Name: Qasim Khan
CMSID: 49375

Lab. 1_____06/10/2022

Introduction to Python Libraries

1. Objectives:

After completing this lab, Students will be able to;

- Understand Python libraries
- Understand the Python's Dataset

2. Corresponding CLO and PLO:

- CLO 1, PLO 5

3. Theory:

Python Libraries:

Normally, a library is a collection of books or is a room or place where many books are stored to be used later. Similarly, in the programming world, a library is a collection of precompiled codes that can be used later on in a program for some specific well-defined operations. Other than pre-compiled codes, a library may contain documentation, configuration data, message templates, classes, and values, etc. A Python library is a collection of related modules. It contains bundles of code that can be used repeatedly in different programs. It makes Python Programming simpler and convenient for the programmer. As we don't need to write the same code again and again for different programs. Python libraries play a very vital role in fields of Machine Learning, Data Science, Data Visualization, etc.

Working of Python libraries:

A Python library is simply a collection of codes or modules of codes that we can use in a program for specific operations. We use libraries so that we don't need to write the code again in our program that is already available. But how it works. Actually, in the MS Windows environment, the library files have a DLL extension (Dynamic Load Libraries). When we link a library with our program and run that program, the linker automatically searches for that library. It extracts the functionalities of that library and interprets the program accordingly. That's how we use the methods of a library in our program

Python libraries Examples:

- Numpy
- Pandas
- Tensor flow
- Sklearn
- Keras
- Scipy

Name: Qasim Khan
CMSID: 49375

4. Equipment's:

- PC or Laptop
- Jupyter Notebook

5. Procedure:

- First we have to open Jupyter NoteBook to write Python code
- To launch a Jupyter notebook, open your terminal and navigate to the directory where you would like to save your notebook
- Then type the command jupyter notebook and the program will instantiate a local server at localhost:8888 (or another specified port).
- A browser window should immediately pop up with the Jupyter Notebook interface, otherwise, you can use the address it gives you
- in the Jupyter Notebook interface, and you can see all of the files in your current directory
- To create a new notebook, go to **New** and select **Notebook - Python 3**
- Cells are how notebooks are structured and are the areas where you write your code. To run a piece of code, click on the cell to select it, then press SHIFT+ENTER or press the play button in the toolbar above
- After your run a cell, the output of the cell's code will appear in the space below. To stop running a piece of code, press the stop button.

6. Observations:

In this lab we learned about how to use python libraries in our python code. We learned about datasets in labs tasks. Now we are able to use python libraries in our python code.

7. Lab Tasks/Assignment:

Task 1: Create a Linear Dataset

Your goal is to create a simple dataset consisting of a single feature and a label as follows:

- Assign a sequence of integers from 6 to 20 (inclusive) to a NumPy array named feature.
- Assign 15 values to a NumPy array named label such that:

Snapshots:

Code:

```
import numpy as np
feature = np.arange(6, 21)
print(feature)
label = (feature * 3) + 4
print(label)
```

Figure 1 Task 1 (Code)

Name: Qasim Khan
CMSID: 49375

Output:

```
[ 6  7  8  9 10 11 12 13 14 15 16 17 18 19 20]
[22 25 28 31 34 37 40 43 46 49 52 55 58 61 64]
```

Figure 2 Task 1 (Output)

Task 2: Add Some Noise to the Dataset

- To make your dataset a little more realistic, insert a little random noise into each element of the label array you already created. To be more precise, modify each value assigned to label by adding a *different* random floating-point value between -2 and +2.
- Don't rely on broadcasting. Instead, create a noise array having the same dimension as label.

Snapshots:

Code:

```
noise = (np.random.random([15]) * 4) - 2
print(noise)
label = label + noise
print(label)
```

Figure 3 Task 2 (Code)

Output:

```
[-1.23464776  1.93997833  1.58754278 -1.30213315  1.57071026  1.14559956
 -0.45835645 -1.72122176  0.74343461  0.11895447 -1.37305733  1.46107659
  1.00290929  0.77250068  0.23289265]
[18.82815506 25.35124871 29.92249875 29.37294403 37.53638112 38.0077609
 39.86915968 40.01690992 46.39634177 48.36522965 50.95373596 58.01272679
 59.28442874 63.24233248 65.32433049]
```

Figure 4 Task 2 (Output)

Name: Qasim Khan
CMSID: 49375

Task 3:

- Create a dataset with 2 features with 30 values/exaples each generating randomly between -10 to +10 inclusive
- Create a label for these features such that each label is twice the sum of features

Snapshots:

Code:

```
import numpy as np
import matplotlib.pyplot as plt
features=(np.random.random([30,2])*20)-10
feature1=(features[:,0])
feature2=(features[:,1])
print(feature1)
print(feature2)
label=(feature1+feature2)*2
print(label)
plt.scatter(feature1,label)
plt.scatter(feature2,label)
plt.show()
```

Figure 5 Task 3 (Code)

Output:

```
[ 7.80283658 -6.40453857  8.25901268  4.41556267  7.0726599  -0.22547998
  4.68178219  3.11685339  6.74494659 -9.8329049  -0.03635661  4.70642853
 -0.67204747  8.75535176  9.41012751  5.31120577 -8.45852772  2.29402984
  5.8554843  -9.82642311 -4.10099033  0.76525024 -9.04301315 -1.14012617
 -9.00283775 -3.83781615  4.33270507 -3.35849953 -0.9349376   0.81548826]
[ 2.65479724 -4.62055919 -6.00484074 -7.45044529 -1.73158076 -5.2042009
  1.5328792   3.93497115  0.40795793 -4.58079812 -8.17323088  7.73905136
 -7.0176676   6.54425398  3.47086914  7.70339931 -7.45230157  1.13652698
 -7.77626075  9.01186896 -6.75420157 -4.96753175  7.48671793 -1.63983159
  0.71499775 -8.08202848 -8.8001341  -8.41104259 -7.16076924 -3.69088884]
[ 20.91526764 -22.05019551   4.50834388  -6.06976525  10.68215828
 -10.85936176  12.42932277  14.10364908  14.30580905 -28.82740604
 -16.41917499  24.89095978 -15.37943015  30.59921149  25.76199331
  26.02921016 -31.82165858   6.86111363  -3.8415529   -1.6291083
 -21.7103838   -8.40456302  -3.11259044  -5.55991551 -16.57567999
 -23.83968926  -8.93485805 -23.53908425 -16.1914137   -5.75080115]
```

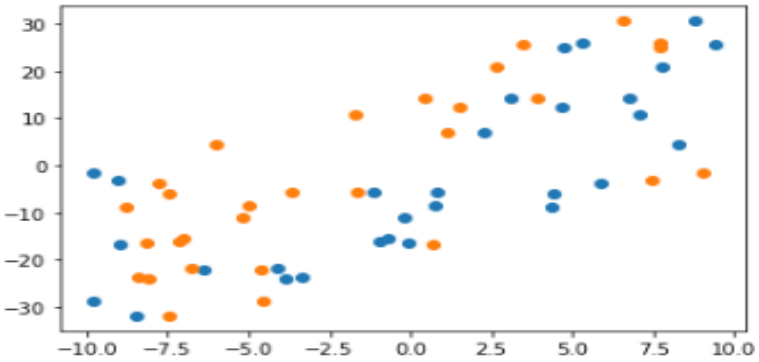

Figure 6 Task 3 (Output)

Name: Qasim Khan
CMSID: 49375

8. Rubrics:

Demonstration	Absent	Student is unable to follow the provided instructions properly. The students can name the hardware or simulation platform, but unable to implement anything or on the software.	Student can understand the provided laboratory instruction and familiar with the lab environment (Trainer/Software/IDE) , but cannot implement on the platform practically or on the software.	Student has followed instructions to construct the fundamental schematic/block diagram/ code/ model on the protoboard/ trainer/ simulation software.	Student has constructed the functional/ working schematic/ model/ block diagram/ code and have successfully executed the program/run circuit on software platform.	Student perfectly implemented a working model/logic/circuit /block diagram/code and successfully executed the lab objective in Realtime or in a simulation environment and produced the desired results.
Category	**Ungraded**	**Very Poor**	**Poor**	**Fair**	**Good**	**Excellent**
Percentage	[0]	[1-20]	[21-40]	[41-60]	[61-80]	[81-100]
Marks	0.0	0.01-0.20	0.21-0.40	0.41-0.60	0.61-0.80	0.81-1.0
Date		**Total Marks**		**Instructor's Signature**		

Laboratory reports	Report not submitted	Plagiarized content presented or incomplete submission	Requirements are listed and experimental procedure is presented	Observations are recoded along with detailed procedure	Appropriate computation or numerical analysis is performed	Correctly drawn conclusion with exact results and complete report in all aspects
Category	**Ungraded**	**Very Poor**	**Poor**	**Fair**	**Good**	**Excellent**
Percentage	[0]	[1-20]	[21-40]	[41-60]	[61-80]	[80-100]
Marks	0.0	0.01-0.20	0.21-0.40	0.41-0.60	0.61-0.80	0.81-1.0
Date		**Total Marks**		**Instructor's Signature**		