Assignment 4

Mobile and Wearable Computing

Alessandro Gobbetti

December 6, 2024

Contents

1 Introduction **1**

2 Dataset **1**

3 Leave One Day (per user) Out Cross Validation **2**

4 Other models **2**

Università della Svizzera italiana
Faculty of Informatics

1 Introduction

This is the fourth assignment for the Mobile and Wearable Computing course. The assignment focuses on machine learning and cross validation. The assignment is based on tutorial 7 from the course material, in which physiological data collected from Empatica E4 wristband is used to predict the engagement level of a user.

The code is available at:
https://github.com/Alessandro-Gobbetti/Mobile-and-Wearable-Computing/tree/main/Assignment4.

2 Dataset

In this section, we'll compute features and train models on the dataset provided for the assignment. The dataset contains EDA (Electrodermal Activity) measurements collected from different users.

The data is split into 15701 windows of 10 seconds each. For each window, the following features are computed:

- Rise time: the amount of seconds the signal takes to rise from 10% of the max value to 90% of the max value, in each window. It is the measure of the ability of the signal to respond to fast inputs.
- Decay time: opposite of rise time.
- Dynamic Range: max - min.
- Slope: this is the slope of a line that fits the data.
- Absolute slope: absolute value of the slope.
- First derivative mean: average of the first derivative
- First derivative std: standard deviation of the first derivative

For comparison, we'll also use the dataset from tutorial 7, which is split into 7440 windows of 10 seconds each. The features are the same as the ones computed for the assignment dataset.

For both datasets, we train XGBoost classifier and a Dummy classifier from Scikit-learn. Table 1 shows the 5-fold cross validation results, while Table 2 shows the leave-one-subject-out cross validation results.

Model	**T07 data**	**Assignment data**
XGBClassifier	68.75 $\pm$ 0.78 %	63.40 $\pm$ 0.67 %
DummyClassifier	47.58 $\pm$ 0.31 %	50.49 $\pm$ 0.24 %

Table 1: Balanced accuracy 5-fold cross validation results for the two datasets.

Model	**T07 data**	**Assignment data**
XGBClassifier	45.27 $\pm$ 23.23 %	41.92 $\pm$ 16.32 %
DummyClassifier	49.47 $\pm$ 1.02 %	49.83 $\pm$ 1.21 %

Table 2: Balanced accuracy Leave-one-subject-out cross validation results for the two datasets.

The two models performs similarly on the two datasets. The XGBClassifier has a higher accuracy than the DummyClassifier, which is obviusly expected.

Leave-one-subject-out cross validation leads to lower accuracy as users have different physiological responses to the same stimuli. Using one user only for testing can thus lead to lower results. However, this validation technique is useful in real-world scenarios where the model is trained on data from different users and tested on a new user. A good model should be able to generalize well to new users.

3 Leave One Day (per user) Out Cross Validation

In this section, we'll perform leave-one-day-out cross validation. The day is identified by the date column. Same dates from different users should be considered different "days". In the validation paradigm only a single session from a single user is left out for validation. The rest of the data is used for training.

```
lodo_groups = eda_features.groupby(["username", "session"]).ngroup()
```

The features are grouped by username and session, and a group number is assigned to each group. This group number is used to split the data into training and validation sets. This results in 27 groups for the assignment dataset.

Model	5-fold	LOSO	LODO
XGBClassifier	63.40 ± 0.67 %	41.92 ± 16.32 %	64.17 ± 2.27 %
DummyClassifier	50.49 ± 0.24 %	49.83 ± 1.21 %	50.21 ± 2.06 %

Table 3: Balanced accuracy for different models on the three cross validation techniques.

Table 3 shows the results of the leave-one-day-out cross validation. This technique leads to higher accuracy as the model is trained on samples from the same user in the test set. Models can thus learn the user-specific patterns and perform better on the test set.

4 Other models

In addition to the XGBClassifier and DummyClassifier, we also trained a RandomForestClassifier, SVC, and GaussianNB. The results are shown in Table 4.

Model	5-fold	LOSO	LODO
XGBClassifier	63.40 ± 0.67 %	41.92 ± 16.32 %	64.17 ± 2.27 %
DummyClassifier	50.49 ± 0.24 %	49.83 ± 1.21 %	50.21 ± 2.06 %
SVC	50.05 ± 0.12 %	47.03 ± 36.52 %	50.17 ± 0.38 %
RandomForestClassifier	63.60 ± 1.06 %	42.00 ± 16.24 %	64.02 ± 2.16 %
GaussianNB	54.81 ± 1.07 %	28.44 ± 22.48 %	54.79 ± 2.25 %

Table 4: Balanced accuracy for different models on the three cross validation techniques.

How do these models compare to the previous ones?

The RandomForestClassifier performs similarly to the XGBClassifier. The SVC and GaussianNB have lower accuracy, and are thus not recommended for this task. The Support Vector Classifier is also way slower to train than the other models.