Please carefully read and follow the general instructions regarding computing assignments. Failing to meet the requirements might lead to penalties. https://moodle.uef.fi/mod/page/view.php?id=2775059
If you suspect that something is wrong with some task instructions, please contact the lecturer.
If you face persistent issues while working on a task, do ask for help, e.g. during a course meeting or by contacting the lecturer via email.

Datasets

iris-SV-sepal.csv  irisSV, two-dimensional binary classification dataset, linearly separable
iris-VV-length.csv  irisVV, two-dimensional binary classification dataset, not linearly separable
creditDE.csv  credit, multi-dimensional binary classification dataset, a variant of the Statlog dataset from https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)

The irisSV and irisVV datasets are variants of the iris dataset from https://archive.ics.uci.edu/ml/datasets/Iris.

Tools

classification_resources.py  some potentially useful code snippets.

! Imports of external libraries other than those that appear in the classification_resources.py file are not allowed.

Task 1. Implement the linear SVM algorithm with hard-margin and soft-margin variants (you may use snippets from classification_resources.py, i.e. fill the dots in the provided code). Apply them respectively to the irisSV and irisVV datasets.

That is, divide the irisSV dataset into training and test subsets in proportions 4/5–1/5 at random, i.e. assign one fifth of instances, chosen at random, to the test dataset and the rest to the training dataset. Train a hard-margin SVM on the training subset, and apply the resulting model to the test subset.

Write down the confusion matrix and compute the accuracy, recall and precision.

Give the equation of the separating hyperplane. Plot the separating hyperplane and highlight the support vectors.

Do the same with soft-margin SVM on the irisVV dataset (setting $c = 2$, for example).

Compare your results to the results obtained in the Python notebook #3.

Task 2. Implement the RBF kernel for the SVM algorithm (you may use snippets from classification_resources.py).

Divide the credit dataset into training and test subsets in proportions 3/4–1/4 at random, i.e. assign one fourth of instances, chosen at random, to the test dataset and the rest to the training dataset.

Plot the ROC curve and compute the AUC for the linear SVM and for the SVM with RBF kernel (setting $\sigma = 1$, for example).