

Slide set 2 exercises

NOTE : One zip package can contain all the exercise codes, there is no need to do separate zip packages, just one for the exercise set.

NOTE2: The classifier/predictor codes are expected to have training and testing phases and thus the data must be divided (close to) 80% / 20% ratio for training and testing.

Exercise 1

There is a KNN classifier (K-means) presented in the slides. Explain verbally how the code must be changed to classify time series. How the time series should be presented to it? What precautions must be done?

Max 2 p.

Exercise 2

Download and modify 'SVM_example.py' from file folder to show the actual recognition accuracy to all (four) included models. Use loop-like structure for different models. You can copy/mimic the recognition accuracy idea from 'Exercise_train_test.py' from practical time series classification example. Show the results in your analysis and include the zipped code.

Max 3 p.

Exercise 3

Download and modify 'SVM_example.py' to contain 3 x 3 confusion matrix mechanism described in <https://onestopdataanalysis.com/confusion-matrix-python/>. Show the result image in your analysis and include the zipped code.

Max 3 p.

Exercise 4

Improve the recognition result of SVM_example.py after the Exercise 1 and/or 2. There are four different models in the code, choose one and show the results (in numbers or confusion matrix) in your analysis and include the zipped code.

Max 3 p.

Exercise 5

Modify the ‘Decision_tree_example.py’ to train the model with iris data properly and not with the cross validation. Verify the result with real classification accuracy, like in exercise 1. For improving the performance, take notes from:

<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>

Verbally:

1) What the ‘cross validation’ means?

Max 3 p.

EXTRA: Exercise 6.

Download the time series classification database ‘pip install aeon’

<https://www.timeseriesclassification.com/dataset.php>

Choose one of the datasets from the link list and train the decision tree structure **or** random forest, copy either

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

or

<https://machinelearningmastery.com/random-forest-for-time-series-forecasting/>

(You can also copy bits and pieces from one and attach them to another, if necessary)

to learn the selected dataset. Show the classifier accuracy in percents **and** confusion matrix result in your detailed analysis. What modifications you had to make to improve the initial result?

Max 10 p.