Assignment 3: heart disease diagnosis using ECG

Background:

In assignment 2, you have developed a method to find the R-peaks of a long ECG signal.

Using that method, we can divide a long ECG signal into small segments.

Each ECG segment (small ECG signal) contains only the data points during a single heartbeat.

Then, we can make heart disease diagnosis by using each of these small ECG signals.

Here, the disease diagnosis is to classify each small ECG signal into five classes

class-0: normal heartbeat

class-1: disease type 1 (e.g. Atrial premature)

class-2: disease type 2 (e.g. Premature ventricular contraction)

class-3: disease type 3 (e.g. Fusion of ventricular and normal)

class-4: disease type 4 (e.g. Fusion of paced and normal)

The good news is that: someone has already done ECG signal segmentation for us

see this webpage about the data

https://www.kaggle.com/shayanfazeli/heartbeat

Steps to complete the task (read the file basic_steps_classification_cv.ipynb)

step1: load the data using Pandas

X=pd.read_csv('ECG_dataX.csv'); Y=pd.read_csv('ECG_dataY.csv')

```
X.head()
```

	0	1	2	3	4	5	6	7	8	9	...	177	178	179	180
0	0.977941	0.926471	0.681373	0.245098	0.154412	0.191176	0.151961	0.085784	0.058824	0.049020	...	0.0	0.0	0.0	0.0
1	0.960114	0.863248	0.461538	0.196581	0.094017	0.125356	0.099715	0.088319	0.074074	0.082621	...	0.0	0.0	0.0	0.0
2	1.000000	0.659459	0.186486	0.070270	0.070270	0.059459	0.056757	0.043243	0.054054	0.045946	...	0.0	0.0	0.0	0.0
3	0.925414	0.665746	0.541436	0.276243	0.196133	0.077348	0.071823	0.060773	0.066298	0.058011	...	0.0	0.0	0.0	0.0
4	0.967136	1.000000	0.830986	0.586854	0.356808	0.248826	0.145540	0.089202	0.117371	0.150235	...	0.0	0.0	0.0	0.0

5 rows × 187 columns

```
Y.head()
```

	class_label
0	0
1	0
2	0
3	0
4	0

Convert X and Y to numpy array. Each row of X contains an ECG signal in a single heartbeat. The class label of a signal is in the corresponding row of Y. e.g. Y[0] is the class label of X[0]

X only contains signal values, NOT time points.

To get timepoints, we need sampling frequency from https://www.kaggle.com/shayanfazeli/heartbeat

fs=125 # sampling frequency

Ts=1/fs # sampling interval

N=187 # the number of timepoints

Duration=N*Ts # duration of a signal

t=np.linspace(0, Duration-Ts, N) # array of timepoints

to use sk-learn for classification, class label should be stored in a 1D array (reshape Y)

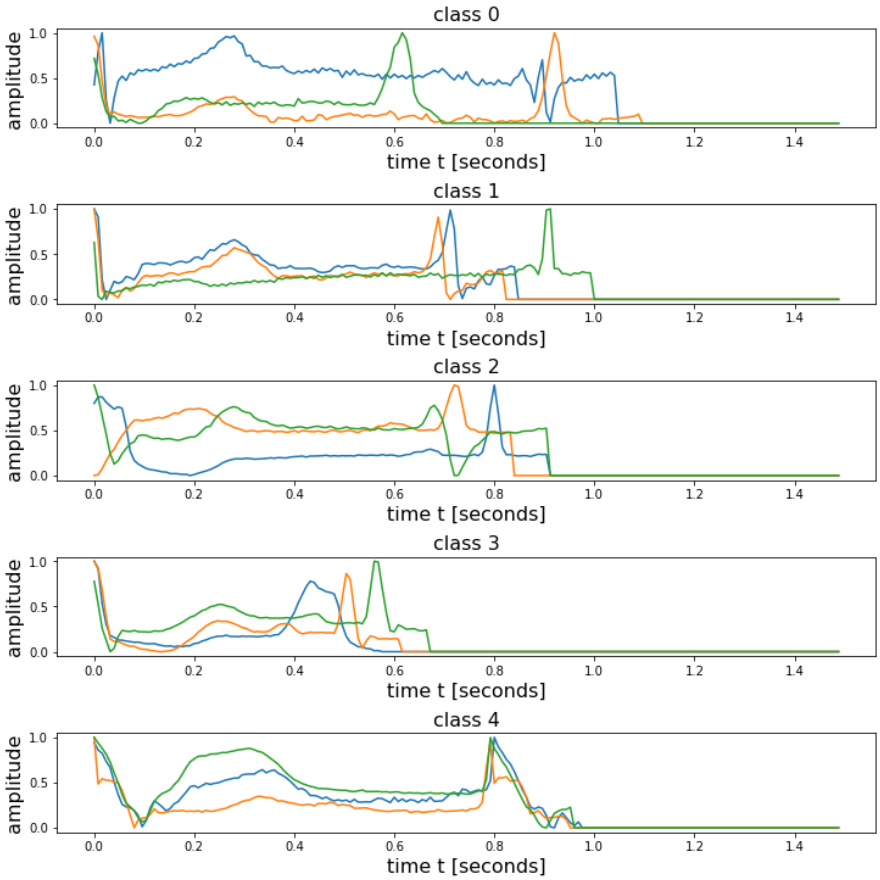
step2: visualize the data

show three signals (you can pick any three signals) per class using random color

the figure has five regions: something like plt.subplots(5,1)

set constrained_layout=True in subplots

adjust figsize in subplots



step3: train-test split to get training and test datasets

80% for training and 20% for testing, set random_state=0 in train_test_split so that the result is reproducible

step4: build a KNN classifier to predict class label of a signal

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

define a KNN model and set **n_neighbors=10**

train the model on the training set

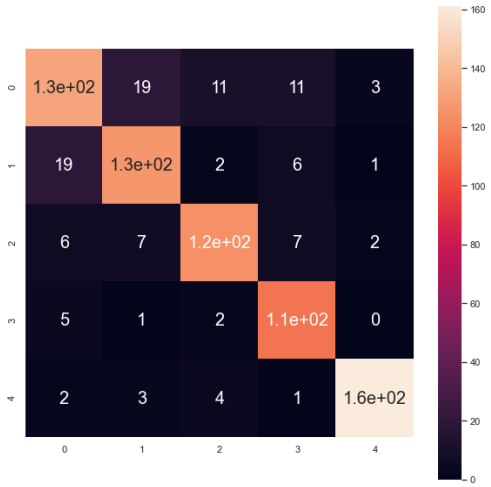
make predictions on training and test sets

```
Y_train_pred = model.predict(X_train)
Y_test_pred = model.predict(X_test)
```

print classification accuracy on the training set (a.k.a. training accuracy)

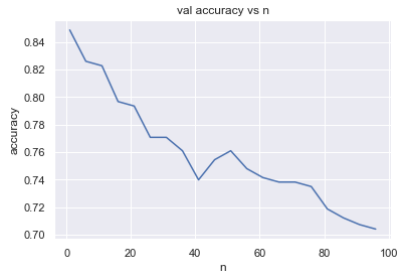
print classification accuracy on the test set (a.k.a. test accuracy)

show the confusion matrix on the test set using a heatmap



Bonus (10 points): find the optimal value of the hyper parameter: n_neighbors
read: basic_steps_classification_cv.ipynb

split training set into a 'pure' training set (80%) and a validation set (20%), and then apply GridSearchCV



The optimal value may be 1. Then build and evaluate a model with the optimal parameter.