**….…………PROGRAMS…..…………**

**PROGRAM 1 :**

x = int(input(“Enter 1st Number : “))

y = int(input(“Enter 2nd Number : “))

z = x +y

print(“Sum of two number is :”,z)

**PROGRAM 2 :**

x = int(input(“Enter a number :”))

r =x%2

if r ==0:

print(“Even”)

elif r==1:

print(“Odd”)

**PROGRAM 3 :**

x = int(input(“Enter a number:”))

r = x%2

if r == 0:

print(“Even”)

else:

print(“Odd”)

**PROGRAM 4 :**

**Generate output**

**Abhi Rocks Rocks Rocks Rocks**

**Abhi Rocks Rocks Rocks Rocks**

**Abhi Rocks Rocks Rocks Rocks**

**Abhi Rocks Rocks Rocks Rocks**

i = 1

while i<=5:

print(‘Abhi’,end = ‘ ‘)

j = 1

while j<=4:

print(‘Rocks’,end = ‘ ‘)

j = j+1

i = i +1

print()

**PROGRAM 5 :**

x = ‘ABHISHANK’

for i in x:

print(i)

**PROGRAM 6 :**

x = { ‘navin’ , 65 ,2.5 }

for i in x:

print(i)

**PROGRAM 7 :**

**Generate Output**

**#**

**# #**

**# # #**

**# # # #**

for i in range(4):

for j in range(i+1):

print(“#”,end= ‘ ‘)

print()

**PROGRAM 8 :**

**Generate Output**

**# # # #**

**# # #**

**# #**

**#**

for i in range(4):

for j in range(4-i):

print(“#”,end = ‘ ‘)

print()

**PROGRAM 9 :**

***MAKE BOXPLOT***

import pandas as pd

import matplotlib.pyplot as plt

import os

os.getcwd()

# get current working directory

dataset = pd.read\_csv(‘exms.csv’)

dataset

dataset.columns

dataset.boxplot(‘science’)

plt.show()

**PROGRAM 10 :**

***MAKE HISTOGRAM***

import pandas as pd

import matplotlib.pyplot as plt

import os

os.getcwd()

// get current working directory

dataset = pd.read\_csv(‘exms.csv’)

dataset

dataset.columns

dataset.boxplot(‘science’)

plt.hist(dataset)

plt.show()

**PROGRAM 11 :**

***PRINT MEAN OF A PARTICULAR COLUMN BY READING A CSV FILE***

import pandas as pd

import matplotlib.pyplot as plt

import os

os.getcwd()

#get current working directory

dataa = pd.read\_csv(‘exms.csv’)

dataa

# Find the name of column whose mean is to be found

print(dataa[‘Earning’].mean())

**PROGRAM 12 :**

***PRINT MEDIAN OF A PARTICULAR COLUMN BY READING A CSV FILE***

import pandas as pd

import matplotlib.pyplot as plt

import os

os.getcwd()

#get current working directory

dataa = pd.read\_csv(‘exms.csv’)

dataa

#Find the name of column whose median is to be found

print(dataa[‘Earning’].median())

**PROGRAM 13 :**

***PRINT MODE OF A PARTICULAR COLUMN BY READING A CSV FILE***

import pandas as pd

import matplotlib.pyplot as plt

import os

os.getcwd()

// get current working directory

dataa = pd.read\_csv(‘exms.csv’)

dataa

// Find the name of column whose mode is to be found

print(dataa[‘Earning’].mode())

**PROGRAM 14 :**

***PRINT STANDARD DEVIATION OF A PARTICULAR COLUMN BY READING A CSV FILE***

import pandas as pd

import matplotlib.pyplot as plt

import os

os.getcwd()

// get current working directory

dataa = pd.read\_csv(‘exms.csv’)

dataa

// Find the name of column whose standard deviation is to be found

print(dataa[‘Earning’].std())

**PROGRAM 15 :**

***PRINT VARIANCE OF A PARTICULAR COLUMN BY READING A CSV FILE***

import pandas as pd

import matplotlib.pyplot as plt

import os

os.getcwd()

#get current working directory

dataa = pd.read\_csv(‘exms.csv’)

dataa

#Find the name of column whose variance is to be found

print(dataa[‘Earning’].var())

**PROGRAM 16 :**

***PRINT THIS PATTERN***

# # # #

# # # #

# # # #

# # # #

for i in range(4):

for j in range(4):

print(“#”, end = “ “)

print()

**PROGRAM 17 :**

x = 8

r = x%2

if r == 0:

print(“Even”)

if x>5:

print(“Greater than five”)

else:

print(“Not Greater than five”)

else:

print(“Odd”)

**PROGRAM 18 :**

***PRINT THIS PATTERN***

nums = { 12 , 16 , 18 , 21 , 26 }

for num in nums :

if num%5 == 0:

print(num)

break

else:

print(“ not found”)

**PROGRAM 19 :**

***MAKE SCATTER PLOT***

import pandas as pd

import matplotlib.pyplot as plt

datta = pd.read\_csv(‘exms.csv’)

datta

plt.scatter(datta[‘distance travel’],datta[‘Earning’])

plt.show()

**PROGRAM 20 :**

***COPYING AN ARRAY***

import numpy

from numpy import \*

arr = array([1, 2,3,4,5])

arr = arr + 5

print(arr)

**PROGRAM 21 :**

***VECTORIZED OPERATION ON ARRAY***

import numpy

from numpy import \*

arr1 = array([1, 2,3,4,5])

arr2 = array([6,7,8,9,10])

arr3 = arr1 + arr2

print(arr3)

**PROGRAM 22 :**

***CONCATENATE two arrays***

from numpy import \*

arr1 = array([1,2,3,4,5])

arr2 = array([6,7,8,9,10])

arr3 = concatenate([arr1,arr2])

arr3

**PROGRAM 23 :**

***ASSIGNING One array to other***

***(same memory address)***

from numpy import \*

arr1 = array([1,2,3,4,5])

arr2 = arr1

id(arr1)

id(arr2)

**PROGRAM 24 :**

***ASSIGNING One array to other***

***(different memory address)***

from numpy import \*

arr1 = array([1,2,3,4,5])

arr2 = arr1.view()

id(arr1)

id(arr2)

**PROGRAM 25 :**

***DEFINE factorial of a number***

def fact(n):

f = 1

for i in range(1,n+1):

f = f\*i

return f

x = int(input(“Enter the number whose factorial to be found”)

result = fact(x)

print(“ Factorial is” , result )

**PROGRAM 26 :**

***Program on Anonymous Function***

f = lambda a : a\*a

result = f(5)

result

**PROGRAM 27 :**

***Program on Logistic Regression***

Data set as follows :

|  |  |
| --- | --- |
| Age | Loan |
| 35 | 1 |
| 47 | 0 |
| 55 | 0 |
| 52 | 0 |
| 32 | 1 |
| 29 | 1 |
| 38 | 0 |
| 36 | 1 |
| 44 | 0 |
| 33 | 1 |
| 39 | 0 |
| 31 | 1 |
| 36 | 1 |
| 37 | 1 |
| 48 | 0 |
| 52 | 0 |
| 59 | 0 |
| 56 | 0 |
| 53 | 0 |
| 54 | 0 |
| 51 | 0 |
| 37 | 1 |
| 28 | 1 |

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import os

os.getcwd()

dataset = pd.read\_csv(“exms.csv”)

plt.scatter(dataset [‘age’], dataset [‘loan’])

plt.show()

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

x\_train , x\_test , y\_train , y\_test = train\_test\_split (dataset [“age”], dataset [“loan”],train\_size = 0.8 , random\_state = 10)

from sklearn.linear\_model import LogisticRegression

model = LogisticRegression()

x\_train = x\_train.values.reshape(-1,1)

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

x\_test = x\_test.values.reshape(-1,1)

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

y\_predicted

model.score(x\_test,y\_test)

**PROGRAM 28 :**

***Program on Linear Regression (Single Variable)***

Data set as follows :

|  |  |
| --- | --- |
| Distance travel | Price |
| 5000 | 80000 |
| 10000 | 75000 |
| 15000 | 70000 |
| 20000 | 65000 |
| 25000 | 60000 |
| 30000 | 55000 |
| 35000 | 50000 |
| 40000 | 45000 |
| 45000 | 40000 |
| 50000 | 35000 |
| 55000 | 30000 |
| 60000 | 25000 |
| 65000 | 20000 |
| 70000 | 15000 |
| 75000 | 12000 |
| 80000 | 10000 |
| 85000 | 9000 |
| 90000 | 8000 |
| 95000 | 7000 |
| 100000 | 6000 |
| 105000 | 5000 |
| 110000 | 4000 |
| 11500 | 3500 |

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

df = pd.read\_csv(“abc.csv”)

df.head(10)

plt.scatter(df[‘Distance travel’],df[‘Price’])

plt.show()

from sklearn.linear\_model import LinearRegression

x = df[“distance travel”]

y = df[“price”]

x = x.values.reshape(-1,1)

reg = LinearRegression()

reg.fit(x,y)

reg.predict([25])

reg.coef\_

reg.intercept\_

**PROGRAM 29 :**

***Program on Linear Regression (Multiple Variable)***

(Bike price prediction example)

|  |  |  |  |
| --- | --- | --- | --- |
| s.no | Distance | Year | price |
| 0 | 500 | 5 | 5000 |
| 1 | 600 | 3 | 4500 |
| 2 | 100 | 1 | 7000 |
| 3 | 200 | 2 | 6000 |
| 4 | 400 | 7 | 2500 |
| 5 | 800 | 9 | 2000 |

Import pandas as pd

df = pd.read\_csv(“abcd.csv”)

df.head()

x = df[[‘distance’,’year’]]

y = df[‘price’]

x

y

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

x\_train , x\_test , y\_train , y\_test =

train\_test\_split(x,y,test\_size = 0.3,random\_state = 10)

from sklearn.linear\_model import LinearRegrssion

clf = LinearRegression()

clf

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

clf.predict(x\_test)

clf.score(x\_test , y\_test)

clf.coef\_

clf.intercept\_

clf.predict([[350,4]])

**PROGRAM 30 :**

***K Means Clustering Algorithm***

from sklearn.cluster import KMeans

import pandas as pd

from sklearn.preprocessing import MinMaxScaler

from matplotlib import pyplot as plt

df = pd.read\_csv(“Book1.csv”)

df.head()

|  |  |  |  |
| --- | --- | --- | --- |
| S.No. | Name | Rollno | Marks |
| 1 | A | 40 | 66 |
| 2 | B | 41 | 53 |
| 3 | C | 43 | 76 |
| 4 | D | 39 | 79 |
| 5 | E | 36 | 72 |
| 6 | F | 38 | 88 |
| 7 | G | 44 | 62 |
| 8 | H | 42 | 64 |
| 9 | I | 33 | 80 |
| 10 | J | 37 | 156 |

pl.scatter(df.rollno,df[‘marks’])

plt.xlabel(‘rollno’)

plt.ylabel(‘marks’)

km = KMeans(n\_cluster = 3)

predicted = km.fit\_predict(df[[‘rollno’ , ‘marks’]])

predicted