Alzheimer Prediction by Handwriting Recognition

Name:	**Sai Satyam Mohapatra**
Registration No./Roll No.:	21233
Institute/University Name:	IISER Bhopal
Program/Stream:	EECS
Problem Release date:	August 17, 2023
Date of Submission:	November 19, 2023

1 Introduction and Data Description

Alzheimer's disease (AD) has received special attention in that it is a severe and progressive neurodegenerative disease that heavily influences the patient's quality of life, as well as the social costs for proper care. In this context, a large variety of methods have been proposed that exploit handwriting and drawing tasks to discriminate between healthy subjects and AD patients. In our case, the handwriting dataset that we have is called the DARWIN data set. It was generated after research studies established a protocol to define a set of standard handwriting tasks that a patient is to perform to generate the data for the test. As for the proposed experimental protocol, it consists of 25 handwriting tasks - ranging from drawing a spiral or writing a few letters to more complex ones targeting different areas brain activity, to be written on A4 white sheets and placed on a graphic tablet which records the movements of the pen used by the subject. The dataset thus generated had 174 entries of a mix of Healthy and Patient subjects.

Our dataset is a subset of the DARWIN dataset. It consists of handwriting data of 156 patients performing 25 different tasks. Each handwriting/drawing movement is described by using 18 features. Thus there are 25 * 18 = 450 features for each data instance. There are two classes – Healthy(H) for 76 out of 156 instances and Patient(P) for 80 out of 156[1]. There are no missing values.

We have the following files -

- **training_data.csv:** This file contains training feature vectors with 156 data points and 450 features.
- **training_data_targets.csv:** This file contains class labels of all training data points.
  It has 76 Healthy (H) labels and 80 Patient (P) labels.
- **test_data.csv:** This file contains test feature vectors with 19 data points and 450 features whom we need to classify into Healthy (H) or Patient (P)

2 Methods

The **Github Page** of the project is https://github.com/saisatyam95/ML$_PROJECT_2023.git$

Steps that I followed to perform the project :

2.1 The Multi-Classifier Approach[2]

Most, if not all of the popular methods adopt a single machine learning technique to achieve the final classification to discriminate between healthy subjects and AD patients. However, as discussed in the Report-I, we are going to explore a new approach to predicting AD using the given dataset- the Multi-Classifier approach. The Multi Classifier approach works as follows :

- Instead of using the entire feature set i.e instead of using all the 450 features to build our classifier, we divide the feature set into 25 groups. This is particularly appropriate for our dataset as the data itself was generated in 25 groups (each feature subset from a subject performing the one of the 25 tasks)
- We use these 25 seperate feature subsets, each with 18 unique features. This makes a seperate feature set for representing each of the 25 tasks presented to the patients. These would be used for the training of the Multi-Classifiers.
- We train each of our classifiers on each one of the the 25 feature groups generated above, so we have 25 basic classifiers, one using each group. Each of the classifiers are then used on the test data to predict whether the patient is Healthy(H) or Patient(P). The final prediction result is produced by combining all the predictions by a majority vote.

2.2 Implementation

We selected these popular top performing classifiers for building the basic classifiers -

1. Random Forest Classifier
2. Logistic Regression
3. Linear Discriminate Analysis
4. Gaussian Naive Bayes
5. Support Vector Classifier

The training data was split into testing data and training data in a randomly selected 85/15 splitting ratio using *sklearn.model_selection.train_test_split()* with stratification 'ON'. The labels were also slpit accordingly. The performance of all the classifiers on the test data would be evaluated after the training them using the train data.

Parameter tuning of the classifiers was performed by using *sklearn.model_selection.GridSearchCV().* The scoring criteria for GridSearchCV was set to macro averaged f1-score (f1_macro) instead of the default accuracy score for better tuning of the performance of the classifiers.

The crossvalidation for the training was done by a 5-fold cross-validation. GridSearchCV has a default inbuilt functionality of performing k-fold cross validation with k=5 by default, on all the models while searching for the best model. So, a CV method was not be explicitly specified as k = 5 is good enough for our data.

Then the Multi-Classifiers were trained. The same GridSearchCV method was used for parameter tuning before fitting each feature subset to the model to build a classifier. The predicted class labels of each basic classifier of the multi-classifier system would be stored in arrays to be combined later for getting the final labels for the test data.

Individual feature selection was avoided as the features were present in groups, which in turn were generated from performing a specific task. Picking one feature from one group and another from another group would be counter-intuitive as the feature vectors of a single group would be related to each-other and not to members another group. Instead a feature selection way is suggested in the discussion towards the end of the report for selecting/rejecting an entire group.

3 Experimental Analysis

Each of the basic and multi-classifiers were trained on the training datapoints and tested on the test datapoints. The train and test datapoints being obtained by the earlier mentioned 85/15 split of the train data. The best parameters returned by GridSearchCV was used for the classifiers. The results for each classifiers are reported in the below table in terms of macro avereged precision, recall and f1-score.

Table 1: Performance Of Basic Classifiers Using All Features

Classifier	**Precision**	**Recall**	**F-measure**
Random Forest	**0.92**	**0.92**	**0.92**
Logistic Regression	0.88	0.88	0.87
Linear Discriminate Analysis	0.90	0.88	0.87
Gaussian Naive Bayes	0.88	0.88	0.87
Support Vector Machine	0.88	0.88	0.87

Table 2: Performance Of Multi-Classifiers Using All Features

Classifier	**Precision**	**Recall**	**F-measure**
Random Forest	**0.96**	**0.96**	**0.96**
Logistic Regression	**0.96**	**0.96**	**0.96**
Linear Discriminate Analysis	0.84	0.83	0.83
Gaussian Naive Bayes	0.90	0.88	0.87
Support Vector Machine	0.93	0.92	0.92

From among the single classifiers, **Random Forest** performs the best, giving a f1-score of **0.92** after parameter tuning.

Now from among the Multi-Classifiers, we can see that the Multi-Classifier approach increases the prediction f1-score by **4% for Random Forest** and **9% for Logistic Regression**. Also for Support Vector Machine,the f1-score increases by 5% On the flip side, for Linear Dicriminant Analysis and Gaussian Naive Bayes classifier the f1-score decreses or remains the same. So, we can conclude that the Multi-Classifier approach does not necessarily improve the performance of the classification technique in all cases.

The Confusion Matrices for all the single and Multi-Classifiers are displayed in the tables below -

Table 3: Confusion Matrices of Single Classifiers

Actual Class	Predicted Class	
	Healthy	Patient
Healthy	11	1
Patient	1	11

Random forest

Actual Class	Predicted Class	
	Healthy	Patient
Healthy	11	1
Patient	2	10

Logistic Regression

Actual Class	Predicted Class	
	Healthy	Patient
Healthy	12	0
Patient	3	9

Linear Discriminant Analysis

Actual Class	Predicted Class	
	Healthy	Patient
Healthy	10	2
Patient	1	11

Gaussiian Naive Bayes

Actual Class	Predicted Class	
	Healthy	Patient
Healthy	11	1
Patient	1	10

Support Vector Machine

Table 4: Confusion Matrices of Multi-Classifiers

Actual Class	Predicted Class	
	Healthy	Patient
Healthy	12	0
Patient	1	11

Random forest

Actual Class	Predicted Class	
	Healthy	Patient
Healthy	11	1
Patient	0	12

Logistic Regression

Actual Class	Predicted Class	
	Healthy	Patient
Healthy	11	1
Patient	3	9

LDA

Actual Class	Predicted Class	
	Healthy	Patient
Healthy	12	0
Patient	3	9

Gaussiian Naive Bayes

Actual Class	Predicted Class	
	Healthy	Patient
Healthy	12	0
Patient	2	10

Support Vector Machine

As observed from the results after running all the proposed models, we have two Multi-Classifier models giving the highest f1-socres of 0.96 - Random Forest and Logistic Regression. We chose **Logistic Regression** out of the two even though both have same f1-score because Logistic Regression Multi-Classifier has greater Recall value (1.0) for the Patient(P) label compared to Random Forest (0.92). This means Logistic Regression is better at classifying the patients as Patients (P) whereas Random Forest is good at classifying the healthy (H).
I have then trained the best proposed model - Logistic Regression Multi-Classifier on the entire dataset and run it to predict the labels for the test data and saved the predictions in a text file.

4 Discussions

Handwriting activity can be an innovative way of characterising the peculiarities of neurodegenerative diseases. Since the dataset is a small one having few datapoints, the single classifiers show lower performance scores. Also the number of features is quite large in comparision to the number of datapoints. In this case, the multi-classifier approach provides a good method for classification. However this way has not been explored much and the state of the arts are still single classifiers.

Also, another method of doing feature selection would be to select some best performing base classifiers of the multi-classifier system to build the final classifier instead of using all the 25 classifiers. To do this we can use f1-score or accuracy of each base classifier and then decide what to combine.

References

[1] Nicole Dalia Cilia, Claudio De Stefano, Francesco Fontanella, and Alessandra Scotto Di Freca. An experimental protocol to support cognitive impairment diagnosis by using handwriting analysis. *Procedia Computer Science*, 141:466–471, 2018. The 9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2018) / The 8th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2018) / Affiliated Workshops.

[2] Giuseppe De Gregorio, Domenico Desiato, Angelo Marcelli, and Giuseppe Polese. A multi classifier approach for supporting alzheimer's diagnosis based on handwriting analysis. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part I*, pages 559–574. Springer, 2021.