# -\*- coding: utf-8 -\*-

"""

Created on Wed Mar 23 11:56:02 2022

@author: krish

"""

import pandas as pd

df = pd.DataFrame({'A':[1,2,3],

'B':[4,5,6],

'C':[7,8,9],

'D':[1,3,5],

'E':[5,3,6],

'F':[7,4,3]})

print (df)

# A B C D E F

# 0 1 4 7 1 5 7

# 1 2 5 8 3 3 4

# 2 3 6 9 5 6 3

print(df.iloc[:, :-1])

# A B C D E

# 0 1 4 7 1 5

# 1 2 5 8 3 3

# 2 3 6 9 5 6

X = df.iloc[:, :-1].values

print(X)

# [[1 4 7 1 5]

# [2 5 8 3 3]

# [3 6 9 5 6]]

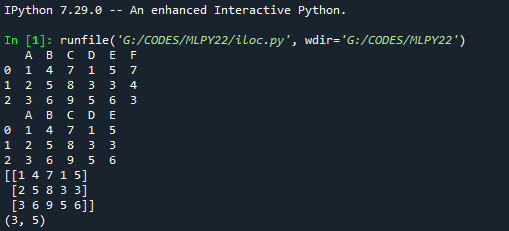
print (X.shape)

(3, 5)

y=df.iloc[:, 1].values

# [4 5 6]

Output:



# import matplotlib.pyplot as plt

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

# y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

# plt.scatter(x, y)

# plt.show()

import matplotlib.pyplot as plt

from scipy import stats

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

y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

slope, intercept, r, p, std\_err = stats.linregress(x, y)

print("slope, intercept, r, p, std\_err",slope, intercept, r, p, std\_err)

def myfunc(x):

return slope \* x + intercept

mymodel = list(map(myfunc, x))

# print(map(myfunc, x))

# print(x,mymodel)

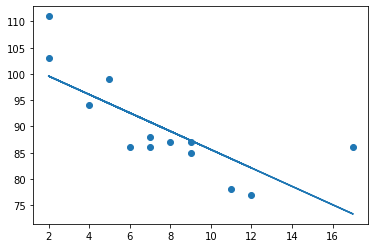
plt.scatter(x, y)

plt.plot(x, mymodel)

plt.show()

newx=15;

print("y(newx)=",myfunc(newx))



output:

y(newx)= 76.83664459161147

program 3: (Record Program1)

# -\*- coding: utf-8 -\*-

"""

Created on Wed Mar 23 11:04:08 2022

@author: krish

Liner Regression2

"""

# import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Importing the dataset

dataset = pd.read\_csv('salary\_data.csv')

X = dataset.iloc[:, :-1].values #get a copy of dataset exclude last column

y = dataset.iloc[:, 1].values #get array of dataset in column 1st (1: index)

# Splitting the dataset into the Training set and Test set

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=1/3, random\_state=0)

# Fitting Simple Linear Regression to the Training set

from sklearn.linear\_model import LinearRegression

regressor = LinearRegression()

regressor.fit(X\_train, y\_train)

# Visualizing the Training set results

viz\_train = plt

viz\_train.scatter(X\_train, y\_train, color='red')

viz\_train.plot(X\_train, regressor.predict(X\_train), color='blue')

viz\_train.title('Salary VS Experience (Training set)')

viz\_train.xlabel('Year of Experience')

viz\_train.ylabel('Salary')

viz\_train.show()

# Visualizing the Test set results

viz\_test = plt

viz\_test.scatter(X\_test, y\_test, color='red')

viz\_test.plot(X\_train, regressor.predict(X\_train), color='blue')

viz\_test.title('Salary VS Experience (Test set)')

viz\_test.xlabel('Year of Experience')

viz\_test.ylabel('Salary')

viz\_test.show()

# Predicting the result of 5 Years Experience

rows, cols = (1, 1)

arr = [[5 for i in range(cols)] for j in range(rows)]

print(arr)

y\_pred = regressor.predict(arr)

print(y\_pred)

s=y\_pred

print("Salary for 5 years of Experience= ")

print(s)

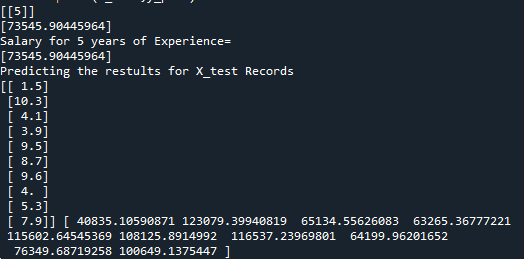
# Predicting the result of X\_test

print("Predicting the restults for X\_test Records")

y\_pred = regressor.predict(X\_test)

print(X\_test,y\_pred)

output:



**RECORD PROGRAM2:**

# -\*- coding: utf-8 -\*-

"""

Created on Wed Apr 20 09:11:24 2022

@author: krish

mysql db connectivity

"""

"""

Install MySQL Driver

Python needs a MySQL driver to access the MySQL database.

In this tutorial we will use the driver "MySQL Connector".

We recommend that you use PIP to install "MySQL Connector".

PIP is most likely already installed in your Python environment.

Navigate your command line to the location of PIP, and type the following:

Download and install "MySQL Connector":

# pip install mysql-connector-python

"""

pip install mysql-connector-python

import mysql.connector as mysql

db = mysql.connect(

host = "localhost",

charset ='utf8',

user = "root",

passwd = "2009"

)

# create a database named "mydatabase":

cursor = db.cursor()

cursor.execute("CREATE DATABASE vgnt;")

cursor.execute("SHOW DATABASES")

for x in cursor:

print(x)

cursor.execute("USE vgnt")

cursor.execute("CREATE TABLE customers (name VARCHAR(255), address VARCHAR(255))")

sql = "INSERT INTO customers (name, address) VALUES (%s, %s)"

val = ("John", "Highway 21")

cursor.execute(sql, val)

db.commit()

print(cursor.rowcount, "record inserted.")

sql = "INSERT INTO customers (name, address) VALUES (%s, %s)"

val = [

('Peter', 'Lowstreet 4'),

('Amy', 'Apple st 652'),

('Hannah', 'Mountain 21'),

('Michael', 'Valley 345'),

('Sandy', 'Ocean blvd 2'),

('Betty', 'Green Grass 1'),

('Richard', 'Sky st 331'),

('Susan', 'One way 98'),

('Vicky', 'Yellow Garden 2'),

('Ben', 'Park Lane 38'),

('William', 'Central st 954'),

('Chuck', 'Main Road 989'),

('Viola', 'Sideway 1633')

]

cursor.executemany(sql, val)

db.commit()

print(cursor.rowcount, "was inserted.")

cursor.execute("select \* from customers")

for x in cursor:

print(x)

sql = "INSERT INTO customers (name, address) VALUES (%s, %s)"

val = ("Michelle", "Blue Village")

cursor.execute(sql, val)

db.commit()

print("1 record inserted, ID:", cursor.lastrowid)

# Select records where the address contains the word "way":

sql = "SELECT \* FROM customers WHERE address ='Park Lane 38'"

cursor.execute(sql)

myresult = cursor.fetchall()

for x in myresult:

print(x)

sql = "SELECT \* FROM customers WHERE address LIKE '%way%'"

cursor.execute(sql)

result = cursor.fetchall()

for x in result:

print(x)

# Sort the result alphabetically by name: result:

sql = "SELECT \* FROM customers ORDER BY name"

cursor.execute(sql)

for x in cursor:

print(x)

# Delete any record where the address is "Mountain 21":

sql = "DELETE FROM customers WHERE address = 'Mountain 21'"

cursor.execute(sql)

db.commit()

print(cursor.rowcount, "record(s) deleted")

cursor.execute("show tables")

for x in cursor:

print(x)

cursor.execute("select \* from customers")

for x in cursor:

print(x)

# Delete the table "customers":

sql = "DROP TABLE customers"

cursor.execute(sql)

cursor.execute("show tables")

for x in cursor:

print(x)

OUTPUT: