ITCS159 Software Lab for Basic Scientific Problem Solving
Lab 10: the use of Python Library
(Scikit-learn: Machine Learning in Python Part2)

October 18, 2021

Introduction

In Lab09, you have learned how to use `sklearn` library to perform supervised machine learning. This week you will learn how to use `sklearn` to perform unsupervised machine learning. Let's recall the different between these two type of machine learning as follow:

- **Supervised learning**: is a machine learning techniques that the system will try to learn from the previous example that is trained or labeled from existing data. Hence, any machine learning techniques deal with labeled data is called supervised learning.
- **Unsupervised learning**: is a machine learning techniques that the system attempts to find pattern directly from the given data. Hence, any techniques deal with unlabeled data is called unsupervised learning.

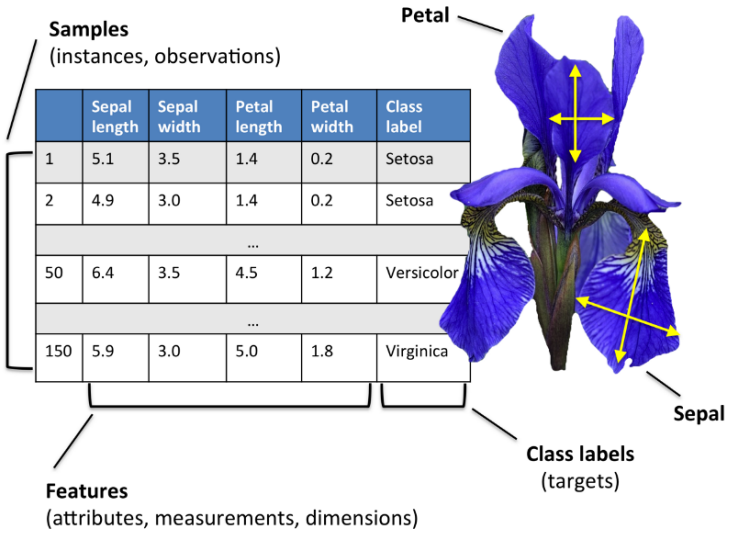
To start learning how to use `scikit-learn` for finding pattern of data using unsupervised learning method. The Iris dataset will be used as previous week. Again, do not forget to download the file named *'iris.data'* from the Mycourse website and place it at the same folder with the python file.

Recall: Iris Dataset Information

The Iris dataset contains 50 samples from each of three species of *Iris flower* that are *Iris setosa*, *Iris virginica* and *Iris versicolor*. Four features are used to identidy the species of each sample that are sepal and petal length; sepal and petal witdh as presented in Figure 1.

Unsupervised Machine Learning with Scikit-Learn

As mentioned earlier, unsupervised learning aim at identifying the patterns directly on the data.To perform unsupervised machine learning process, there are **Five** basic steps to follows:



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Figure 1: Example of information from Iris dataset `source:https://sebastianraschka.com/images/blog/2015/principal_component_analysis_files/iris.png`.

- Python Preparation
- Data Preparation
- Data Classification
- Measuring Performance

Python Preparation

Exercise 1) Make sure that all Python environment in your computer was installed successfully. Try the script below for checking all the important library (All versions must be more recent).

```
import sys
import scipy
import numpy
import matplotlib
import pandas
import sklearn

print('Python: {}'.format(sys.version))
print('scipy: {}'.format(scipy.__version__))
print('numpy: {}'.format(numpy.__version__))
print('matplotlib: {}'.format(matplotlib.__version__))
print('pandas: {}'.format(pandas.__version__))
print('sklearn: {}'.format(sklearn.__version__))
```

Exercise 2) Once everything ready, Then, load all the important library:

```
import pandas
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.cluster import KMeans
```

At this point, all components should be loaded properly without errors.

Data Preparation

Exercise 3) Next, load the iris dataset from file. Please ensure that the download file named 'iris.data' is located in the same folder as the python file (created from Jupyter Notebook).

```
names = ['sepal-length', 'sepal-width', 'petal-length', '
    petal-width', 'class']
ds = pandas.read_csv('iris.data', names=names)
```

Display the dataset 'ds', how it look like?

Exercise 4) Next step is to slicing the dataset based on features of iris flower.

```
array = ds.values
X = array[:,0:4]
Y = array[:,4]
```

What are the results of X and Y?

Exercise 5) Next create the scatter plot of x=Sepal length and y=Petal length.

```
plt.scatter(X[:,0], X[:,2])
```

Do you see any pattern here?

Exercise 6) Try to create another scatter plot of x=Sepal width and y=Petal width. What are the outputs?

Milestone 1) You have now completed the Milestone 1. You should get sign off by your lab assistant.

Data Clustering

Next step is to build a clustering model from the dataset using the K-Means clustering techniques. **K-means** is a clustering algorithm that aims to partition dataset into K cluster. The algorithm start with specify the number of clusters that we need to partitioning. For example, K=3. Next step is to randomly select three points(inputs) in order to represent each cluster. distance between each point and centroid will be calculated and each point will be segregated into respected clusters. Now, re-computing the centroids for all the clusters until the value of centroid is not changed.

Exercise 7) specify number of clusters. In this dataset we set cluster=3 because we know that there are 3 species of iris flowers. Then fit the model to create cluster of the dataset.

```
model = KMeans(n_clusters=3)
model.fit(X)
```

What are the output?

Exercise 8) Try to predict the iris dataset on the fitted model.

```
labels = model.predict(X)
print(labels)
```

What are the output? Do you see any useful information now?

Exercise 9) Using knowledge from week8 to plot the scatter plot between sepal length and petal length.

```
fig = plt.figure(figsize=(15,8))
fig1 = fig.add_subplot(121)
fig2 = fig.add_subplot(122)

fig1.set_title('Unclustered Data')
fig2.set_title('Clustered Data')
fig1.scatter(X[:,0], X[:,2])
fig2.scatter(X[:,0], X[:,2], c=labels)
fig2.scatter(model.cluster_centers_[:,0] ,model.
    cluster_centers_[:,2], color='red')
fig1.set_xlabel('sepal length')
fig1.set_ylabel('petal length')
fig2.set_xlabel('sepal length')
fig2.set_ylabel('petal length')
```

What are the output?

Exercise 10) Try to plot the scatter plot between sepal width and petal width by yourself. What are the output? Do you see any pattern?

Exercise 11) Now, you can try to predict only a single input.

```
predicted_label = model.predict([[7.2, 3.5, 0.8, 1.6]])
```

What are the outputs? What specie the iris flower belong to?

Milestone 2) You have now completed the Milestone 2. You should get sign off by your lab assistant.

Measuring Performance

What you have done is called unsupervised machine learning based on the K-Mean clustering model. There are many approach to evaluate the model. In this case we will measure performance of K-means towards iris flower dataset by cross-tabulation since we know the answer set.

Exercise 12) Try to align labels and species together by using the following command:

```
import pandas as pd
df = pd.DataFrame({'labels': labels, 'species': Y})
print(df)
```

What is the result of dataFrame alignment?

Exercise 13) Next, perform the cross-tabulation of labels and species together to evaluate the results of using K-means clustering:

```
ct = pd.crosstab(df['labels'], df['species'])
print(ct)
```

What is the result of cross-tabulation? What does the result mean?

Sometime, we do not have the answer set information (e.g., species), we need to measure cluster quality using only samples and their cluster labels. The *Inertia* measure can be used to measures how spread out the clusters are (lower is better). This score is calculated from each centroid of its cluster after calling **fit()** function.

Exercise 14) Try the following code to see the clustering quality:

```
print(model.inertia_)
```

What does the score mean?

Milestone 3) You have now completed the Milestone 3. You should get sign off by your lab assistant.