[Octobit8 Assignment] [2021]

[5/23/2021]

[OCTOBIT8]

Authored by: [Rahul Kharatmol]

1.Write a Python Program to compute the multiplication of two matrices and then print it.

Matrics multiplication

In python matrx multiplication we used for loops hence we can multiply with rows and column.

Our matrix is 3 * 3

Each row is multiply by each column

The logic behind is that

```
result = [[sum(a * b for a, b in zip(A_row, B_col))
           for B_col in zip(*B)]
          for A_row in A]
```

Input

```
J = [[1, 17, 43],
     [44, 55, 61],
     [74, 85, 93]]

K = [[5, 8, 1, 2],
     [6, 7, 3, 0],

     [4, 5, 9, 1]]

result = [[sum(a * b for a, b in zip(A_row, B_col))
           for B_col in zip(*B)]
          for A_row in A]

for r in result:
    print(r)
```

Output

```
[431, 610, 796]
[536, 742, 964]
[890, 1108, 1314]
```

2. Write a Python program to find yesterday, today and tomorrow date.

```
import datetime
today = datetime.date.today()
yesterday = today - datetime.timedelta(days = 1)
tomorrow = today + datetime.timedelta(days = 1)
print('Yesterday date : ', yesterday)
print('Today date : ', today)
print('Tomorrow date  : ', tomorrow)
```

Output

```
Yesterday date :  2021-05-22
Today date :  2021-05-23
Tomorrow date  :  2021-05-24
```

3. Write a Python program to find Nth largest number in a list.

In this program we used logic is that for loop and check the in the list which is maximum elements using the logic is greater than if the **maximum element is greatest than that number is printed as last.**

Logic of this program

```
    for j in range(len(list1)):
        if list1[j] > max1:
            max1 = list1[j];
```

Input

```
def Maximumelements(list1, N):
    List = []

    for i in range(0, N):
        max1 = 0

        for j in range(len(list1)):
            if list1[j] > max1:
                max1 = list1[j];

        list1.remove(max1);
        List.append(max1)

    print(List)


# Driver code
list1 = [2, 6, 41, 85, 0, 3, 7, 6, 10]
N = 1
```

Output

```
[85]
```

4.Explain NLTK module in python and execute each step involved. Document the complete process.

Nltk

Nltk full form is Natural language toolkit. Nltk is python library which is used in python for text Analysis.

Nltk process

- **Stop word Remove**
- **Tokenization**
- **Stemming**
- **Lemitization**
- **Pos tagging**

Stop word Remove

A stop word is a commonly used word (such as {'mightn', 'over', 'isn', 'down', 'themselves', "haven't", 'into', 'shan', 'from', 'before', 'how', 'these', 'being', 'where', 'whom', 'needn', 'll', "hasn't", 'some', 'itself', 'wasn', 'if', 'here', 'through', 'what', 'but', 'does', 'myself', 'the', 'above', 'further', 'haven', 've', 't', 'few', 'off', 'i', 'd', 'a', 'an', 'its', 'yourself', 'who', 'more', 'theirs', 'yourselves', 'such', 'just', 'no', 'about', "hadn't", 'been', 'once', 'so', "needn't", 'am', 'we', 'had', 'those', 'any', 'yours', 'which', "doesn't", 'nor', 'under', "mightn't", 'all', 'this', "you're", 'o', 'shouldn', 'them', "isn't", 'will', 'can', "didn't", 'he', "you'd", 'mustn', 'won', 'aren', 'their', 'until', 'at', 'himself', 'below', 'hasn', 'why', 'there', 'do', 'did', 'when', 'most', 'than', 'by', 'weren', 'with', "shouldn't", "weren't", 'on', 'his', 'was', 'in', "wasn't", 'doing', 'herself', 'didn', 'or', 'ours', 'having', "wouldn't", 'of', "that'll", 'is', 'it',

'while', 'between', 'each', 'and', 'hers', 'don', 'wouldn', 'against', "it's", 'm', 'ain', "she's", 'couldn', 'out', 'ourselves', 'as', 'be', 'our', 'my', "don't", 'hadn', 'only', 'him', "couldn't", "you'll", 'her', 'have', 'not', 'are', 'she', 'after', 'then', 's', "mustn't", 'for', 'own', 're', 'same', 'ma', "should've", "shan't", 'during', 'y', 'up', 'your', 'doesn', 'were', 'they', 'both', 'should', 'that', 'you', 'me', 'very', "you've", 'has', 'to', 'because', 'other', 'now', "won't", 'again', "aren't", 'too'} that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.

```
from nltk.corpus import stopwords

stop_words = set(stopwords.words("english"))
print(stop_words)
```

Using this stop words can remove the stop words which are present in sentence.

```
text = """people are good in india. They are verry helpfull"""
```

This is our sentence in that are and in are stop word that stop word removed by this import stopwords.

```
['people', 'good', 'india', 'They', 'verry', 'helpfull']
```

Tokenization

In tokenization that text are converted into small chunks. Hence we can get tokenized words for further text analysis. In this process each word get tokenized.

In our project there is small chunks are like

Stemming

Stemming is nothing but removed the prefix and postfix word according that we will get the words.for example

Caring =car

This is remove ing but the meaning forthis word is totally change

In our project

Code

```
ps = nltk.PorterStemmer()
w = [ps.stem(word) for word in words_new]
print(w)
```

Output

```
['peopl', 'good', 'india', 'they', 'verri', 'helpful']
```

In our project the people of e is removed they will get meaning less
Also verry word is converted into verri

lemitization

in lemmatization it will get meaningful words hence we will get meaningful word this is verry useful for text analysis.

In our project

Code

```
wn = nltk.WordNetLemmatizer()
w = [wn.lemmatize(word) for word in words_new]
print(w)
```

Output

```
['people', 'good', 'india', 'They', 'verry', 'helpfull']
```

Pos tag
In pos tagging we will get noun, adverb and Adjective using for particular Text.
Pos tagging

Code

```
# pos tagging
tagged = nltk.pos_tag(w)

print(tagged)
```

Output

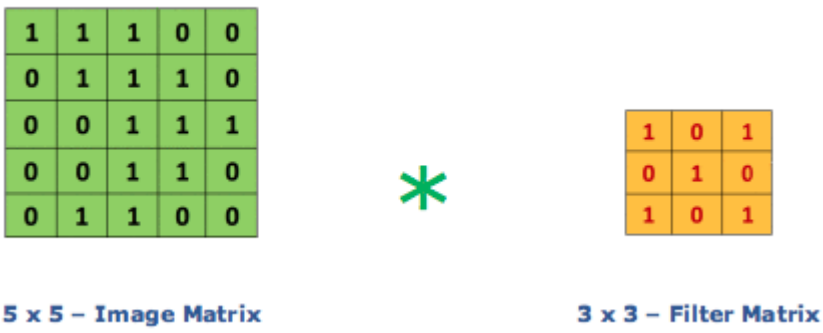
```
[('people', 'NNS'), ('good', 'JJ'), ('india', 'NN'), ('They', 'PRP'), ('verry', 'VBP'), ('helpfull', 'VB')]
```

5. What is Convolution Neural Network? Illustrate using python and document the process.

What is cnn in neural network.

Cnn is mostly used in image precession. For detecting the object . in cnn technique used feature extraction technique for that extracft features them the data and convolute them and using function get better output.

In the feature map we used 3 * 3 multiplication of two matrix.



With the convolution used the activation function like Relu .

Relu Activation function is negative values are converted into positive values for linearity of matrix.

In our code is handwritten recognition the recognized what are the handwritten words are present with using CNN algo.

```
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Flatten
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.utils import np_utils
# load data
(X_train, y_train), (X_test, y_test) = mnist.load_data()
# reshape to be [samples][width][height][channels]
X_train = X_train.reshape((X_train.shape[0], 28, 28, 1)).astype('float32')
X_test = X_test.reshape((X_test.shape[0], 28, 28, 1)).astype('float32')
# normalize inputs from 0-255 to 0-1
X_train = X_train / 255
X_test = X_test / 255
# one hot encode outputs
y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
num_classes = y_test.shape[1]
```

In above code this is handwritten recognition dataset .load from the google.and the store into x_train and Y_train. After that will be stored as numeric. categorical data is converted into numeric.

```
def baseline_model():
    # create model
    model = Sequential()
    model.add(Conv2D(32, (5, 5), input_shape=(28, 28, 1), activation='relu'))
    model.add(MaxPooling2D())
    model.add(Dropout(0.2))
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dense(num_classes, activation='softmax'))
    # Compile model
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
# build the model
model = baseline_model()
# Fit the model
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=10, batch_size=200)
# Final evaluation of the model
scores = model.evaluate(X_test, y_test, verbose=0)
print("CNN Error: %.2f%%" % (100-scores[1]*100))
```

Activation function is used for particular dataset is Relu.

Accuracy

```
Epoch 1/10
300/300 [==============================] - 65s 111ms/step - loss: 0.4877 - accuracy: 0.8617 - val_loss: 0.0758 - val_accuracy: 0.9762
Epoch 2/10
300/300 [==============================] - 31s 105ms/step - loss: 0.0786 - accuracy: 0.9771 - val_loss: 0.0530 - val_accuracy: 0.9823
Epoch 3/10
300/300 [==============================] - 32s 107ms/step - loss: 0.0551 - accuracy: 0.9834 - val_loss: 0.0474 - val_accuracy: 0.9852
Epoch 4/10
300/300 [==============================] - 33s 111ms/step - loss: 0.0423 - accuracy: 0.9872 - val_loss: 0.0413 - val_accuracy: 0.9871
Epoch 5/10
300/300 [==============================] - 106s 354ms/step - loss: 0.0294 - accuracy: 0.9908 - val_loss: 0.0343 - val_accuracy: 0.9887
Epoch 6/10
300/300 [==============================] - 31s 103ms/step - loss: 0.0271 - accuracy: 0.9910 - val_loss: 0.0356 - val_accuracy: 0.9885
Epoch 7/10
300/300 [==============================] - 32s 107ms/step - loss: 0.0211 - accuracy: 0.9936 - val_loss: 0.0312 - val_accuracy: 0.9896
Epoch 8/10
300/300 [==============================] - 34s 115ms/step - loss: 0.0163 - accuracy: 0.9947 - val_loss: 0.0389 - val_accuracy: 0.9882
Epoch 9/10
300/300 [==============================] - 32s 108ms/step - loss: 0.0166 - accuracy: 0.9943 - val_loss: 0.0306 - val_accuracy: 0.9903
Epoch 10/10
300/300 [==============================] - 32s 108ms/step - loss: 0.0113 - accuracy: 0.9967 - val_loss: 0.0339 - val_accuracy: 0.9905
CNN Error: 0.95%

Process finished with exit code 0
```

6. Implement NLP analysis of a restaurant review in python.

In Restaurant review analysis used the dataset is restaurant.tsv

In that first install nlk library for preprocessing the dataset .

That data converted into numbers after that into numeric values and that converted as process . algorithms are Random forest and xgboost.

Dataset

In restaurant dataset I collect from Kaggle that is restaurant.tsv

Restaurent code

Input

```
import pandas as pd

# Import dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter='\t')
dataset
import re

# Natural Language Tool Kit
import nltk

nltk.download('stopwords')

# to remove stopword
from nltk.corpus import stopwords

# for Stemming propose
from nltk.stem.porter import PorterStemmer

# Initialize empty array
# to append clean text
corpus = []

# 1000 (reviews) rows to clean
for i in range(0, 1000):
    # column : "Review", row ith
```

```
model = RandomForestClassifier(n_estimators=501,
                               criterion='entropy')

model.fit(X_train, y_train)

# Predicting the Test set results

y_prediction = model.predict(X_test)
print(y_prediction)
from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_test, y_prediction)
print(accuracy)
from xgboost import XGBClassifier


# fit model no training data
model = XGBClassifier()
model.fit(X_train, y_train)
y_prediction1 = model.predict(X_test)
print(y_prediction1)
accuracy1 = accuracy_score(y_test, y_prediction1)
print(accuracy1)
```

Output

```
 1 0 1 0 0 1 1 0 0 1 1 1 1 0 1 1 0 1 0 0 1 1 0 0 0 1 0 0 0 1 1 0 0 1 0 1
 0 1 0 1 0 0 0 0 1 0 0 0 0 1 0 1 1 0 0 0 1 0 1 0 1 1 0 0 0 1 0 0 0 0 1 0
 0 1 1 1 0 1 0 0 1 1 1 1 0 0 1 1 0 0 0 0 0 1 0 0 0 0 1 1 1 0 1 0 1 0 0 1
 0 0 1 0 1 0 1 1 1 0 0 1 1 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0
 1 1 1 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 1 1 1 0 0 1 1 0 0]
0.736
```

.

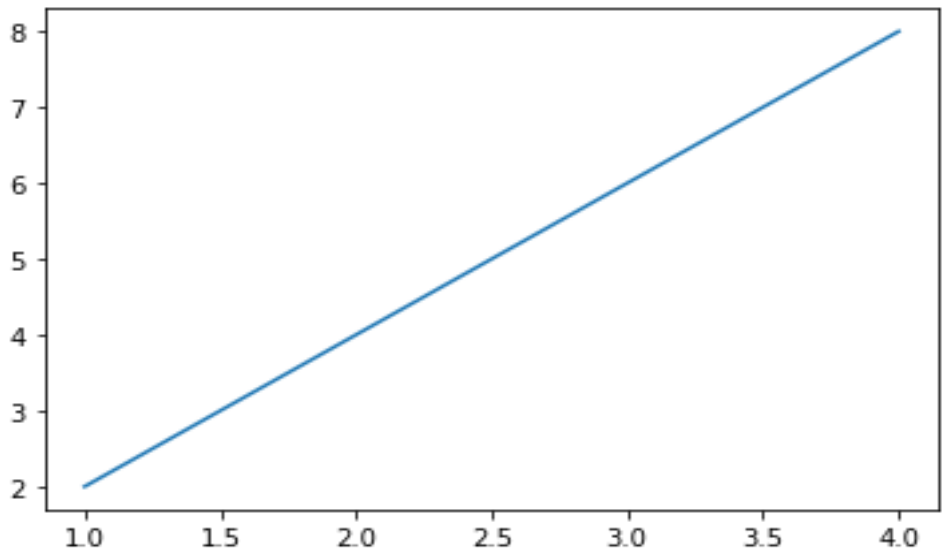
7. Create Line Graph, Bar chart, Histograms, Scatter plot, Pie Chart, 3D plots using matplotlib in python.

Line graph

Input

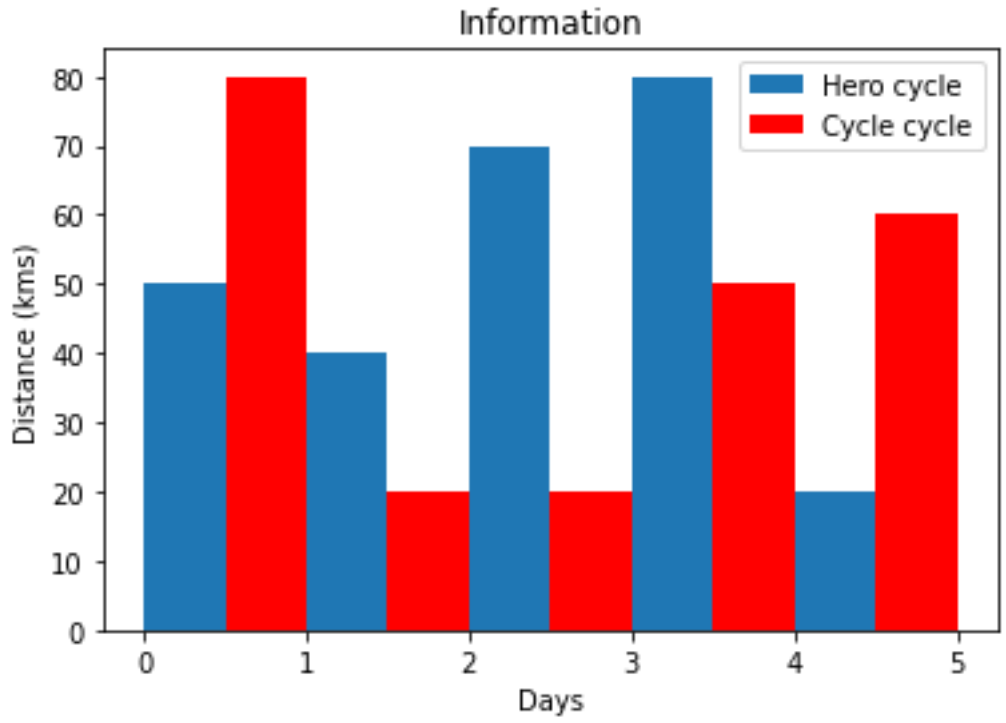
```
import numpy as np

# define data values
x = np.array([5, 6, 7, 8])# X-axis points
y = x*2# Y-axis points

plt.plot(x, y)# Plot the chart
plt.show()# display
```

Output



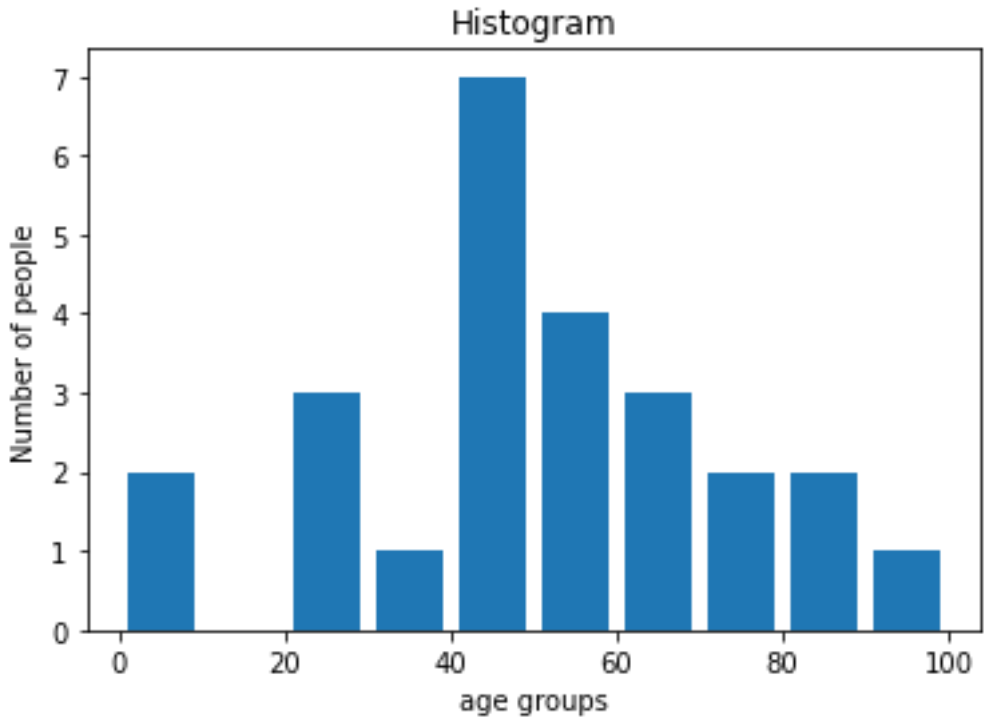
Bar chart

```
from matplotlib import pyplot as plt

plt.bar([0.25, 1.25, 2.25, 3.25, 4.25], [50, 40, 70, 80, 20],
        label="Hero cycle", width=.5)
plt.bar([.75, 1.75, 2.75, 3.75, 4.75], [80, 20, 20, 50, 60],
        label="Cycle cycle", color='r', width=.5)
plt.legend()
plt.xlabel('Days')
plt.ylabel('Distance (kms)')
plt.title('Information')
plt.show()
```



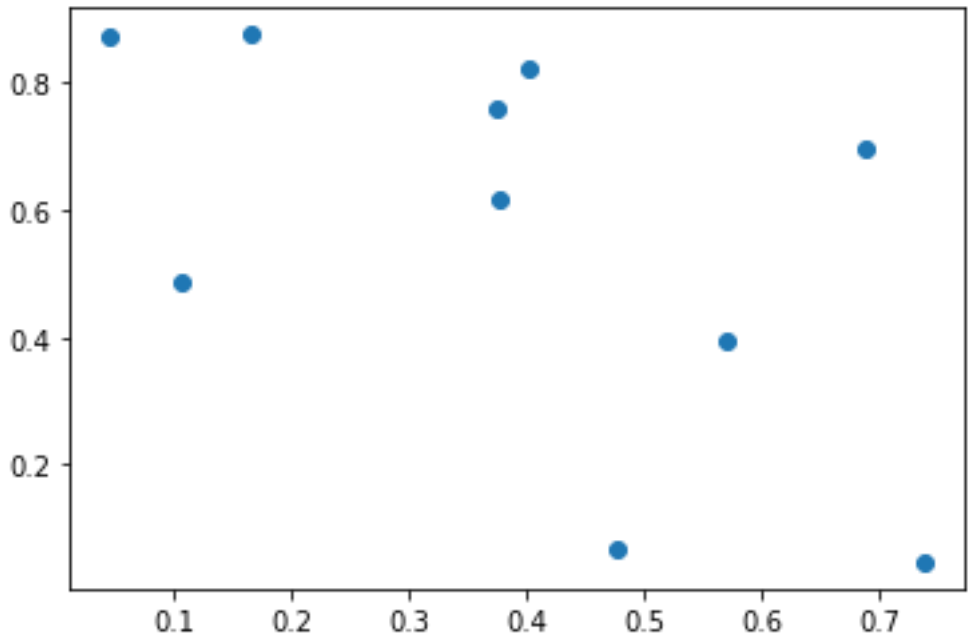
Histogram

```
import matplotlib.pyplot as plt
population_age = [22,55,62,45,21,22,34,42,42,4,2,102,95,85,55,110,120,70,65,55,111,115,80,75,65,54,44,43,42,4
bins = [0,10,20,30,40,50,60,70,80,90,100]
plt.hist(population_age, bins, histtype='bar', rwidth=0.8)

plt.xlabel('age groups')
plt.ylabel('Number of people')

plt.title('Histogram')
plt.show()
```



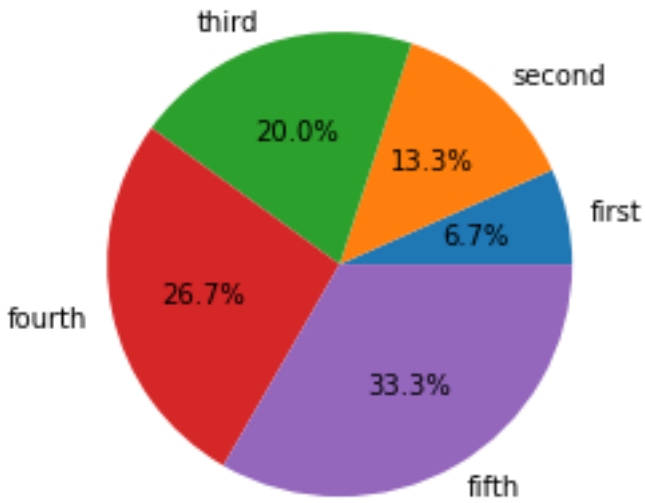
Scatter plot

```
import matplotlib.pyplot as plt
import numpy as np

n = 10
x = np.random.rand(n)
y = np.random.rand(n)

plt.scatter(x, y)
plt.show()
```



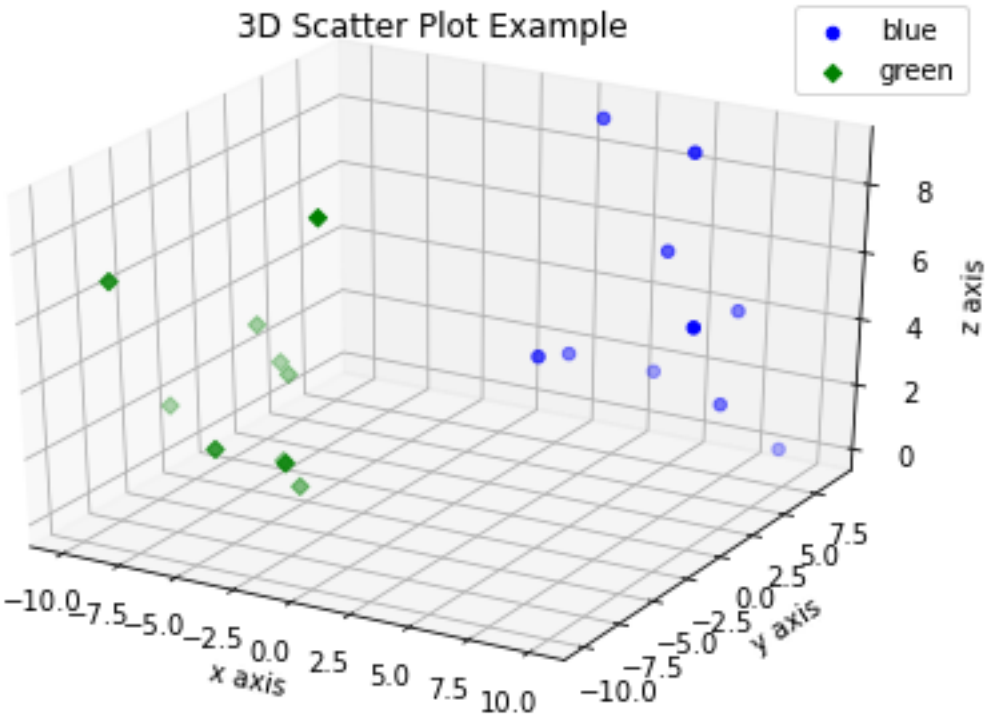
Piechart

```
import matplotlib.pyplot as plt
import numpy as np

x = [100, 200, 300, 400, 500]
labels = ['first', 'second', 'third', 'fourth', 'fifth']

plt.pie(x, labels=labels, autopct='%1.1f%%')
```



3 d plot

```
import numpy as np
from mpl_toolkits.mplot3d import axes3d

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

x1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
y1 = np.random.randint(10, size=10)
z1 = np.random.randint(10, size=10)

x2 = [-1, -2, -3, -4, -5, -6, -7, -8, -9, -10]
y2 = np.random.randint(-10, 0, size=10)
z2 = np.random.randint(10, size=10)

ax.scatter(x1, y1, z1, c='b', marker='o', label='blue')
ax.scatter(x2, y2, z2, c='g', marker='D', label='green')

ax.set_xlabel('x axis')
ax.set_ylabel('y axis')
ax.set_zlabel('z axis')
plt.title("3D Scatter Plot Example")
plt.legend()
plt.tight_layout()
plt.show()
```


3D Scatter Plot Example
blue
green
z axis
x axis
y axis

8 .Implement a class in python and explain all the properties of a class using OOPs in python

Class

Class is s nothing but blueprint of object.

Like one example if object is xyz then class also xyz.

Class xyz
  Xyz

Object

Object is nothing but instance of class

Obj = xyz()

Creating Class and Object in Python

```
class language:
    def __init__(self, language1):
        self.language1 = language1


    def display(self):
        print(self.language1)


c1 = language("java")
c1.display()
```

In above example object is language and class is also language .
__init__ is initializer method to used initialization of class.
create instances of the language class. Here java are locations (value) to our new objects.

Output

```
java
```

Creating Methods in Python

```
# Creating Methods in Python
class language:

    # instance attributes
    def __init__(self, language1, feature):
        self.language1 = language1
        self.feature = feature

    # instance method
    def language(self, language1):
        return "language {} ".format(self.language1)

# instantiate the object
blu = language("java", "dependent")

# call our instance methods
print(blu.language("''"))
```

In that one methods are called that’s is languages .

Output

```
language java
```

Inheritance

```
class Parents:

    def __init__(self):
        print("Parenet is ready")

    def whoisThis(self):
        print("Parents")

    def swim(self):
        print("Swim faster")
# child class
class child(Parents):

    def __init__(self):
        # call super() function
        super().__init__()
        print("child is ready")

    def whoisThis(self):
        print("child")
peggy = Parents()
peggy.whoisThis()
peggy.swim()
```

In this inheritance one is parent calss and one is child class.in inheritance in the child class can call parent class but in parent class we cant call child class.

```
Parenet is ready
Parents
Swim faster
```

In above example parent class and child class also called.

Polymorphism

```
class Crow:
     def fly(self):
        print("Crow can fly")

    def swim(self):
        print("Crow can't swim")
class dolphin:

    def fly(self):
        print("dolphin can't fly")

    def swim(self):
        print("dolphin can swim")
# common interfacedef
# instantiate objects

def flying_test(bird):
    bird.fly()

#instantiate objects
n1 = Crow()
n2 = dolphin()

# passing the object
flying_test(n1)
```

In polymorphism method are same but function are different in above example methos is fly but function of particular crow and dolphin are different.

Output

```
Crow can fly
dolphin can't fly
```

Encapsulation

In encapsulation the prevents the direct modification in python.

```
class Computer:

    def __init__(self):
        self.__maxprice = 900

    def sell(self):
        print("Selling Price: {}".format(self.__maxprice))

    def setMaxPrice(self, price):
        self.__maxprice = price

c = Computer()
c.sell()

# change the price
c.__maxprice = 1200
c.sell()

# using setter function
c.setMaxPrice(1000)
c.sell()
```