EEG Classification

Angel Lopez Manriquez
Juan Diego Arango Ramos

Contents

Abstract

In this document we present our approach to the problem of predicting Electroencephalogram (EEG) results, a test that provides really useful information about a person's brain. We used the public "EEG Eye State Data Set". We implemented 6 supervised classification methods: Logistic "Regression", Naive bayes (Gaussian), KNeighbors classifier, Perceptron, Decision Tree Classifier and Random forest classifier. We used the following libraries for the implementation : numpy, matplotlib, pandas and sklearn. Our dataset was divided into test and train data. Train was used to learn the classification models and test was used to compute the accuracy of the different models. We obtained really good accuracies from the KNeigbors, decision tree and the Random forest models.

1 Description of the problem

The task we have to solve is the classification of the "EEG Eye State Data Set", introduced in [1] [2] [3] and available from the UCI machine learning repository https://archive.ics.uci.edu/ml/datasets/EEG+Eye+State#.

The dataset includes features extracted from one continuous EEG measurement of a person. During a 117 second period, the eye state of the person was detected and recorded by a camera. Then analyzed frame by frame and added to the dataset. Each frame was tagged with a 0 or a 1 where 0 represents the eye is opened and 1 represents the eye is closed. The goal of the project is to predict eye state of an individual (opened or closed).

The database has the following characteristics:

- 14 attributes.
- 2 classes: '0', '1'.
- 14979 instances.
- The data is decently balanced. The number of instances in each class are: 8257, 6722

2 Description of our approach

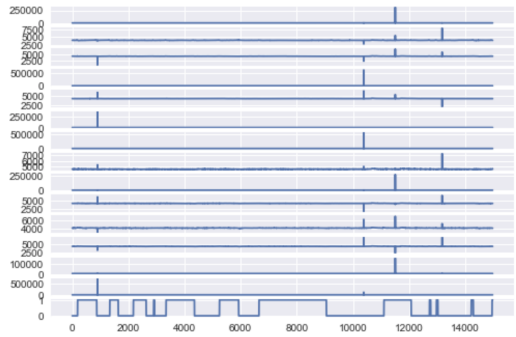
We organized the implementation of the project according to the tasks:

1. Preprocessing of the dataset.
2. Define and learn the classifiers using the training data.
3. Design the validation method to evaluate the accuracy of the proposed classification approaches.

2.1 Preprocessing

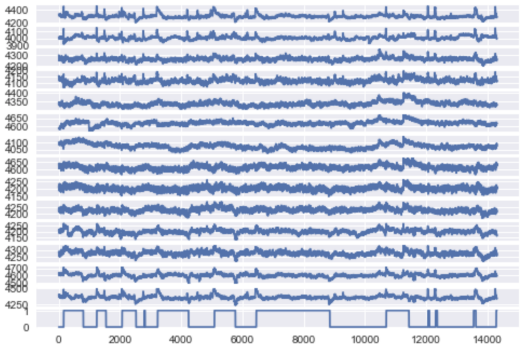
We use the arff library to load the dataset (it was a .arff file). We use the pandas library to load it in a data frame. After that, the class tags are all converted to integers because some of the 0's and 1's are saved as strings. Graphics are made for each of the features of the dataset, we identify some clear outliers, see Figure 1.

Figure 1: Data before cleaning



We use the Standard deviation method to eliminate the outliers, the method essentially eliminates the data that is more than 3 standard deviations from the mean, see Figure 2

Figure 2: Data after cleaning



The data was split into two different sets: train and test. We use the train set to learn all the classifiers and the test set for evaluating their accuracy.

2.2 Classifiers

We use six different classifiers:

1. Perceptron with parameters:
   - **`max_iter =1000`**
   - **`tol = 0.001`**
   - **`eta0=0.1`**
2. Logistic regression with parameters:
   - **`random_state = 0`**
3. Decision tree with parameters
   - **`random_state = 0`**
4. KNeighborsClassifier with parameters
   - **`n_neighbors = 100`**
5. GaussianNB with no parameters provided.
6. RandomForestClassifier with parameters
   - **`random_state=0`**

For each classifier, these were the parameters by default of the scikit-learn library. [1]

2.3 Validation

To validate our results we compute the classifier accuracy in the test data. Another possibility was to compute the cross-validation in the complete dataset but we used the split between train and test because it was simpler.

As an additional validation step we computed the confusion matrices for the six classifiers.

3 Implementation

All the project steps were implemented in Python. We used pandas for reading and preprocessing the dataset, and scikit-learn for the classification tasks. We illustrate how the implementation works in the Python notebook EEG jupyter notebook

4 Results

The accuracies produced by the Perceptron, Logistic regression, Decision tree, GaussianNB, KNeigbors and Random forest classifiers were, respectively: 0.5823, 0.6445, 0.8245, 0.6469, 0.8924 and 0.8891. Even though the best accuracy was given by the KNeigbors

The results of the computation of the confusion matrices for Perceptron, Logistic regression, Decision tree, GaussianNB, KNeigbors and Random forest classifiers are respectively shown in Tables 1, 2, 3, 4, 5 and 6.

col_0	0	1
0	1570	14
1	1181	96

Table 1: Confusion matrix produced by the Perceptron classifier, Time execution: 0.04653s

[1] Perceptron(max_iter=1000, tol=1e-3, eta0=0.1) KNeighborsClassifier(n_neighbors=100)

col_0	0	1
0	1234	350
1	667	610

Table 2: Confusion matrix produced by the Logistic regression classifier, Time execution: 0.2815s

col_0	0	1
0	1321	263
1	239	1038

Table 3: Confusion matrix produced by the Decision tree classifier, time execution: 0.1344s

col_0	0	1
0	1215	369
1	641	636

Table 4: Confusion matrix produced by the GaussianNB classifier, time execution: 0.2454s

col_0	0	1
0	1496	88
1	234	1043

Table 5: Confusion matrix produced by the KNeigbors classifier, time execution: 0.7037s

col_0	0	1
0	1493	91
1	226	1051

Table 6: Confusion matrix produced by the Random forest classifier. Time execution: 0.2264s

5 Conclusions

In our project we have applied Perceptron, Logistic regression, Decision tree, GaussianNB, KNeigbors and Random forest to the EGG eye state dataset. We have computed the accuracy of these classifiers and observed that KNeigbors classifier produces the highest accuracy also the decition tree classifier and random forest classifier produced really good accuracies. The three models achieve the goal of the project in a pretty accurate way.

References

[1] F. Laurent S. Baillet J. Martinerie M. Besserve, K. Jerbi and L. Garnero. Classification methods for ongoing eeg and meg signals. biological research. page 415–437, 2007.

[2] R. Santana el crack W.-L Zheng and B.-L Lu. Comparison of classification methods for eeg-based emotion recognition. *In Proceedings of the 2015 World Congress on Medical Physics and Biomedical Engineering*, page 1184–1187, 2015.

[3] C.W. Anderson D. Garrett, D.A. Peterson and M.H. Thaut. Comparison of linear, nonlinear, and feature selection methods for eeg signal classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):141–144, 2003.