, model selection, training, and evaluation. And then the Random Forest classifier is chosen for its ability to

handle both numerical and categorical data and its robustness against overfitting.

Data Preprocessing

The dataset has been cleaned by handling missing values, encoding categorical features, and scaling numerical features using StandardScaler. The features are then split into training and testing datasets, and model evaluation is done using accuracy and classification metrics.

Model Selection: Random Forest

The algorithm Random Forest is chosen due to its interpretability, ability to handle large datasets, and efficient handling of feature interactions. It builds multiple decision trees and averages their results to improve prediction accuracy.

Hyperparameter Tuning with RandomizedSearchCV

To optimize the Random Forest model, a RandomizedSearchCV approach is used,

which performs a randomized search over a specified hyperparameter space to identify the best model parameters.

3.Implementation and Exceution

Data Exploration

The dataset is explored by examining the summary statistics, data types, and the distribution of various features. Visualizations such as histograms and correlation matrices are generated to understand the relationships between different features.

Model Training and Evaluation

The Random Forest classifier is trained using the pre-processed data. Hyperparameter optimization is performed using RandomizedSearchCV to find the best combination of parameters. The model's performance is evaluated using

accuracy, precision, recall, F1-score, and a classification report.

Code Implementation

The implementation involves importing necessary libraries (e.g., pandas, scikit-learn, seaborn, matplotlib), loading the dataset, preprocessing it, plotting each and every column with its frequency, correlation matrix for the dataset is plotted, splitting into training and testing and applying RandomizedSearchCV to the dataset. The best model is evaluated on the tested data and feature importance is extracted.

```
[1]  import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     from sklearn.model_selection import train_test_split
     from sklearn.ensemble import RandomForestClassifier
     from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
     from sklearn.model_selection import RandomizedSearchCV
```

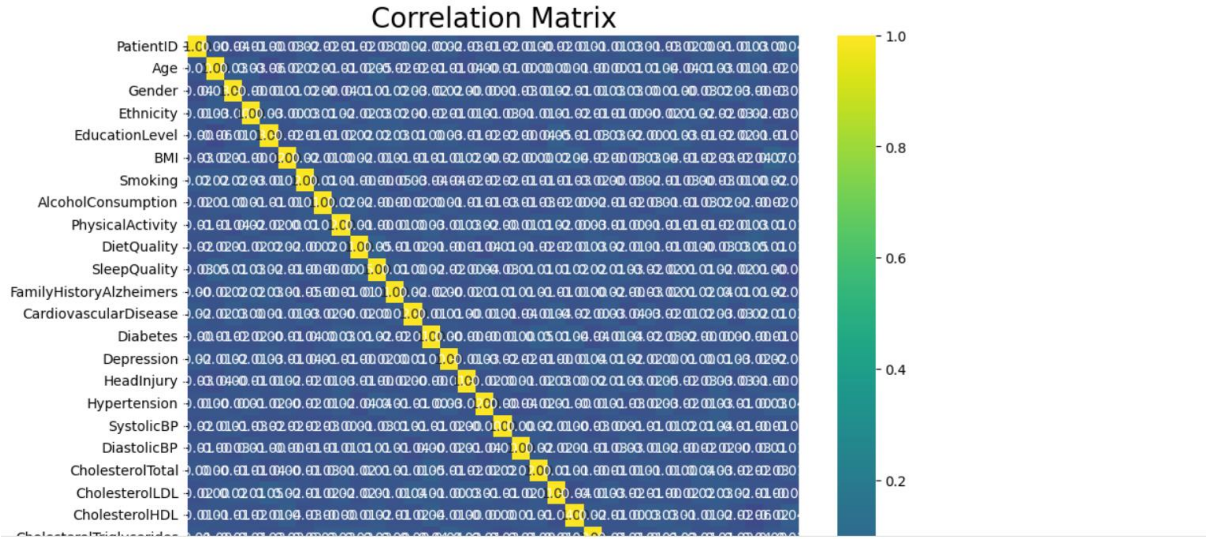
```python
from sklearn.preprocessing import StandardScaler
X = df.drop(columns=["DoctorInCharge", "Diagnosis", "Forgetfulness", "DifficultyCompletingTasks",
                     "Hypertension", "Confusion", "HeadInjury", "Depression", "FamilyHistoryAlzheimers",
                     "CardiovascularDisease"])
Y = df["Diagnosis"]
X_scaled = StandardScaler().fit_transform(X)
```

```python
X_train, X_test, y_train, y_test = train_test_split(X_scaled, Y, test_size=0.2, random_state=42)
```

```python
rf_model = RandomForestClassifier(random_state=42)
param_dist = {
    'n_estimators': [50, 100, 200, 300],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['sqrt', 'log2', None],
    'bootstrap': [True, False],
}
random_search = RandomizedSearchCV(estimator=rf_model, param_distributions=param_dist,
                                   n_iter=50, scoring='accuracy', cv=3, verbose=1,
                                   random_state=42, n_jobs=-1)
random_search.fit(X_train, y_train)
print("Best Parameters:", random_search.best_params_)
print("Best Cross-Validation Accuracy:", random_search.best_score_)
best_rf_model = random_search.best_estimator_
y_pred = best_rf_model.predict(X_test)
print("Test Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

Visual Aids

Visual aids such as plots for data distribution, correlation matrices, and feature importance charts are included to support the analysis and help understand the key patterns in the data.



4.Results and Discussion

Model Performance Evaluation

The Random Forest model's performance is analysed based on various evaluation metrics, including accuracy and the classification report, showing precision, recall, F1-score, and support.

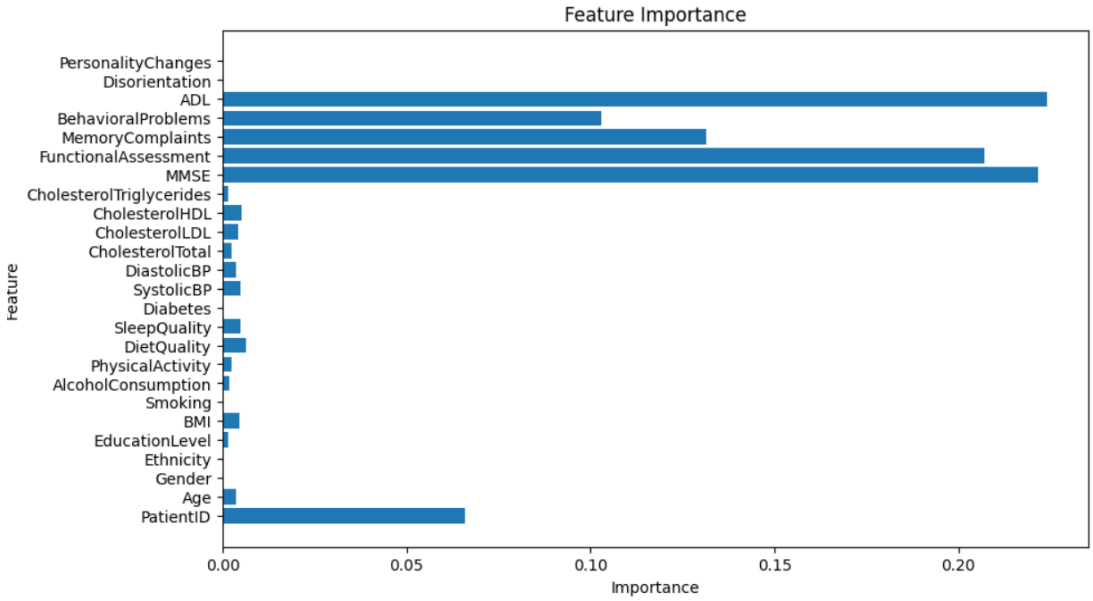
```
random_search = RandomizedSearchCV(estimator=rf_model, param_distributions=param_dist,
                                   n_iter=50, scoring='accuracy', cv=3, verbose=1,
                                   random_state=42, n_jobs=-1)
random_search.fit(X_train, y_train)
print("Best Parameters:", random_search.best_params_)
print("Best Cross-Validation Accuracy:", random_search.best_score_)
best_rf_model = random_search.best_estimator_
y_pred = best_rf_model.predict(X_test)
print("Test Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Fitting 3 folds for each of 50 candidates, totalling 150 fits
Best Parameters: {'n_estimators': 50, 'min_samples_split': 2, 'min_samples_leaf': 4, 'max_features': None, 'max_depth': None, 'bootstrap': True}
Best Cross-Validation Accuracy: 0.9429901105293775
Test Accuracy: 0.9534883720930233
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.96      0.96       277
           1       0.93      0.93      0.93       153

    accuracy                           0.95       430
   macro avg       0.95      0.95      0.95       430
weighted avg       0.95      0.95      0.95       430
```

Feature Importance

The feature importance plot is used to identify which features play a crucial role in predicting Alzheimer's disease. This is valuable for interpreting the model and

understanding the most significant predictors.



Result

The result of the project is that the model is now ready for the prediction of the Alzheimer's disease and it is well trained with an accuracy of 95.34%.

5.Conclusion

Summary of Findings

The summary of this project is that the Random Forest model successfully predicts Alzheimer's disease diagnosis,

achieving high accuracy and providing valuable insights into feature importance.

Limitations

Possible limitations of this project include the dataset's size, the imbalance in the number of Alzheimer's and non-Alzheimer's cases, and the potential for overfitting despite hyperparameter optimization.