**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## Machine Learning Using Python

## LAB-1

# **Task : Understand Graph Plotting using Matplotlib in Python**

**Sol :** **Matplotlib** is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is used for plotting various plots in Python like scatter plot, bar charts, pie charts, line plots, histograms, 3-D plots and many more. We will learn about the scatter plot from the [matplotlib](https://www.geeksforgeeks.org/python-matplotlib-an-overview/) library.

1 )**subplot()** function adds subplot to a current figure at the specified grid position.  It is similar to the subplots() function however unlike subplots() it adds one subplot at a time. So to create multiple plots you will need several lines of code with the subplot() function.

Syntax

**subplot(nrows, ncols, index, \*\*kwargs)**

**subplot(pos, \*\*kwargs)**

**subplot(ax)**

**Parameters :**

* **args:** Either a 3-digit integer or three separate integers describing the position of the subplot.
* **pos** is a three-digit integer where the first, second, and third integer are nrows,ncols, index.
* **projection :** [{None, ’aitoﬀ’, ’hammer’, ’lambert’, ’mollweide’, ’polar’, ’rectilinear’, str}, optional]. The projection-type of the subplot (Axes). The default None results in a ’rectilinear’ projection.
* **label :** [str] A label for the returned axes.
* **\*\*kwargs:**This method also takes the keyword arguments for the returned axes base class;  
  except for the ﬁgure argument, for e.g facecolor.
* The ylim() function in pyplot module of matplotlib library is used to get or set the y-limits of the current axes. Parameters: This method accept the following parameters that are described below: bottom: This parameter is used to set the ylim to bottom. top: This parameter is used to set the ylim to top

Example :

**import matplotlib.pyplot as plt**

**import numpy as np**

**animal\_names = ['Lion', 'Deer', 'Cheetah']**

**mph\_speed = [50, 60, 75]**

**fig, ax = plt.subplots() //figure and subplot axes**

**bar\_container = ax.bar(animal\_names, mph\_speed)**

**ax.set(ylabel='speed in MPH', title='Running speeds', ylim=(0, 80))**

**ax.bar\_label**

**(**

**bar\_container, fmt=lambda x: '{:.1f} km/h'.format(x \* 1.61)**

**)**

**Output :**

**[Text(0, 0, '80.5 km/h'), Text(0, 0, '96.6 km/h'), Text(0, 0, '120.8 km/h')]**

****

## Example 2 matplotlib.pyplot.scatter()

Scatter plots are used to observe relationship between variables and uses dots to represent the relationship between them. The **scatter()** method in the matplotlib library is used to draw a scatter plot. Scatter plots are widely used to represent relation among variables and how change in one affects the other.   
**Syntax**   
The syntax for scatter() method is given below:

**matplotlib.pyplot.scatter(x\_axis\_data, y\_axis\_data, s=None, c=None, marker=None, cmap=None, vmin=None, vmax=None, alpha=None, linewidths=None, edgecolors=None)**

The scatter() method takes in the following parameters:

* **x\_axis\_data-** An array containing x-axis data
* **y\_axis\_data-** An array containing y-axis data
* **s-** marker size (can be scalar or array of size equal to size of x or y)
* **c-** color of sequence of colors for markers
* marker- marker style
* **cmap-** cmap name
* **linewidths-** width of marker border
* **edgecolor-** marker border color
* **alpha-** blending value, between 0 (transparent) and 1 (opaque)

Except x\_axis\_data and y\_axis\_data all other parameters are optional and their default value is None. Below are the scatter plot examples with various parameters.

**Example :**

**import matplotlib.pyplot as plt**

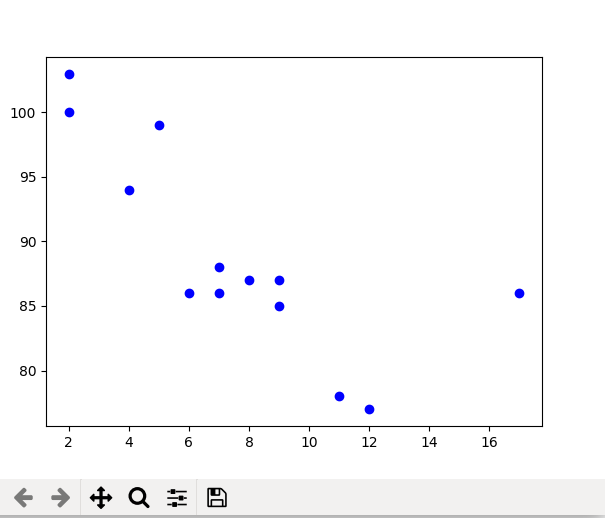
**x =[5, 7, 8, 7, 2, 17, 2, 9,4, 11, 12, 9, 6]**

**y =[99, 86, 87, 88, 100, 86,103, 87, 94, 78, 77, 85, 86]**

**plt.scatter(x, y, c ="blue")**

**# To show the plot**

**plt.show()**



**Example 2: Scatter plot with different shape and colour for two datasets.**

**import matplotlib.pyplot as plt**

**# dataset-1**

**x1 = [89, 43, 36, 36, 95, 10,66, 34, 38, 20]**

**y1 = [21, 46, 3, 35, 67, 95,53, 72, 58, 10]**

**# dataset2**

**x2 = [26, 29, 48, 64, 6, 5,36, 66, 72, 40]**

**y2 = [26, 34, 90, 33, 38,20, 56, 2, 47, 15]**

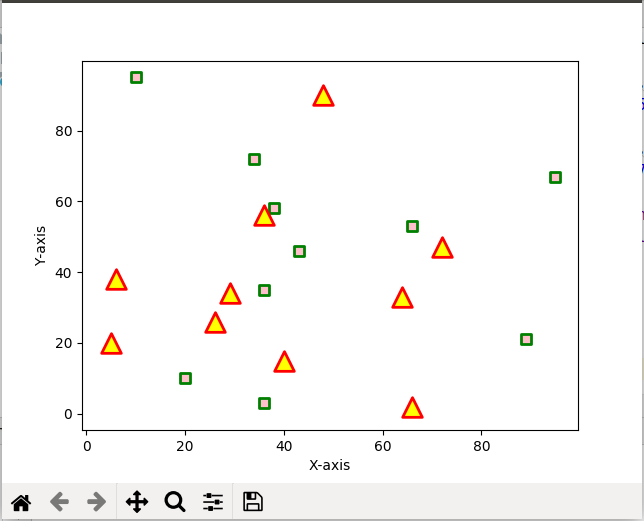
**plt.scatter(x1, y1, c ="pink",linewidths = 2,marker ="s",edgecolor ="green",s = 50)**

**plt.scatter(x2, y2, c ="yellow",linewidths = 2,marker ="^",edgecolor ="red",s = 200)**

**plt.xlabel("X-axis")**

**plt.ylabel("Y-axis")**

**plt.show()**



**Example 3:**

**import matplotlib.pyplot as plt**

**import numpy as np**

**Country = ('USA', 'China', 'India')**

**Gender\_counts =**

**{'Male': np.array([73, 34, 61]),'Female': np.array([73, 34, 58]),**

**}**

**width = 0.6 # the width of the bars: can also be len(x) sequence**

**fig, ax = plt.subplots()**

**bottom = np.zeros(3)**

**for Gender, gender\_count in Gender\_counts.items():**

**p = ax.bar(Country, gender\_count, width, label=Gender, bottom=bottom)**

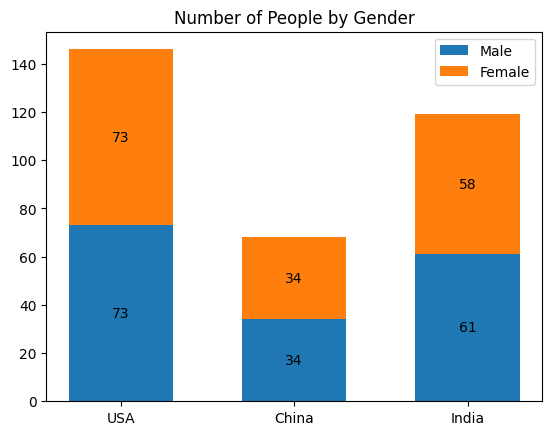
**bottom += gender\_count**

**ax.bar\_label(p, label\_type='center')**

**ax.set\_title('Number of People by Gender')**

**ax.legend()**

**plt.show()**

****

# **Example : Plot CSV data using Matplotlib and Pandas in Python**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**plt.rcParams["figure.figsize"] = [7.50, 3.50] // Canvas Area**

**plt.rcParams["figure.autolayout"] = True**

**headers = ['Name', 'Age', 'Marks']**

**df = pd.read\_csv('student.csv', names=headers)**

**df.set\_index('Name').plot()**

**plt.show()**

