Lab of Applied Computational Intelligence

IST

2024/2025

Some Examples in Scikit-learn

Guide 4

20 September 2024

(Week 2)

1 – Objetives

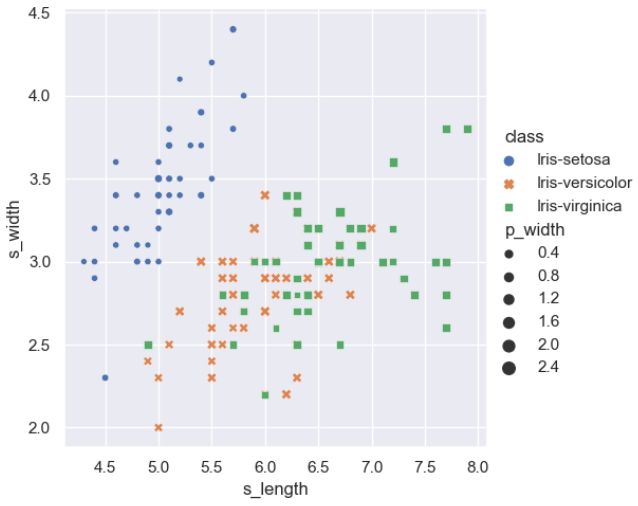
With this work the student should be able to learn how to work with Scikit-learn.

2 – Data Libraries

For this work we are going to use two popular datasets, “iris” and “Haberman”. The first has information about 3 types of iris flowers and the second about survivability of cancer. See what information is in the datafiles. You will see that the file does not have the name of the columns, so you will have to add manually those names. Read the datafiles with pandas. And use the following function (head, tail and dtypes) just to see some elements of the database and their types.

3 – Seaborn plots

To make some interesting plots we are going to use Seaborn. Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. We are going to use the function relplot() to make a plot similar to figure 1. Here the classes have different colors, and we are using 3 variables (s_width, p_width, s_length) in a two dimension plot to better visualize the data. Now plot several of similar figures with different variables for the datasets used in this lab.



4 – Scikit library

Now we are going to use the Scikit-learn library to try to identify the class of a sample from our database. To do that we need to train a model with part of the data and then test the model with the rest of the data. Scikit-learn has many different algorithms to perform the classification of the data. But all of them have similar interfaces so we can test several algorithms with just a few lines of code.

First, we need to split the data in train and test datasets, use the function train_test_split() to achieve this goal. Find out what are the parameters “test_size” and “random_state”.

Then use the following algorithms to classify our data:

Naïve Bayes (from sklearn.naive_bayes import GaussianNB)

LinearSVC (from sklearn.svm import LinearSVC)

SVM (from sklearn import svm)

K-Neighbors (from sklearn.neighbors import KNeighborsClassifier)

Then use the function fit() to get the model that best fits your training data and then the function predict() to find the prediction of the model for your test set. Finally get the accuracy, precision, recall and confusion matrix to obtain some insight about the quality of the prediction.