PPG Field Study Dataset Explanation and Solution

Ge Qiu[1,*]

Ecole Supérieure d'Ingénieurs

Léonard de Vinci, France

Email: ge.qiu@edu.devinci.fr or qiugehhht@hotmail.com

1.1 Introduction (Dataset overview)

Photoplethysmography (PPG) is widely used for continuous heart rate monitoring.

The dataset consist of 3 major parts: data from chest sensor, data from wrist sensor, activity situation (target variable) and some personal information of the subject (e.g., age, gender, etc.).

Chest sensor measures motions of the subject (3D-accelerometer), heart rate ground truth (ECG), Electrodermal Activity (EDA), electromyogram (EMG), temperature, and respiratory signal, with a frequency of 700Hz (700 records/second).

Wrist sensor measures motions of subject (3D-accelerometer, 32Hz), Blood Volume Pulse (BVP, 64Hz), Electrodermal Activity (EDA, 4Hz), and temperature (4Hz).

1.2 Introduction (Covariates processing)

According to my calculation of the target variable (activity), 36848 records in about 2.5h (each subject were measured in an about 2.5h period) is approximately 4Hz, which means other covariates should compressed to the same length, so the machine learning algorithm can be deployed.

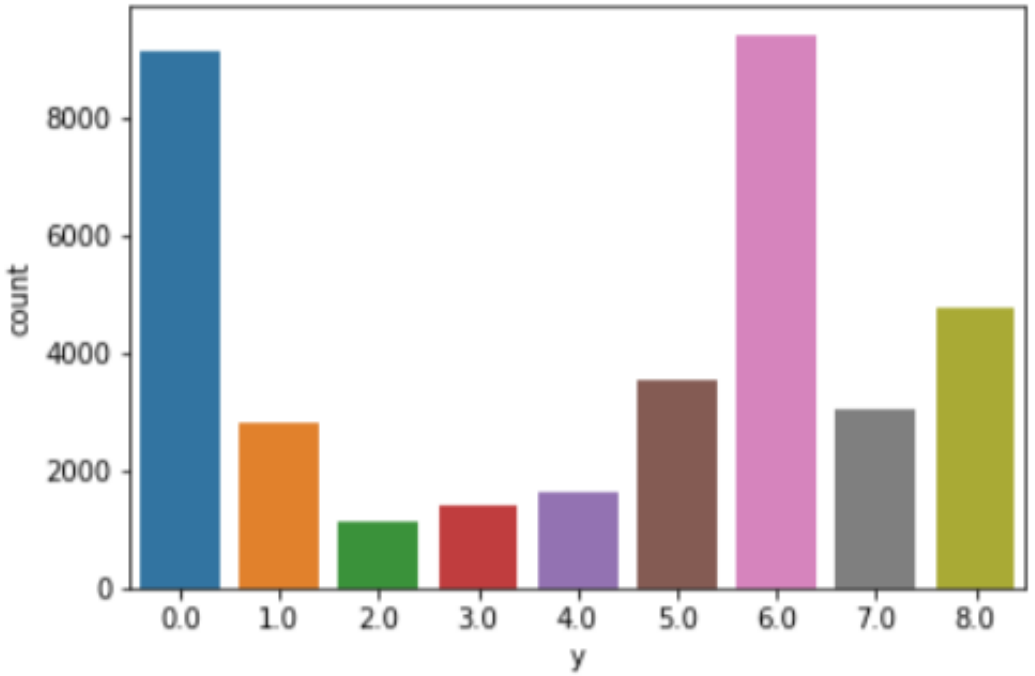
For each variables measured by chest sensor, every 175 (700Hz/4Hz) records correspond to a records of target. I used mean value of every 175 record to reduce the length of variables from chest sensor. The same method were also use in variables from wrist sensor according to their frequency, respectively (details can be seen in “preprocessing.ipynb”).

For 3D-ACC data in both chest and wrist sensor, each record consists of 3 elements representing 3 different directions of movement. I divided each record into 3 element (e.g., c_acc1, c_acc2, c_acc3), and each element is a column (through this step, 6 new feature were built (3 for wrist 3D-ACC and 3 for chest 3D-ACC)).

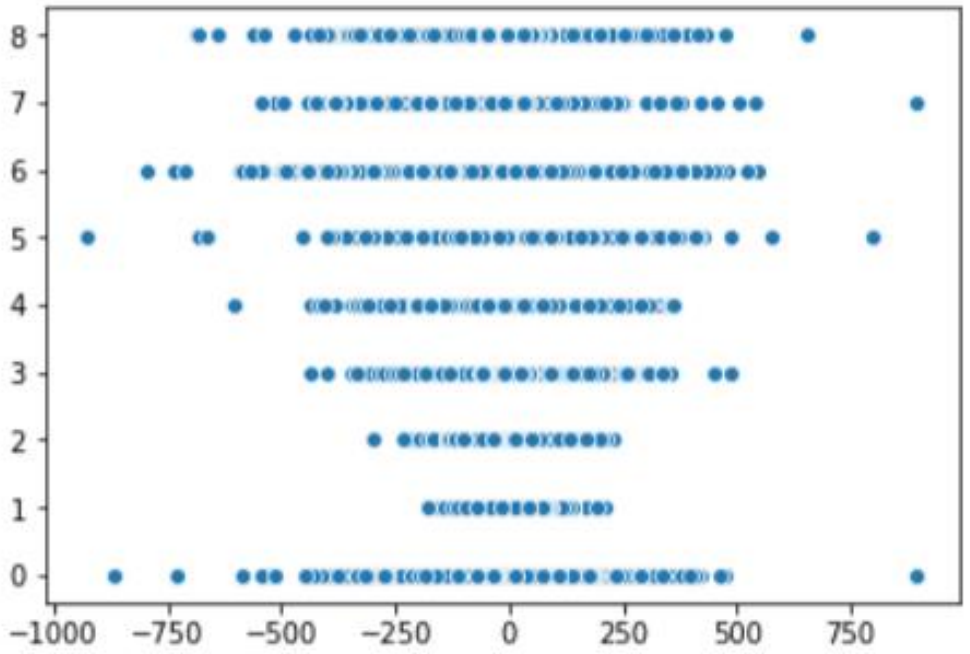
- The target variable consists of 9 classes, 0 represent transition of movement (movement between 2 standard movement), 1-8 represent standard movement such as walking, driving, etc.
- All the other variables are continuous.

- *Note: Since there are total 15 subjects and each subject recorded about 1-3Go data, it is difficult to use all subjects to build a model (time consuming), I only used S1 as dataset. (if you want to use all subjects, personal information such as gender, age should also be considered as variables)

1.3 Introduction (Type of variables)



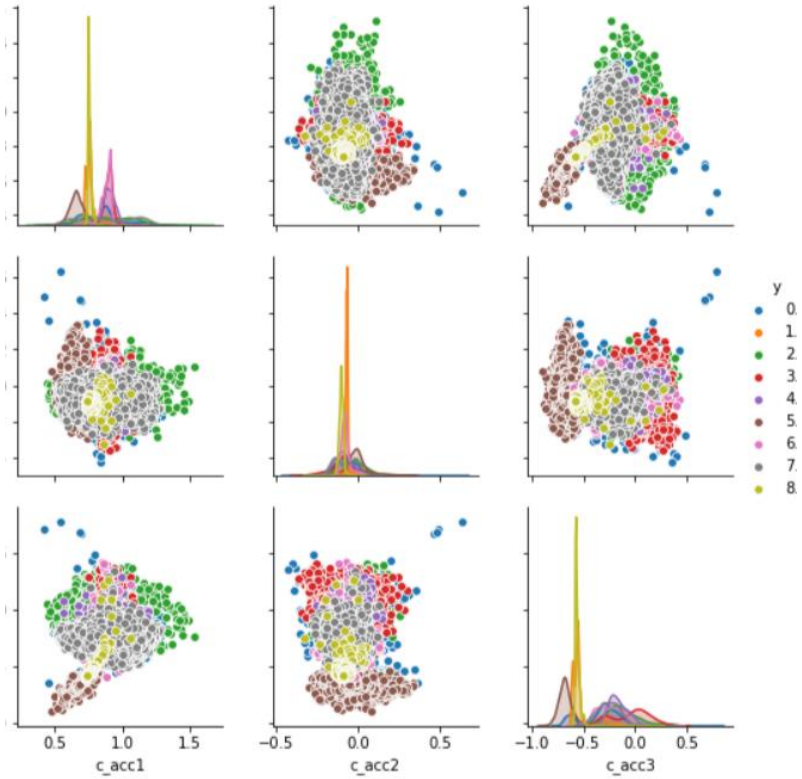
Number of records in each class in target variable



Scatter plot between BVP and target variable

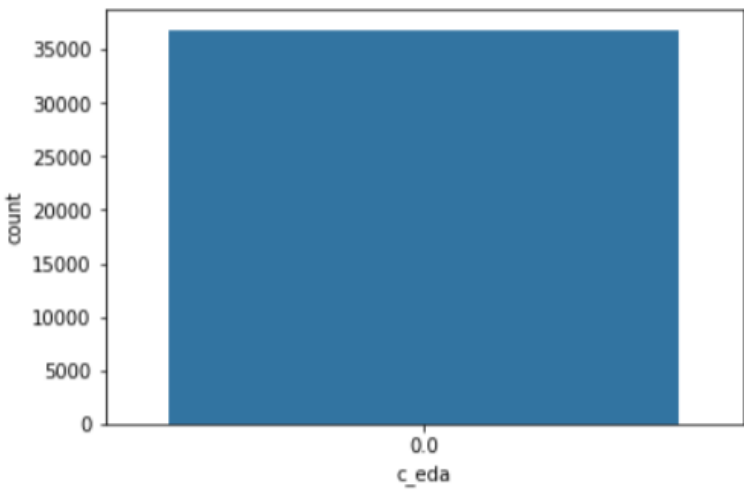
2. Data visualization (feature selection)

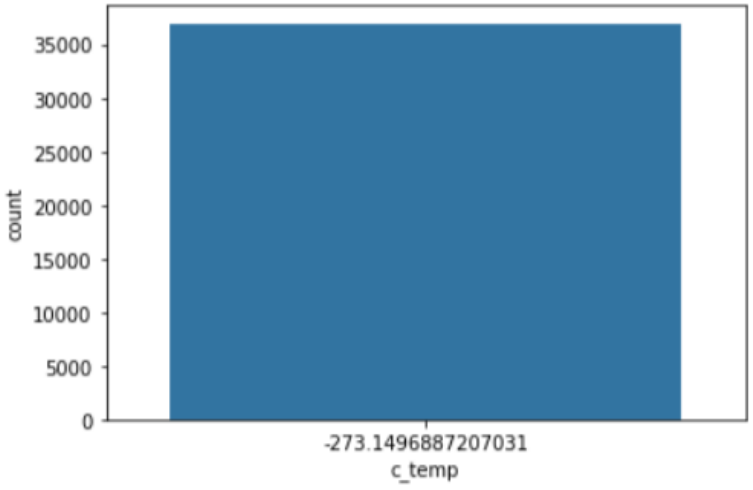
- Different methods were used to explore the relationship between each covariate and target (y).

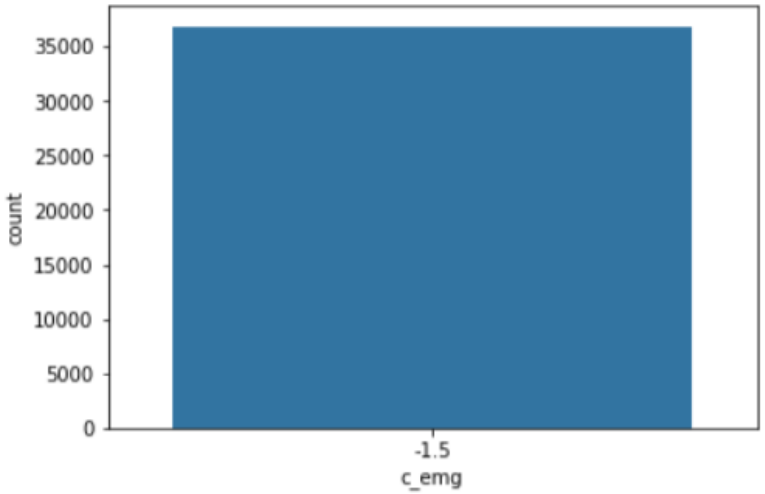


2. Data visualization (feature selection)

- Pair plot shows the relationship between variables of c_acc1, c_acc2, c_acc3 (3D-ACC of chest)
- Heatmap of all data





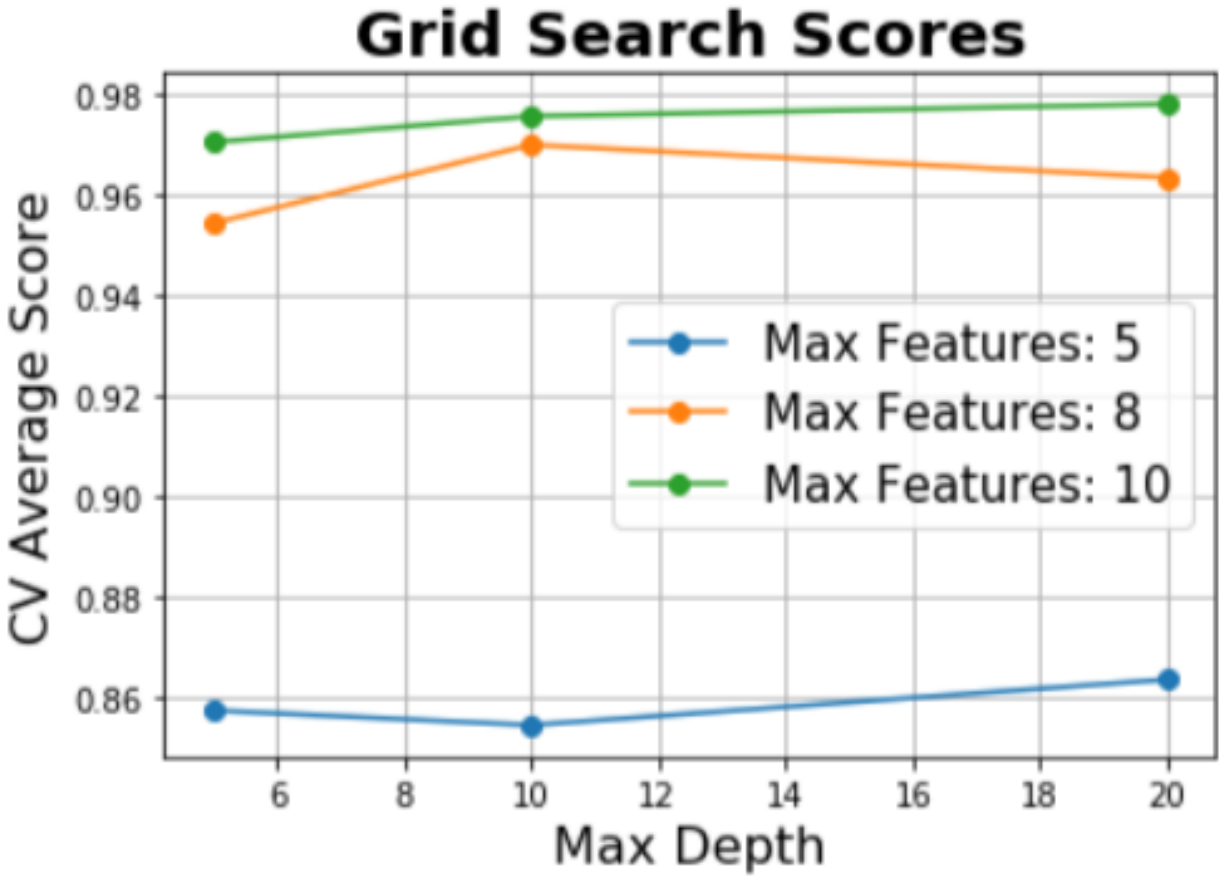


2. Data visualization (feature selection)

- I found three variables (EDA chest, EMG chest, and temperature chest) may probably useless when building model, because all records of each variables are the same. I drop these three columns from the dataset.

3. Methods

- I used 3 algorithm including Random Forest, Decision Tree and Logistic regression to build three different models. The former two algorithm used grid search to choose the best hyperparameters (detailed code can be seen in “processing.ipynb”).
- Training and testing dataset were splited based on default ratio of 0.25.



Visualization of Grid Search of Decision Tree

4. Result: Very high accuracy of RF and Decision tree

```
(0.9881679354619392,
 RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
             max_depth=20, max_features=5, max_leaf_nodes=None,
             min_impurity_decrease=0.0, min_impurity_split=None,
             min_samples_leaf=1, min_samples_split=2,
             min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=1,
             oob_score=False, random_state=None, verbose=0,
             warm_start=False))
```

```
(0.9769503978423033,
 DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=20,
             max_features=10, max_leaf_nodes=None,
             min_impurity_decrease=0.0, min_impurity_split=None,
             min_samples_leaf=1, min_samples_split=2,
             min_weight_fraction_leaf=0.0, presort=False, random_state=None,
             splitter='best'))
```

```
from sklearn.linear_model import LogisticRegression
lrmodel = LogisticRegression(random_state=0).fit(X_train, Y_train)
lrmodel.score(X_test, Y_test)
```

```
0.825119409465914
```

5. Discussion

- Limitations of this project:
- 1). I didn't normalize and standardize variables, which could lead to decrease of accuracy (The accuracy is already satisfied with almost 99% of accuracy on test dataset).
- 2). Further feature engineering work could be applied. The relationship between covariates could be interesting, future work should focus on this issue. Also the MDA or MDI of random forest can used to further remove irrelevant features.
- 3). Deep learning algorithms such CNN, RNN, could also be applied on this dataset, which may find new patterns of the dataset (For example, is the previous movement affect prediction?).

localhost:5000/index

	y	c_acc1	c_acc2	c_acc3	c_ecg	c_eda	c_emg	c_temp	c_resp	w_acc1	w_acc2	w_acc3	w_bvp	w_eda	w_temp
0	0	0.852157	-0.066488	-0.367784	0.035650	0	-1.5	-273.149689	4.834438	-0.763672	-0.076172	0.669922	0.138125	4.722437	32.13
1	0	0.851262	-0.066655	-0.370442	0.103145	0	-1.5	-273.149689	4.238839	-0.763672	-0.078125	0.671875	-26.262500	4.728843	32.16
2	0	0.851570	-0.064677	-0.370251	0.099101	0	-1.5	-273.149689	2.058559	-0.761719	-0.078125	0.671875	-40.959375	4.718594	32.16
3	0	0.852142	-0.065486	-0.371278	-0.170757	0	-1.5	-273.149689	-0.463981	-0.753906	-0.078125	0.671875	24.173125	4.717312	32.16
4	0	0.851958	-0.065894	-0.370466	-0.036633	0	-1.5	-273.149689	-1.867397	-0.761719	-0.076172	0.671875	25.781250	4.713469	32.16
5	0	0.853454	-0.066655	-0.367393	0.027540	0	-1.5	-273.149689	-1.400373	-0.763672	-0.078125	0.671875	8.678125	4.712188	32.15
6	0	0.848446	-0.068993	-0.377715	0.119724	0	-1.5	-273.149689	0.611511	-0.755859	-0.076172	0.673828	-17.215625	4.716032	32.15
7	0	0.848849	-0.063319	-0.363015	0.134410	0	-1.5	-273.149689	2.554940	-0.765625	-0.070312	0.671875	-35.286875	4.704501	32.15
8	0	0.847730	-0.070083	-0.379448	-0.212572	0	-1.5	-273.149689	2.310625	-0.751953	-0.078125	0.671875	21.870000	4.698095	32.15
9	0	0.848426	-0.065433	-0.378317	-0.074286	0	-1.5	-273.149689	0.124163	-0.755859	-0.078125	0.671875	28.181250	4.700657	32.15
10	0	0.851305	-0.064755	-0.372001	0.011253	0	-1.5	-273.149689	-1.979283	-0.761719	-0.076172	0.671875	10.370625	4.695532	32.15
11	0	0.850046	-0.065035	-0.370826	0.024181	0	-1.5	-273.149689	-2.932199	-0.759766	-0.078125	0.671875	-11.989375	4.691689	32.15

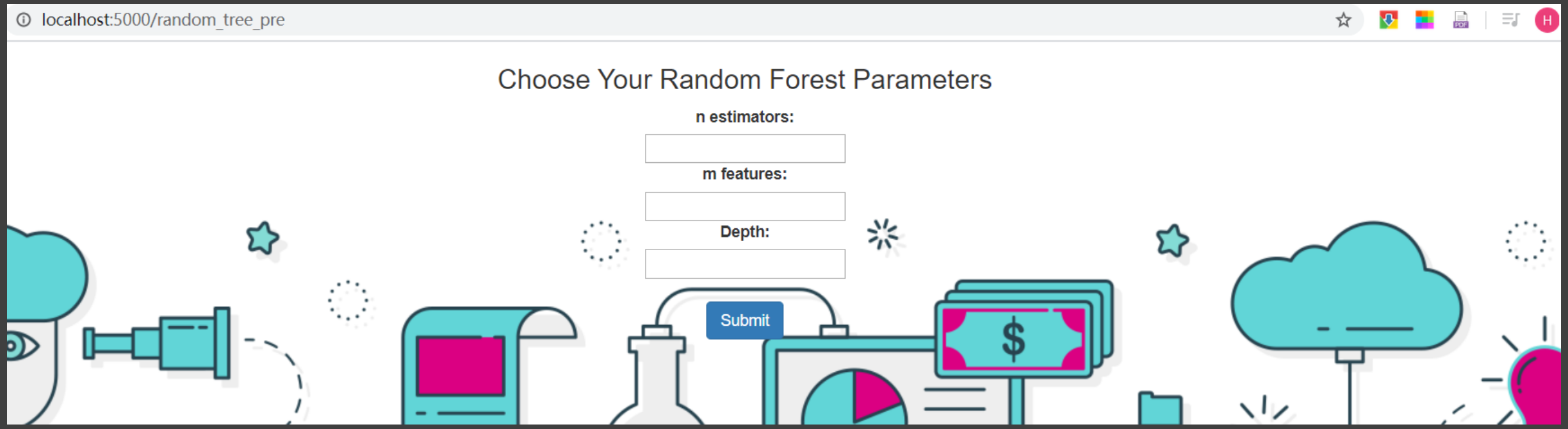
5. Flask API

- My flask API can be used to interact with stakeholders on web (If it doesn't work on your computer, please do not hesitate to contact qiugehhht@hotmail.com or ge.qiu@edu.devinci.fr).
- There are totally 4 functions in this API. 1). Display preprocessed dataset.

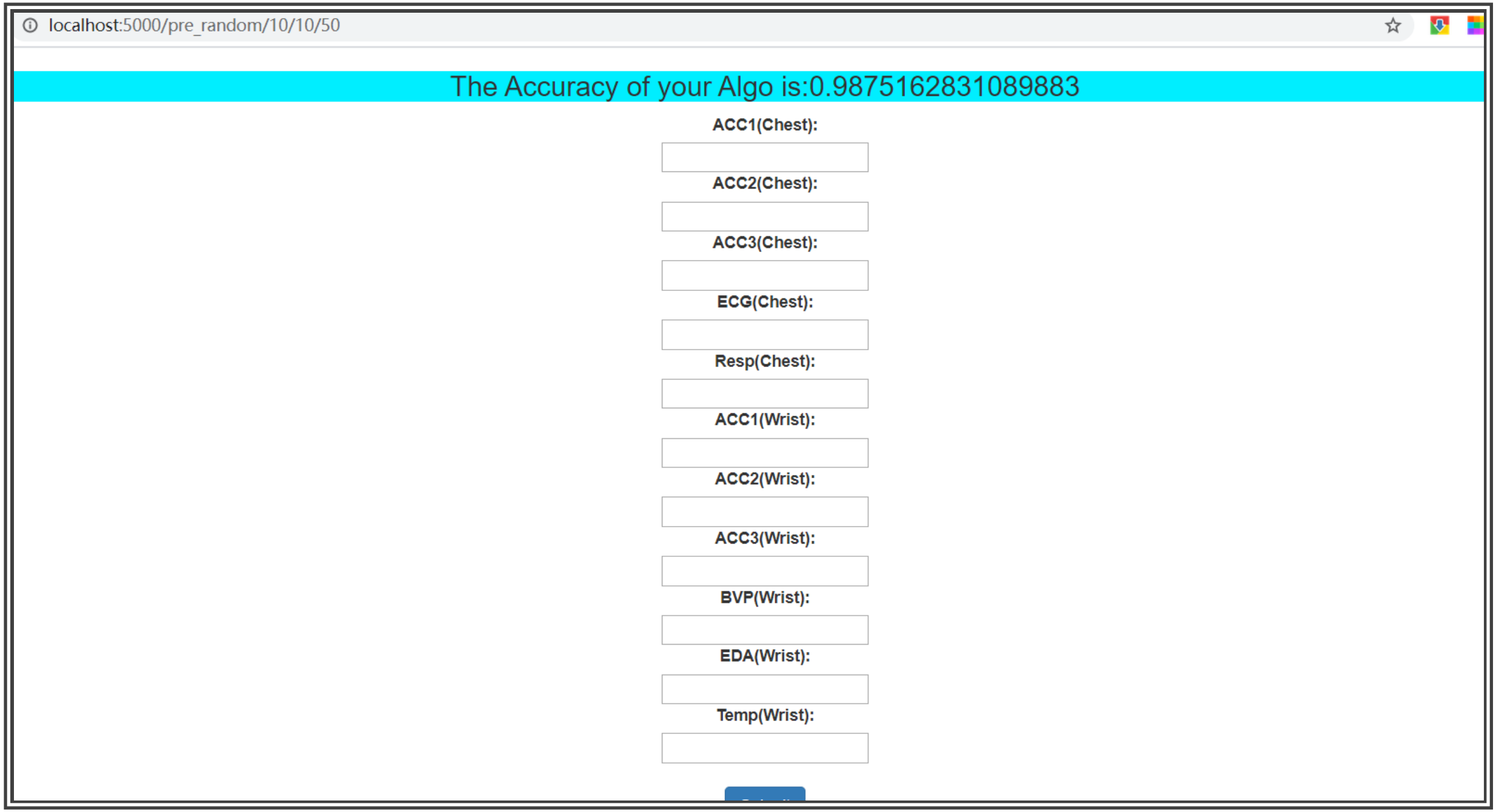
localhost:5000/decision

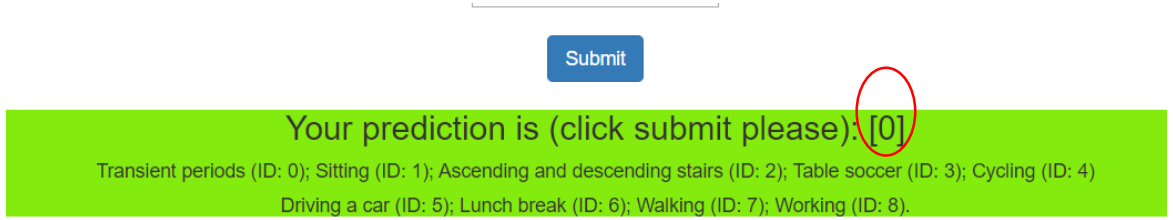
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini', max_depth=20, max_features=10, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, presort='deprecated', random_state=None, splitter='best')

- 2). Display results of Grid search, and help stakeholders to select best hyperparameters.



- 3). Input user defined hyperparameters and return the accuracy of the model using these hyperparameters.


localhost:5000/pre_random/10/10/50
The Accuracy of your Algo is:0.9875162831089883
ACC1(Chest):
ACC2(Chest):
ACC3(Chest):
ECG(Chest):
Resp(Chest):
ACC1(Wrist):
ACC2(Wrist):
ACC3(Wrist):
BVP(Wrist):
EDA(Wrist):
Temp(Wrist):



- 4). Input user defined data and make prediction of target (activity)

Acknowledge

Thank you for your time on evaluating my project, and hope you all the best!

PS: I'm not an expert of git, so I unloaded my initial code "preprocess.ipynb" on 13 Jan to prove that I was continually work on this project.