Exercise: Clustering

Applied Data Analytics and Machine Learning in Python

In this exercise, you will use the python library *sklearn* to implement clustering using different algorithms. The data will be randomly generated by *sklearn*. You also need to visualize the clustering result.

Further instructions for this task are given in Exercise_Clustering_Template.py.

After completing the task, you can try change the hyperparameters to observe their impact on data generation and classification results.

Exercise: Regression

Applied Data Analytics and Machine Learning in Python

In this exercise, you will use the python library *sklearn* to implement regression using different algorithms and visualize the results. In addition, you need to use *pipeline* to assemble several steps of the regression workflow i.e. preprocessor and regressor.

Further instructions for this task are given in Exercise_Regression_Template.py.

After completing the task, you can try change the components in the pipeline to experience the impact of different processor combinations.

Exercise: MNIST classification

Applied Data Analytics and Machine Learning in Python

"The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems. The database is also widely used for training and testing in the field of machine learning. […] The MNIST database contains 60,000 training images and 10,000 testing images."

(Source text and image: https://en.wikipedia.org/wiki/MNIST_database)

In this exercise you will use the tensorflow-based python library *keras* to design and train a deep neural network to classify the images of the MNIST database. All images are greyscale images with a size of 28x28 pixels.

Further instructions for this task are given in MNIST_Classification_Template.py.

After completing the task, you can try different architectures for the neural net. E.g., increase/decrease the number of hidden neurons or epochs or add some hidden layers. Some possible architectures including their error rate can be found in the Wikipedia entry mentioned above.