1.

l1=[1,2,3,4,5]

l2=[1,2,3,8]

l=len(l2)

count=0

for i in l1:

for j in l2:

if i==j:

count+=1

if count==l:

print("Trueee")

else:

print("Falseeeeee")

--------------------------------------

2.

marks=[{

'name':"daya","m1":33,'m2':99

},{"name":"prasi","m1":99,'m2':88}]

for mark in marks:

m1=mark.pop('m1');

m2=mark.pop('m2')

mark['avg']=(m1+m2)/2

print(marks)

----------------------------------------

3.

a=(1,2,3,3,5)

print(a[:2])

b=tuple("Dhayaaaaaaaaaaa")

print(b[:])

------------------------------------------------

4.

import numpy as np

mat=np.array([[1,2,3],[1,2,3]])

mat1=np.array([[1,2,3],[1,2,3]])

print(np.multiply(mat,mat1))

print(np.divide(mat,mat1))

------------------------------------------------------

5.

import pandas as pd

m={"values":[1,2,3,4,5,6,76,78,89,90]}

df=pd.DataFrame(m)

df['sq']=df["values"]\*\*2

print(df)

--------------------------------------------------------

6.

text=input().lower()

print(text)

l=list(set(text))

if " " in l:

l.remove(" ")

if len(l)==26:

print("anangaram")

else:

print("not an anagram")

print(l)

-----------------------------------------------------------

7.

from numpy import array

from sklearn.model\_selection import KFold

data = array([0.1, 0.2, 0.3, 0.4, 0.5, 0.6,0.7])

kfold = KFold(3)

for train, test in kfold.split(data):

print('train: %s, test: %s' % (data[train], data[test]))

-----------------------------------------------------------

8a.

import csv

col\_names=['Name','Age'];

data=[["Ammaaa",'0'],["Dhayaa",'100']]

file="rec.csv"

with open(file,'w') as csvfile:

csvwriter=csv.writer(csvfile);

csvwriter.writerow(col\_names)

csvwriter.writerows(data)

8b.

import csv

with open('rec.csv','r') as csvfile:

rows=csv.reader(csvfile)

for row in rows:

print(row)

-----------------------------------------------------------

9.

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

import sys

x=[4,5,10,4,3,22,2,84]

y=[21,19,24,16,25,24,22,21]

plt.scatter(x,y)

plt.show()

d=list(zip(x,y))

kmeans = KMeans(n\_clusters=2)

kmeans.fit(d)

plt.scatter(x,y,c=kmeans.labels\_)

plt.show()

plt.savefig(sys.stdout.buffer)

sys.stdout.flush()

-----------------------------------------------------------

10.

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import accuracy\_score

df = pd.read\_csv("dataset.csv")

x = df.drop("diabetes",axis=1)

y = df["diabetes"]

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.25, random\_state=42)

model = GaussianNB()

model.fit(x\_train,y\_train)

y\_pred=model.predict(x\_test)

accuracy=accuracy\_score(y\_test,y\_pred)\*100

print(accuracy)

OUTPUT:

Accuracy: 92.7710843373494

DATASET:

https://github.com/dhirajk100/Naive-Bayes

-----------------------------------------------------------

11.

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.linear\_model import LogisticRegression

data = pd.read\_csv('irisdata.csv')

x = data[['length', 'width']]

y = data['Type']

model = LogisticRegression(solver='liblinear', random\_state=0)

model.fit(x,y)

z=model.predict(x)

print(z)

plt.plot(z)

plt.show()

-----------------------------------------------------------

12.

import sklearn.datasets as sd

import sklearn.model\_selection as al

import sklearn.ensemble as se

import sklearn.linear\_model as sl

import sklearn.svm as sv

import time

X, Y = sd.load\_diabetes(return\_X\_y=True)

X\_train, X\_test, Y\_train, Y\_test = al.train\_test\_split(X,Y,random\_state=42)

stacked = se.StackingRegressor( estimators =[('SVR', sv.SVR()),('Liner',sl.LinearRegression())])

st = time.time()

stacked.fit(X\_train, Y\_train)

et = time.time()

print("Coefficient of determination: {}".format(stacked.score(X\_test, Y\_test)))

print("Computation Time: {}".format(et - st))

-------------------------------------------------------------------------------

13.

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

from sklearn import svm

data = pd.read\_csv('irisdata.csv')

x = data['length']

y = data['width']

tr\_x = np.vstack((x, y)).T

tr\_y = data['Type']

clf = svm.SVC(kernel='linear', C=1.0)

clf.fit(tr\_x,tr\_y)

w =clf.coef\_[0]

a = -w[0] /w[1]

xx = np.linspace(0, 13)

yy = a\* xx -clf.intercept\_[0] / w[1]

plt.plot(xx,yy,'k-')

plt.scatter(tr\_x[:,0],tr\_x[:,1],c=tr\_y)

plt.show()

---------------------------------------------------------------------------------------