Report on BCI Challenge @ NER 2015

https://www.kaggle.com/competitions/inria-bci-challenge/overview

**Introduction**

The goal of this Project was to detect error related potential during p300 spelling task using EEG. In the task each subject (16 in total) was presented letters and numbers of the screen (36 possible items displayed in matrix shape) to spell words. Each item of word is selected one at a time, by flashing screen items in group and in random order. The selected item is the one for which the online algorithm could most likely recognize the typical target response. The task is to detect if the selected item is correct or not. This is done by analyzing brain signals from different electrodes on subjects’ head.

**The Dataset**

The data consisted of 56 channels from EEG sensors located according to the 10-20 system. Eye movements were recorded using EOG (1 channel). EEG channels were downsampled to 200Hz, filtered with a fifth-order Butterworth filter between 1 and 40Hz, and divided into epochs of 1.3 seconds (260 time points) each. This resulted in a dataset of shape (5440, 56, 260) [Epochs, Channels, Time], with each of the 16 participants contributing 340 samples. Two participants were then set aside for the test set, resulting in two sets: 4760 samples for training and 680 samples for testing.

To select important EEG channels, and therefore reduce the amount of noise in the data, we create a function that compares how signals differ between positive and negative epochs. It looks at the average signals of positive and negative epochs separately, calculates the variation (standard deviation) in these signals for each EEG channel, and then checks if this variation meets a specified threshold (1.5). Channels where the variation is below the threshold are excluded from further consideration. Only 10 channels were selected: Fp1, Fp2, AF7, AF3, AF4, AF8, F7, F5, Pz, P2. Figure 1 shows averaged signals across 56 EEG channels. The dataset was not balanced, so majority of classes were positive, this can be seen in figure 2.



Figure 1. Average difference between signals with 0 and 1 labels across 56 channels.

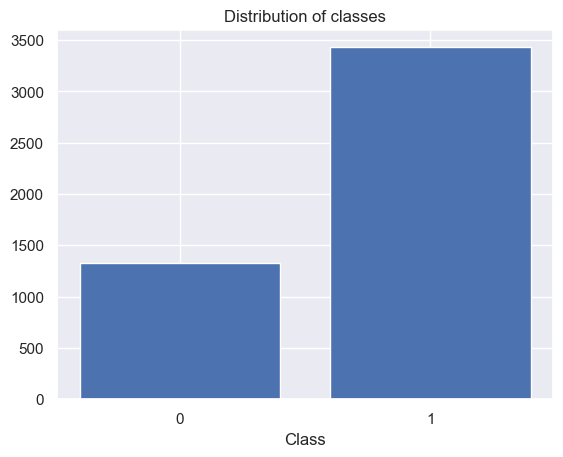


Figure 2. Distribution of classes. Class 1 is more than twice as common as class 0.

**The Method**

Before feeding data to our SVM model, we apply a wavelet transform (db4) at level 5, merge all the generated coefficients, and reshape them into a 2D array with the shape (4760, 2910). We then normalize the data across the last dimension using the L2 norm. We choose the best hyperparameters via grid search with 14 splits, without shuffling. Since the training dataset consists of 14 participants, each with the same number of samples, this method prevents inter-subject contamination. The parameters considered in the grid search are shown in Figure 3. This grid search is by no means exhaustive, we limit hyperparameter tuning to a small number of parameters due to the simplicity and high computational cost of a vanilla grid search. The best hyperparams are: *C*=10, *gamma*=”scale”, *kernel*=”rbf”.

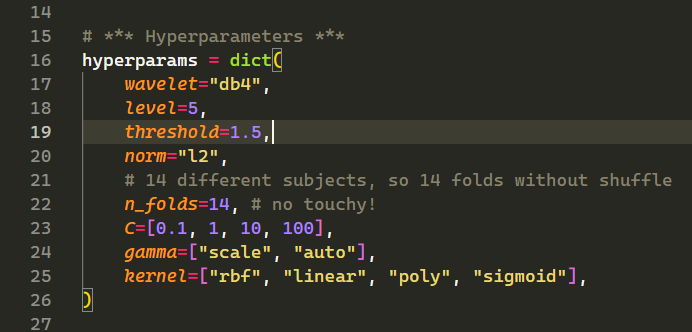


Figure 3. Hyperparameters considered in grid search.

**Evaluation**

Since the distribution of classes is not balanced, regular accuracy would give inaccurate results. Therefore, in this project, we decided to use precision and recall in addition to accuracy. The results are pretty bad for this binary classification task, with our model achieving an accuracy of 0.68, precision of 0.75, recall of 0.83, and an F1 score of 0.79 on train data. Figure 4 presents the confusion matrix for the training dataset.

A recall of 0.83 means that our model correctly identified 83% of the positive samples in the training data. Precision of 0.75 indicates that when our model predicted a positive class, it was correct 75% of the time. The F1 score combines precision and recall into a single metric using their harmonic mean.

For test dataset the model achieved accuracy of 0.60, precision of 0.57, recall of 0.60, F1 score of 0.57. This clearly means that the model has overfitted to the train dataset. In order to improve performance some sort of regularization could be further employed. To achieve a higher recall, we could increase the regularizer C to 100. However, this would come at the expense of accuracy, highlighting the trade-offs in model performance. Figure 5 shows confusion matrix for test dataset.

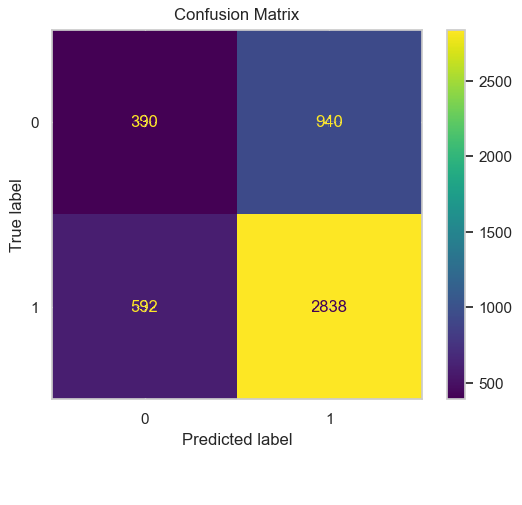


Figure 4. Confusion matrix on train dataset.

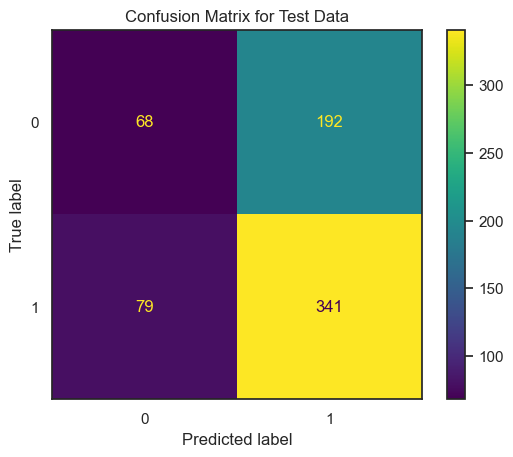


Figure 5. Confusion matrix on test dataset.

**Conclusion**

The created model is fairly weak and overfitted. The highest accuracy score we found for this task was 0.85, although the authors of that model did not mention precision or recall (we assume they were decent). They admitted to using metadata such as the number of sessions and trials, which significantly boosted their model's performance. They explained that participants tend to become tired and lose attention in later trials, which leads to more errors.

However, in this project, we chose not to use such metadata and relied solely on EEG and EOG data. As a result, the project did not meet our (low) expectations. Future improvements could include applying different transformations to the data or using more advanced techniques to select EEG channels.