TASK

Machine Learning II

Visit our website

Introduction

WELCOME TO THE SECOND MACHINE LEARNING TASK!

This task is designed to give you a better understanding of the scikit-learn library, which we will be using for the remainder of the machine learning tasks.

Get in touch

Connect for support

Remember that with our courses, you're not alone! You can contact an expert code reviewer to get support on any aspect of your course.

The best way to get help is to login to **www.hyperiondev.com/portal** to start a chat with a code reviewer. You can also schedule a call or get support via email.

Our expert code reviewers are happy to offer you support that is tailored to your individual career or education needs. Do not hesitate to ask a question or for additional support!

A note from the

HyperionDev Team

If this is your first venture into machine learning, you may be wondering why learn/use scikit-learn? Well, for starters, scikit-learn is perhaps the most popular machine learning library available for the Python programming language. This is because scikit-learn provides simple and efficient tools for data mining and data analysis. It is also an open-source library, which means you have complete license to edit, modify, and/or redistribute any code contained within the module — similarly to NumPy, Matplotlib, and SciPy. Thus, you can use the package as is or, alternatively, you can tailor the package to suit your needs. Furthermore, if you plan to pursue a career in machine learning, it is worth learning scikit-learn to start familiarising yourself with machine learning models (since scikit-learn encompasses some of the most popular and widely used machine learning algorithms) you can apply to different machine learning problems.

HOW TO INSTALL SCIPY, NUMPY, MATPLOTLIB AND SCI-KIT LEARN

- Open the command prompt or terminal
- The command for installing any package in python is **`pip install package_name.`** If you are using a Windows machine, you may need to start by installing pip using the command "**`python -m pip install`**". The following are the commands you will need:
    - To install SciPy, type "**`pip install scipy`**" and press Enter.
    - To install NumPy, type "**`pip install numpy`**" and press Enter.
    - To install Matplotlib, type "**`pip install matplotlib`**" and press Enter.
    - To install Sci-Kit Learn, type "**`pip install sklearn`**" and press Enter.

This is the easy way of installing packages. However, in the unlikely case that pip doesn't work, Windows binaries can be found **here**. To install:

- Find the package you want to install (e.g. NumPy), then select the **.whl** for the Python version that you have (e.g. cp36 for Python 3.6), and check that it has 'amd' and '64' in the name.
- Download and save this file in your site-packages folder (you can find this using `python -m site` in your command line).
- Type in **`pip install`** *`\path\to\the\file\thefilename.whl`* into your command line.

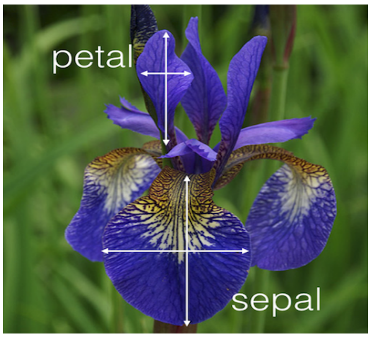
If you have problems completing the installation, contact your mentor to assist you.

A note from Riaz...

Sorry to interrupt, but did you know that scikit-learn is an industry favourite and is used by popular websites such as **Spotify**, **Inria**, **Evernote**, and more. Spotify uses machine learning to give you suggestions to your favourite songs, whereas, Inria, a public research body dedicated to science and technology, uses it to support cutting-edge research for their computer vision and medical analyses. Evernote, on the other hand, uses scikit-learn to perform classifications and predictions.

SCIKIT-LEARN EXAMPLE USING THE IRIS DATASET

Let’s load the popular iris dataset in pattern recognition using scikit-learn. Before we do that, here is a short description of the iris dataset. The iris dataset contains 3 classes of 50 instances each; a total of 150 observations of the iris flower specifying some measurements: sepal length, sepal width, petal length and petal width. Each class refers to the type of iris plant. The three classes are Setosa, Versicolour and Virginica. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other. The goal is to predict the class of the iris plant. Hence, it is said to be supervised learning.



LOADING THE DATASET

There are two ways in which you can load a dataset from `sklearn`. If you are using a slow PC or laptop this might take a few seconds.

1. The first is to use the `sklearn` module to import all datasets and then use the dot (`.`) notation to specify which dataset to load.

```
from sklearn import datasets
iris = datasets.load_iris()
```

2. Another way is to use `sklearn.datasets` module to import `load_iris`. Using this approach will omit the use of the dot (`.`) notation and just invoke the `load_iris()` function directly.

```
from sklearn.datasets import load_iris
iris = load_iris()
```

The second option is considered convenient since you load only the dataset you will be working on rather than all `sklearn` datasets, which you might not use. `Sklearn` uses the data type called "bunch", which is a container storing datasets and their attributes.

IRIS DATASET "BUNCH" ATTRIBUTES

These are the attributes contained in the iris dataset, namely `target_names`, `data`, `target`, `DESCR`, and `feature_names`. Let's print out these attributes and the dataset object type.

```
print(type(iris))
print(iris['target_names'])
print(iris['data'])
print(iris['target'])
print(iris['DESCR'])
print(iris['feature_names'])
```

You can print out the dataset object with all its attributes and type by typing:

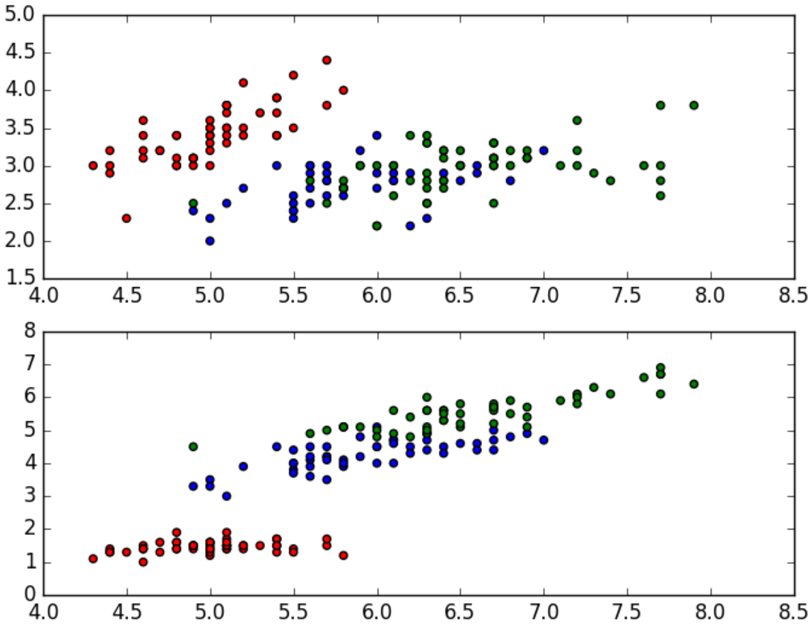
```
print(iris) #iris is the dataset object
```

SCATTERPLOT OF THE IRIS DATASET

Let's make a scatterplot of the iris dataset. The first thing to do is to import any packages we want to use i.e. **matplotlib.pyplot**, **sklearn.datasets** and **numpy**. In lines 5 to 7, a figure and subplots are created. In lines 9 to 11, an iris dataset is loaded to the "**iris**" object and then the **iris** object is used to extract the information in data and target attributes. In line 16, a dictionary is created to reference colours for the 3 classes. The reason why 0, 1 and 2 are used is that they represent the 3 classes, namely; Setosa, Versicolor and Virginica. Next, an array of colours is created so that the 3 classes are plotted in different colours. Lastly, the scatterplots are plotted in their respective axes and shown. The first scatterplot uses column 0 and 1. The second scatterplot uses column 0 and 2.

```
1    import matplotlib.pyplot as plt
2    from sklearn.datasets import load_iris
3    import numpy as np
4
5
6    fig = plt.figure()
7    ax1 = plt.subplot(2,1,1)
8    ax2 = plt.subplot(2,1,2)
9
10   
11   iris = load_iris()
12   data = np.array(iris['data'])
13   targets = np.array(iris['target'])
14
15
16   cd = {0:'r',1:'b',2:"g"}
17   cols = np.array([cd[target] for target in targets])
18
19   ax1.scatter(data[:,0], data[:,1], c=cols)
```

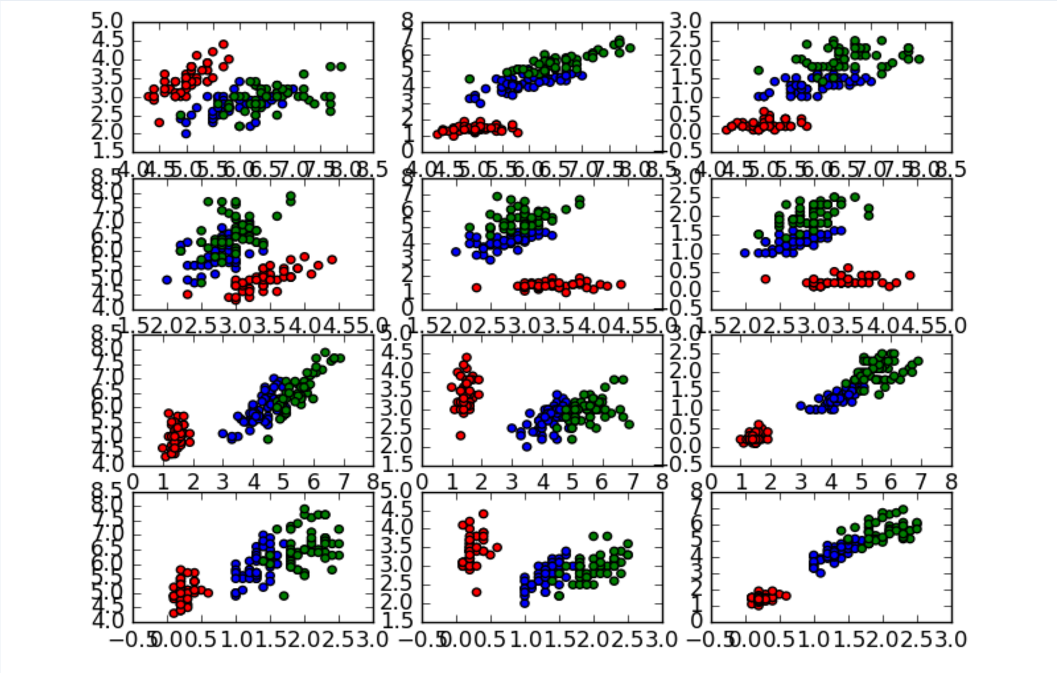
```
ax2.scatter(data[:,0], data[:,2], c=cols)
plt.show()
```



Compulsory Task

Follow these steps:

- Read **Example.py**
- Create a python file called **IrisDatasetPlot.py**.
- Write code to produce the following figure and save your figure.



- *Hint: Note that there are 2 features being used for each subplot. I .e.The first 2 subplots came from our example of plotting the iris dataset. This is a permutation problem. From 4 features choose 2 which equals 12 choices in total. That is why there are 12 subplots in total.*

Completed the task(s)?

Ask an expert to review your work!

Review work

Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved or think we’ve done a good job?

Click here to share your thoughts anonymously.