Alzheimer Prediction by Handwriting Recognition

Name:	**Manish Mallapur**
Registration No./Roll No.:	21163
Institute/University Name:	IISER Bhopal
Program/Stream:	EECS
Problem Release date:	August 17, 2023
Date of Submission:	November 19, 2023

1 Introduction

The DARWIN dataset contains handwriting data from 174 participants. The protocol used to collect the data comprised of 25 tasks, with different levels of complexity and targeting different areas of the brain. The handwriting/drawing movements performed to execute each task are then described by using 18 features. This dataset includes 156 training and 18 test instances, and all the instances are put into 2 classes - Patient(P) (80 'H' instances in the training dataset) and Healthy(H) (76 'P' instances in the training dataset).

This is a classification problem where the task is binary classification, distinguishing between 'Patient' (P) and 'Healthy' (H) instances.

2 Methods

Github Link

2.1 Proposed Models:

2.1.1 Logistic Regression

Logistic Regression is a linear model that predicts the probability of an instance belonging to a particular class. It is suitable for binary classification tasks. The best parameters: ('C': 10, 'penalty': 'l2')

2.1.2 Support Vector Classifier (SVC)

SVC is a powerful classification algorithm that separates data points into different classes by finding the optimal hyperplane. The best parameters: ('C': 1, 'kernel': 'rbf')

2.1.3 Random Forest Classifier

Random Forest is an ensemble learning method that builds multiple decision trees and combines their predictions. The best parameters: ('criterion': 'gini', 'max depth': 50, 'n estimators': 30)

2.1.4 Decision Tree Classifier

Decision Trees make decisions based on the features of the data. They split the dataset into subsets and recursively apply the process. The best parameters: ('ccp alpha': 0.009, 'criterion': 'entropy', 'max depth': 10, 'max features': 'log2')

2.1.5 Adaptive Boosting

Adaptive Boosting is an ensemble method that combines multiple weak learners to create a strong learner. It sequentially corrects the errors of the preceding model. The best parameters: ('base estimator': SVC(C=0.09, class weight='balanced', kernel='linear', probability=True), 'random state': 10)

2.1.6 K Nearest Neighbours (KNN)

KNN is a simple and effective algorithm that classifies an instance based on the majority class of its k-nearest neighbors. The best parameters: ('n neighbors': 3, 'weights': 'uniform')

2.2 Data Preprocessing

2.2.1 Scaling:

MinMaxScaler is applied to normalize the data.

2.2.2 Dimensionality Reduction:

PCA is used for dimensionality reduction.

2.2.3 Feature Selection:

SelectKBest(with chi squared) method is employed for feature selection.

2.3 Hyperparameter Tuning:

2.3.1 GridSearchCV:

GridSearchCV is utilized to find the best hyperparameters for each model.

3 Experimental Setup

3.1 Cross-validation:

StratifiedKFold with 10 folds is used for cross-validation. This was implemented within GridSearchCV using the cross validation(cv) parameter.

3.2 Evaluation Criteria:

The evaluation criteria for the different models are based on several metrics commonly used in classification tasks. The key metrics include precision, recall, F1-score, and confusion matrix. These metrics provide insights into different aspects of the model's performance.

3.2.1 Precision:

Precision measures the accuracy of the positive predictions made by the model. It is calculated as the ratio of true positive predictions to the total predicted positives (true positives + false positives). A high precision indicates a low false positive rate.

3.2.2 Recall:

Recall (Sensitivity or True Positive Rate) measures the ability of the model to capture all the positive instances in the dataset. It is calculated as the ratio of true positive predictions to the total actual positives (true positives + false negatives). High recall indicates a low false negative rate.

3.2.3 F1-Score:

F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall, making it a useful metric when there is an uneven class distribution. F1-score ranges between 0 and 1, where 1 is the best possible score.

3.2.4 Confusion Matrix:

The confusion matrix is a table that describes the performance of a classification model. It provides a breakdown of true positive, true negative, false positive, and false negative predictions. It is useful for understanding the types and frequencies of errors made by the model.

3.3 Significant Parameters or Hyperparameters:

The hyperparameters that are tuned for each model are mentioned in the code snippets. Here are the key hyperparameters for each model:

3.3.1 Logistic Regression:

Hyperparameters tuned: Penalty (L1 or L2), Regularization strength (C).

3.3.2 Support Vector Classifier (SVM):

Hyperparameters tuned: C (Regularization parameter), Kernel type.

3.3.3 Random Forest Classifier:

Hyperparameters tuned: Criterion (Entropy or Gini), Number of estimators, Maximum depth.

3.3.4 Decision Tree:

Hyperparameters tuned: Criterion (Entropy or Gini), Maximum features, Maximum depth, Cost-complexity pruning alpha.

3.3.5 Adaptive Boosting:

Hyperparameters tuned: Base estimator (SVM, Logistic Regression, Decision Tree, Random Forest, KNN), Random state.

3.3.6 K Nearest Neighbours (KNN):

Hyperparameters tuned: Number of neighbors, Weight function (Uniform or Distance).

3.4 Libraries used:

3.4.1 scikit-learn:

Used for implementing machine learning models, hyperparameter tuning, and evaluation metrics.

4 Results and Discussion

Table1 1 and Table2 2 used to show the experimental results. I've used the Precision, Recall, F-measure and confusion matrix of models run on the pre-processed data providing the best results.

5 Conclusion

In conclusion, this study has explored the feasibility of predicting Alzheimer's through handwriting recognition.

Table 1: Performance Of Different Classifiers Using All Features

Classifier	Precision	Recall	F-measure
Adaptive Boosting	0.875	0.875	0.875
Decision Tree	0.875	0.875	0.875
K-Nearest Neighbor	0.944	0.9375	0.93725
Logistic Regression	0.944	0.9375	0.93725
Random Forest	0.9	0.875	0.873
Support Vector Machine	0.875	0.875	0.875

Table 2: Confusion Matrices of Different Classifiers

Actual Class	Predicted Class	
	H	P
H	7	1
P	1	7

Adaptive Boosting

Actual Class	Predicted Class	
	H	P
H	7	1
P	1	7

Decision Tree

Actual Class	Predicted Class	
	H	P
H	8	0
P	1	7

K-Nearest Neighbor

Actual Class	Predicted Class	
	H	P
H	8	0
P	1	7

Logistic Regression

Actual Class	Predicted Class	
	H	P
H	6	2
P	0	8

Random Forest

Actual Class	Predicted Class	
	H	P
H	7	1
P	1	7

SVM