ITCS159 Software Lab for Basic Scientific Problem Solving
Lab 9: the use of Python Library (Scikit-learn: Machine Learning in Python)

October 11, 2021

Introduction

In this lab session, you will learn a basic python library for solving scientific problem named `sklearn`. `sklearn` is a tools for data mining and data analysis. It is built on `NumPy`, `SciPy` and `matplotlib`. In order to start using the library, it is good to understand basic concept of the machine learning first.

What is Machine Learning

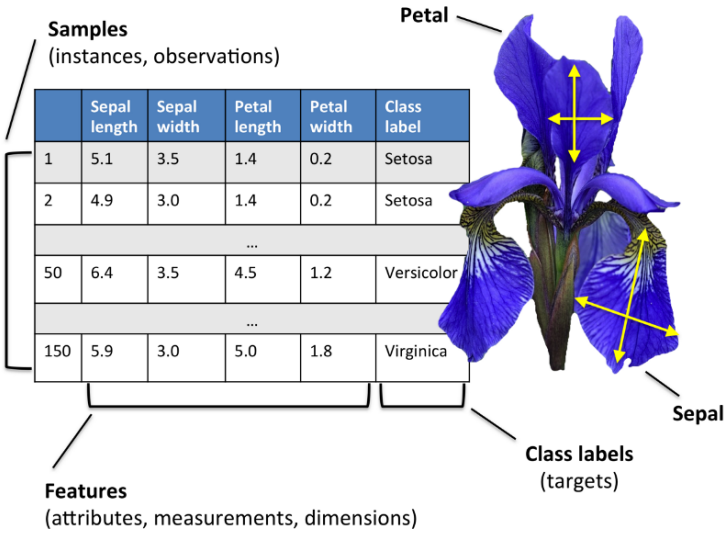
Machine learning (ML) is a field of computer science studying on how to create systems that have ability to learn from data, identify pattern and make decision on their own with minimal intervention from human.

ML can be classified into three major types that are:

- **Supervised learning**: learning from labeled data by training from existing data.
- **Unsupervised learning**: learning from unlabeled data by clustering techniques.
- **Reinforcement learning**: learning from rewards and punishments.

In this lab, you will learn how to use `scikit-learn` or `sklearn` library to do a supervised machine learning. The **Iris flower dataset** which is a classic dataset for learning ML is used.

Before start coding: Open the Web browser. Then, go to Mycourse website, then download the file name *'iris.data'* and place it at the same folder with the python file.



	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`.

Iris Dataset Information

Iris dataset contains 50 samples from each of three species of *Iris flower* that are *Iris setosa*, *Iris virginica* and *Iris versicolor*. To classify the species **FOUR** features were measured from each sample that are (1) the length of sepals (cm), (2) the width of sepals (cm), (3) the length of petals (cm) and (4) the width of petals (cm) as illustrated in Figure 1.

Basic Machine Learning with Scikit-Learn

To perform supervised machine learning process, there are **Five** basic steps to follows:

- Python Preparation
- Data Preparation
- Training Data (labeling data)
- Testing Model
- Measuring Performance

Python Preparation

Exercise 1) It is good to make sure that the 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.neighbors import KNeighborsClassifier
```

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

Data Preparation

Data preparation is a process of collecting, cleaning and consolidating data into format that ready to analyze. In this lab, the iris dataset is used and it is already prepared for analysis. Thing you need here is to load the data and understand the structure of it as follow.

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) Try the following commands to display basic information of iris dataset.

```
print(ds.shape)
print(ds.head(20))
print(ds.describe())
print(ds.groupby('class').size())
```

What is the result? What does it mean?

Exercise 5) Recall knowledge from week 7 to create a box plots on iris dataset

```
ds.plot(kind='box', subplots=True, layout=(2,2), sharex=
    False, sharey=False)
plt.show()
```

Explain the results.

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

Training Data (labeling data)

Training data is a group of data used for learning. In order to built a classification model from training data. First, we need to split dataset into two groups one is for training and one is for testing. As presented in Figure 1, iris.data contain five columns that are sepal length, sepal width, petal length, petal width, and the class label. The class label is the species of iris. In machine learning the class label is used as a *target* for classification.

Exercise 6) Next step is to separate features of iris flower from the target.

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

What are the results of X and Y?

Exercise 7) Now you should have X which contain four features of iris flowers and Y which contains target name of iris' species. Next, we need to split iris.data into two groups (one for training and one for testing).

```
test_size = 0.20
seed = 7
X_train, X_test, Y_train, Y_test = model_selection.
    train_test_split(X, Y,
test_size=test_size, random_state=seed)
```

What are the outputs? What is method `.train_test_split` do?

Exercise 8) Try to change the *test_size* and *seed* number. What will happens? What are the outputs of $X\_train, X\_test, Y\_train, Y\_test$?

Having got two set of data (Training set and Test set), the next step is to built a model for prediction from the training dataset by using machine learning techniques. The simple technique used in this example is *K-nearest Neighbors*.

K-nearest Neighbors is a machine learning techniques for both classification and regression. In this lab, K-nearest Neighbors will be used as a classification technique, where input is a K-closest training example in the feature space. The output is a class membership. The new object will be classified to the class most common among the majority vote of it k-nearest neighbors. For example,

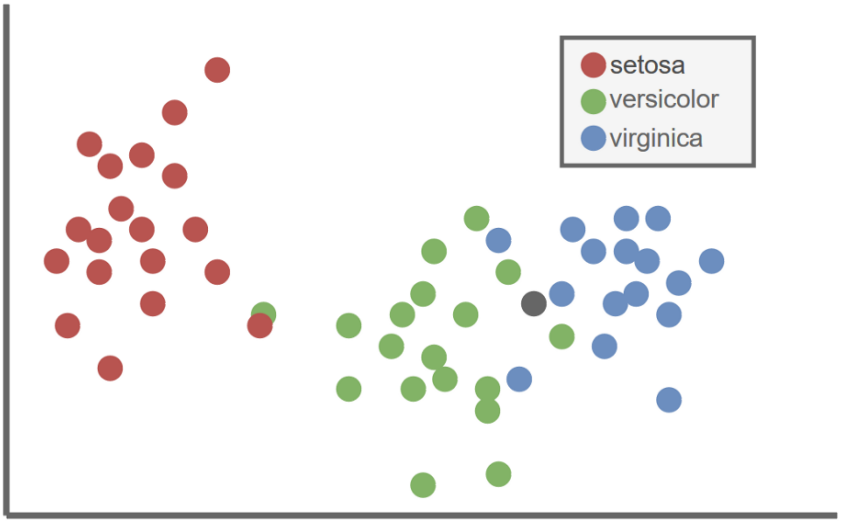


Figure 2: Example of three iris species plotted with a new iris flower object.

if k=3, then the new object is assigned to the class of three nearest neighbors. Let's see the example below:

Figure 2 illustrates the group of three iris' species plotted by four features mention earlier (length and width of both sepals and petals). The **grey** dotted is considered as a new iris flower that we want to find out what specie it is.

Exercise 9) The K-nearest Neighbors state that the new object will be classified to the majority vote of it k-nearest neighbors. Hence, if we considered k=1 and k=3 as presented in Figure 3, what did you see?
What would be the output after classification? Which one is more reliable between setting k=1 or k=3? Should k be Odd or Even number? Why?

Figure 3: The comparison between k=1 and k=3 of using K-Nearest Neighbors technique.

Exercise 10) Now, we should be ready to create a training set using the K-nearest Neighbors techniques. In this case, we use 7 neighbors to identified the new object.

```
irisML = KNeighborsClassifier(n_neighbors=7)
irisML.fit(X_train, Y_train)
```

What are the outputs?

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

Testing Model

Having finished training data, it is time to test the model by passing the test dataset to the model. This is to ensure how accurate the model is.

Exercise 11) Try the following command to pass the test dataset to the model for prediction.

```
prediction = irisML.predict(X_test)
```

What is the result of prediction?

Exercise 12) Let's add the following command to see the score of prediction model:

```
irisML.score(X_test, Y_test)
```

What does the score mean?

Exercise 13) Now, we can try input new iris object to the model in order to classify it specie:

```
new_obj1 = [[5.2,4.3,5.6,1.6]]
output1 = irisML.predict(new_obj1)
```

Finally, Can you input four features that will result in *'Iris-setosa'* specie?

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