University of Innsbruck – Department of Computer Science

Methods of Software- und Data Engineering

Prof. Dr. Michael Felderer, Yakup Ipek

EXERCISE SHEET 3

DEADLINE:

31 October 2018 - 3 p.m.

TOPICS:

Machine Learning (Clustering, Classification), Visualization

DESCRIPTION:

The following exercises allow you to gain hands-on experience with some machine learning techniques.

SUBMISSION:

Please upload your solution in OLAT according to the following instructions:

Solve the tasks within either a single or multiple notebook files. Add explanations to your solutions using the Markdown cell. Feel free to use any feature provided by Jupyter Notebook to enhance your solution. Name your solution files according to the following designation rules:

Notebook-File:

When submitting your solution please adhere to the following structure: <Surname>_ExSheet_<Sheet Nr.>_<Task Nrs. Separated with a '–'>.<file-ending>; e.g. Ipek_ExSheet_1_1.ipynb, Ipek_ExSheet_1_1-2-3.ipynb

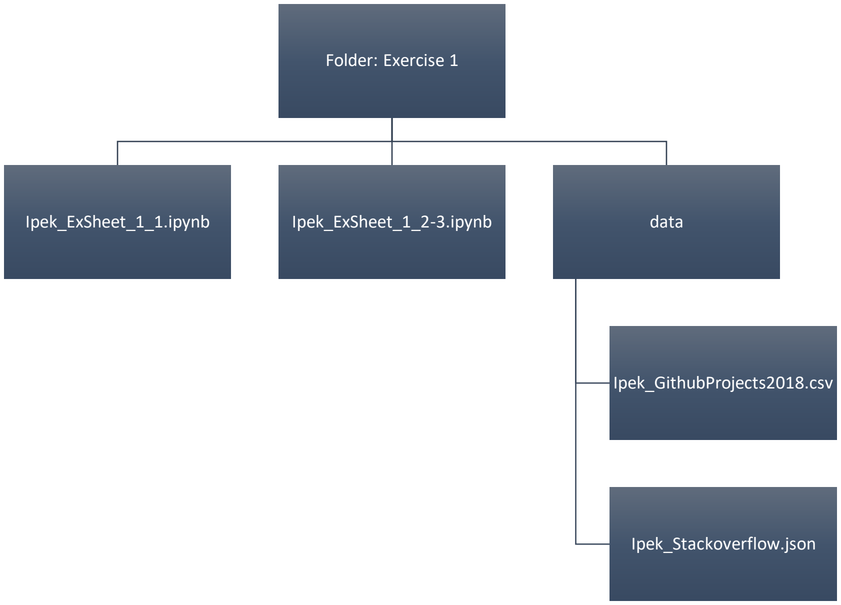
Data (CSV, JSON, etc.):

Any data source on your local filesystem that is explicitly accessed within your scripts should be placed within a **data** folder. The naming convention follows a slightly different structure than above: <Surname>_<Desired designation>.<file ending>

E.g. Ipek_GithubProjects2018.csv or Ipek_GithubProjects2018.json

PLEASE SUBMIT YOUR SOLUTION AS A SINGLE ZIPPED FILE AND NAME IT ACCORDINGLY:

<Surname>_ExSheet_<Sheet number>.zip/tar.gz; e.g. Ipek_ExSheet_1.zip or Ipek_ExSheet_1.tar.gz



LINKS:

Scikit Learn	http://scikit-learn.org/stable/ https://stackapi.readthedocs.io/en/latest/
arff file-format	https://www.cs.waikato.ac.nz/ml/weka/arff.html
Plotly	https://plot.ly/python/
Jupyter Notebook Shortcuts	https://www.dataquest.io/blog/jupyter-notebook-tips-tricks-shortcuts/

PREREQUISITES:

In this exercise sheet the following python packages are required and need to be installed:

- jupyter
- numpy
- liac-arff
- sklearn
- plotly

Task 1.1: Classification 2P

Required: defect.arff

Divide your dataset in two halves, the first half is dedicated for training purposes and the second for testing. Apply the ***DecisionTreeClassifier*** on your training dataset to build a classifier. Predict the classes for the testing dataset using the model.

Task 1.2: Confusion Matrix (1/2) 1P

Create a confusion matrix to evaluate the accuracy of your classifier and interpret the result.

Task 1.3: Visualization (1/2) 1P

Visualize the predicted and actual classes of the samples using a stacked bar chart.

Task 2.1: Clustering 2P

Required: defect.arff

Divide your dataset in two halves, the first half is dedicated for training purposes and the second for testing. Apply the ***KMeans*** algorithm to build a model with **2 clusters** from the training set afterwards predict the cluster affiliation for each of the sample of the training set.

Task 2.2: Confusion Matrix (2/2) 1P

Create a confusion matrix to evaluate the accuracy of your cluster model and interpret the result.

Task 2.3: Visualization (2/2) 1P

Visualize the predicted and actual classes of the samples using a stacked bar chart.

Task 3: Questions 2P

1. In Task 1.1 and 2.1 the training set accounted for a percental share of 50% of the data set, explain whether this was a good choice or not.
2. Unsupervised Learning requires to determine the number of clusters. What would be a good method to figure out a good cluster number?
3. What is the difference between Supervised and Unsupervised Learning?