Advanced Machine Learning Projects

Konstantinos Barmpas
(TeamBeugi)
Eidgenössische Technische Hochschule Zürich
(ETH Zürich)
kbarmpas@student.ethz.ch

Abstract

This report outlines our submissions for the four tasks of the module "Advanced Machine Learning" at ETH Zurich during the academic year 2019-2020.

1. Task 1

Description: "Predict the age of a person from their MRI scan. This task is primarily concerned with regression. The task1 team has perturbed the original MRI features in several ways. We needed to perform outliers detection, feature selection, and other preprocessing to achieve the best result."

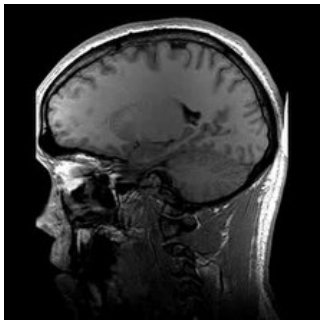

Figure 1: Brain MRI Scan

1.1. Our Solution

The task1 was to predict a person's age from the brain image data: a standard regression problem. The original dataset included 832 features as well as a lot of NaN values and a few outliers. A good preprocessing stage was necessary in order to have a well defined dataset that could be used in our regression model. First step was the imputation of the dataset. Filling each NaN value with the median of each feature column. The use of the median instead of other value (e.g. mean) is justified since a lot of outliers are included in the dataset.

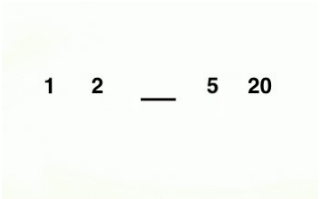

Figure 2: Example: median: 3 mean: 7

Next step was the feature extraction. By using the "autofeat" library [2], we extracted the 21 most important features. The way the algorithm works is the following: Goes through a loop of correlation of features with target, selects promising features, trains Lasso regression model with promising features, filters the good features keeping the ones with non-zero regression weights. We updated the datasets by keeping only the 21 most important features. Finally, we used these updated datasets for the training of our final regression model. A lot of outlier detection techniques were used but we decided to keep the outliers and use a tree-based method for our final model. Tree-methods have been proved to be robust to outliers and we avoid risking to exclude important features / points from the dataset. The "ExtraTreesRegressor" model from the "sklearn" package was used and fine tuned based on the R2 score performance in cross-validation. The final model had a cross-validation score >0.6 and in the submission leaderboard scored 0.6812 while the hard baseline was set to 0.65 by the Task1 team.

2. Task 2

Description: "Multi-class classification in an unbalanced dataset. This task is primarily concerned with multi-class classification where you have 3 classes. The task2 team has changed the original image features in several ways. We needed to deal with class imbalance; in the training set, there are 600 examples from class 0 and 2 but 3600 exam-

ples from class 1. Test set has the same class imbalance as the training set."

2.1. Our Solution

The problem was a classification task. We had 3 classes but these were unbalanced with one to have a significantly higher number of samples. In order to fix the unbalanced classes different techniques were used including upsampling of the classes with lower number of samples and downsampling of the class with higher number of samples. All of these techniques were outperformed though by using different weights for the penalty term during the training of our classification model. In this way we use all the power of the real dataset since we do not lose information like we did in the downsampling scenario or we introduce "artificial / fake" information which might be wrong like we did in the upsampling scenario. Finally, using cross-validation we tested different models for the classification task. The best model proved to be through the cross-validation a Support Vector Machine model. More precisely, we used the SVC function from sklearn package. For the hyperparameters we used balanced classes weights and One -Versue -Rest strategy and C = 1.0 for the penalty term error. The model has 0.702 mean score in the cross-validation and 0.7185 in the public scoreboard.

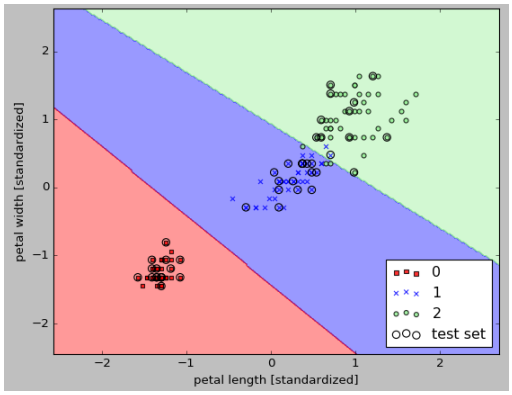

Figure 3: Support Vector Machine

3. Task 3

Description: "Heart rhythm classification from raw ECG signals. This task is primarily concerned with the classification of entire time series into one of 4 classes. We needed to deal with raw ECG recordings of different length sampled at 300Hz to predict heart rhythm."

3.1. Our Solution

The problem was a classification task. We had ECG measurements of 4 classes that were unbalanced and they did not have the same length. We used different techniques

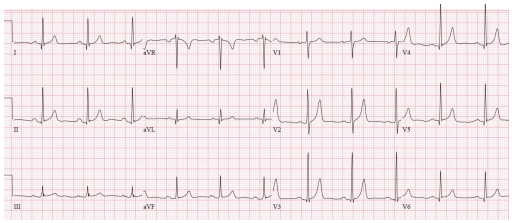

Figure 4: Raw ECG Signal

to extract features that we used for the classification. For each ECG signal we extracted the autocorrelation, the average and the power. We also extracted 15 coefficients of their FFT. For each ECG using biosspy we extracted the heartbeats, averaged them (to reduce the noise) and created a characteristic average of the same length of each patient.

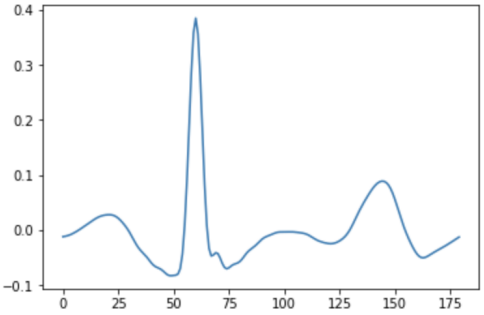

Figure 5: Characteristic Average Heartbeat of the Same Length of Each Patient

For each of these signals (after normalization) we extracted the energy of the wave, the T, S, P, R, Q peaks, the ST QRS PR intervals, QRS/T and QRS/P ratios, the median, mean and interval of the amplitude and the db2 coefficients.

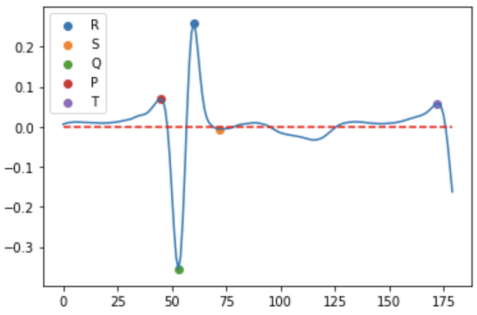

Figure 6: Extract Characteristics from Heartbeat

Finally, the library biosspy [3] gave us the locations of peaks in the original wave, the timings as well as the heart beats and their timings. For all of them we calculated the mean, median and standard deviation. We also extracted

the mean, median and standard deviation of the differences between the peaks' timings (important feature to classify noise, normal heart rate and abnormal heart rhythms). Using all of these features we trained a GradientBoosting model which was fine-tuned using a Cross-validation grid search. The model has 0.817 mean score in the cross-validation and 0.833 in the public scoreboard.

4. Task 4

Description: "Sleep staging classification from raw EEG/EMG signals. This task is primarily concerned with the classification of entire time series into one of 3 classes. We needed to deal with raw short recordings of 4 seconds each."

4.1. Our Solution

The problem was a classification task. We had EEG / EMG measurements of 3 classes that were unbalanced. We used different techniques to extract features that we used for the classification.

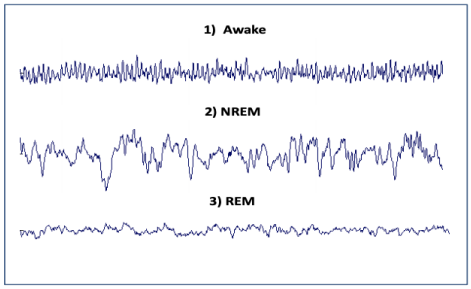

Figure 7: Signals for different sleep stage [4]

For each EEG signal using the biosspy library we extracted the energy waves bands (alpha, beta, gamma, delta and theta). For each of these waves we calculated the mean, average, standard deviation, range and energy. These features were important to distinguish between NOT-REM and REM stages. Another feature that we added was sum of the power in all five energy bands.

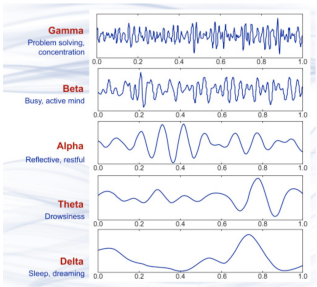

Figure 8: Brain Wave Samples [1]

For each EMG signal we extracted the energy, the zero-crosses and the sign changes of the singal. EMG characterizes brain muscle activity and these features are important to distinguish if the subject is asleep or not. With RandomForest we performed feature selection (dimensionality reduction). Using all of the remained features we trained a Support Vector Machine which was fine-tuned using Cross-validation and Leave-one-subject-out (LOSO) validation. The model has 0.93 mean score in the LOSO cross-validation and 0.9397 in the public scoreboard.

References

[1] Animal and T. M. for CNS Drug Discovery. Electroencephalogram https://www.sciencedirect.com/topics/medicine-and-dentistry/electroencephalogram.

[2] M. R. Franziska Horn, Robert Pack. The autofeat python library for automated feature engineering and selection `https://arxiv.org/pdf/1901.07329.pdf`.

[3] PIA-Group. Biosspy https://biosppy.readthedocs.io/en/stable/.

[4] M. G. Sara Mahvash Mohammadi, Shirin Enshaeifar and S. Sanei. Classification of awake, rem, and nrem from eeg via singular spectrum analysis http://150.162.46.34:8080/embc-2015/papers/12451428.pdf.