Alzheimer Prediction by Handwriting Recognition

Name:	**Sayandeep Pal**
Registration No./Roll No.:	21244
Institute/University Name:	IISER Bhopal
Program/Stream:	DSE
Problem Release date:	Aug 17,2023
Date of Submission:	17/11/23

1 Introduction

The objective of this project is to explore whether Alzheimer's disease can be predicted through handwriting recognition. The DARWIN dataset includes handwriting data from 174 participants which has been split into training and test datasets with 156 and 18 participants each for binary classification into patients and healthy individuals and made available to us for this project. It has 450 features in total, 25 groups with 18 features in each group.

Few relevant papers to understand the background of the work:

- Early-Stage Alzheimer's Disease Prediction Using Machine Learning Models
- A study of auxiliary screening for Alzheimer's disease based on handwriting characteristics
- Diagnosing Alzheimer's disease from on-line handwriting: A novel dataset and performance benchmarking

2 Methods

Here in this project, the methodology followed was:

- First task was to visualize the number of people in either class (P for Patients and H for Healthy) to check for class imbalance in the training dataset provided to us. It was observed that we have 80 people and 76 people belonging to the respective classes. Thus no class imbalance
- Next task was to check for any NaN values and check for any duplicate rows and redundancy. No NaN values or data redundancy is present.
- Now to general observation it was clear that the dataset had widely ranging values so a common choice was to scale the dataset. I used the MinMaxScaler for this purpose as it would be easier to visualize and interpret data varying between 0 and 1.
- The dataset had 156 rows and 450 columns for features, so it was imperative that a dimensionality reduction should be employed to avoid the chance of overfitting the model. This was achieved by using the low variance filter after finding out the variance for each feature, this was followed by the high correlation filtering and the feature that was to be dropped was the one with the least variance. Thus this ensured not just dimensionality reduction but also retaintion of the majority of information. Finally, the dataset contained 230 features.

- Still we were left with a lot of features so we opted for PCA(Principal Component Analysis). Now we initially wanted to retain 95 percent of the explained variance but that resulted in a mismatch in the no of principal components in train and test data after similar pre-processing. Moreover, we can take the number of principal components as the min(number of entries, number of features). So 15 principal components was a good choice.
- Since this is a classification problem the choice for classifiers were Logistic Regression(LR), K-Nearest Neighbours(KNN), Support Vector Machine(SVM), Decision Trees(DT), Random Forest(RF), Naive Bayes(GaussianNB).
- Each of the models was trained on 10-fold cross-validation and the best parameters and the best scores were calculated by optimization using GridSearchCV.
- The evaluation metrics for validation are macro-averaged Accuracy, macro-averaged Precision, macro-averaged Recall, macro-averaged F1 score, average Specificity. A detailed classification report for each validation model was generated and the confusion matrix was also used for a better understanding of the classification qualitatively.
- Finally we are performing similar data pre-processing techniques on the test dataset and the best-performing model with the selected best set of parameters was used for the classification and the class labels were then saved in a .txt file.

Here is the link to my Github Profile project repository where the codes, labels and the report are uploaded Github

3 Experimental Analysis

The evaluation metrics used for evaluating the validation models were discussed above. Here the parameters for each classifier will be discussed:

- Classifier: Random Forest Classifier
  Parameters:n estimators, max depth, criterion, max features
- Classifier: Decision Tree Classifier
  Parameters:max features, max depth, criterion, ccp alpha
- Classifier: Support Vector Classifier
  Parameters:kernel, C
- Classifier: Logistic Regression Classifier
  Parameters:penalty, solver, max iter
- Classifier: K Nearest Neighbors Classifier
  Parameters:n neighbors, weights, algorithm
- Classifier: Gaussian Naive Bayes Classifier
  Parameters:,

The performance analysis for each classifier on the training features for the specified evaluation metrics and the confusion matrix has been attached in the following tables:

We see that the Logistic Regression Classifier is performing the best with a specificity of 93 percent and F1 score of 88 percent. We also used the GridSearch Algorithm to find the best parameters for training the Logistic Regression Classifier and hence use this to predict the target variable of the test dataset. We also see that it has significantly the least number of misclassifications among all the other classifiers.

The best set of parameters for the Logistic Regression classifier is ’max iter’: 100, ’penalty’: ’l1’, ’solver’: ’saga’ and the best score(F1 macro) is 88 percent using GridSearchCV.

Table 1: Performance Of Different Classifiers Using All Features

Classifier	Precision	Recall	F1 score	Specificity	Accuracy
Random Forest Classifier	0.87	0.87	0.87	0.89	0.87
Decision Tree Classifier	0.78	0.78	0.78	0.77	0.78
Support Vector Classifier	0.90	0.90	0.90	0.93	0.90
Logistic Regression Classifier	0.88	0.88	0.88	0.93	0.88
K Nearest Neighbors Classifier	0.88	0.86	0.86	0.99	0.86
Gaussian Naive Bayes Classifier	0.85	0.85	0.85	0.80	0.85

Table 2: Confusion Matrices of Different Classifiers

Actual	Predicted Class	
Class	Healthy	Patient
Healthy	69	7
Patient	10	70

Random Forest Classifier

Actual	Predicted Class	
Class	Healthy	Patient
Healthy	53	23
Patient	11	69

Decision Tree Classifier

Actual	Predicted Class	
Class	Healthy	Patient
Healthy	75	1
Patient	21	59

K-Nearest Neighbor Classifier

Actual	Predicted Class	
Class	Healthy	Patient
Healthy	69	7
Patient	12	68

Logistic Regression Classifier

Actual	Predicted Class	
Class	Healthy	Patient
Healthy	61	15
Patient	9	71

Gaussian Naive Bayes Classifier

Actual	Predicted Class	
Class	Healthy	Patient
Healthy	69	7
Patient	9	71

Support Vector Machine Classifier

4 Discussions

This project aims at classifying individuals into 2 classes- Patient and Healthy based on certain features. This can be extended into a Computer Vision project where an image dataset can be made available where people from either class can write their names and based on that we can classify them and confirm with more certainty.