**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

Program: B.Tech\MBA.Tech

**Course: Machine Learning**

**Experiment No.04**

PART A

(PART A : TO BE REFFERED BY STUDENTS)

**A.1 Aim:** To implement simple linear regression

**A.2 Prerequisite:**

Python Programming, Pandas library, Numpy Library, MatplotLib, Seaborn Library

**A.3 Outcome:**

**After successful completion of this experiment students will be able to:**

1. Implement simple linear regression by using NumPy arrays
2. Implement and verify the results using Sklearn package.

**A.4 Theory:**

**A.4.1 Linear Regression**

The mathematical approach to find the relationship between two or more variables is known as Regression in AI. Regression is widely used in [Machine](https://www.ssla.co.uk/) Learning to predict the behavior of one variable depending upon the value of another variable.

Linear regression is used for finding linear relationship between target and one or more predictors. Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).

There are two types of linear regression- Simple and Multiple.

When there is a single input variable (x), the method is referred to as simple linear regression. When there are multiple input variables, refers to the method as multiple linear regression.

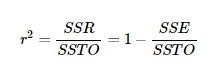
One additional coefficient is also added, giving the line an additional degree of freedom (e.g. moving up and down on a two-dimensional plot) and is often called the intercept or the bias coefficient.

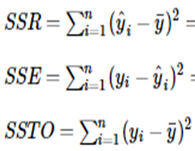
For example, in a simple regression problem (a single x and a single y), the form of the model would be:

y = b0 + b1\*x

In higher dimensions when we have more than one input (x), the line is called a plane or a hyper-plane. The representation therefore is the form of the equation and the specific values used for the coefficients (e.g. b0 and b1 in the above example).

**Coefficient of determination:**





**Tasks:**

**Task 1:**

* + - 1. Define two Numpy arrays x and y which represents BMI and Cholesterol

x= [5,15,25,35,45,55] and y= [11,16,18,30,22,38].

* + - 1. Plot a scatter plot of x and y.
      2. Write a python function to compute the values of the coefficients of linear regression, b0 and b1.
      3. Determine the predicted value of y for x=27.
      4. Plot the regression line on the scatter plot.
      5. Write a python function to determine the value of r2.

**Task 2:**

* + - 1. Import LinearRegression from SKlearn.

1. Reshape x to make it two dimensional array.
2. Create a model for linear regression.
3. Train the model using model.fit
4. Determine the value of intercept (b0) and slope(b1). Compare the values as obtained from task 1.
5. Determine the value of r2. Compare the value with the one obtained in task 1.

**Task 3:**

* + - 1. Import salary.csv into your notebooks.
      2. Explore the dataset using head and describe.
      3. Repeat steps 2 to 6 from task 2.
      4. Plot a scatter plot of work experience vs salary
      5. You can cross verify with your function written by

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

***(Students must submit the soft copy as per following segments within two hours of the practical.)***

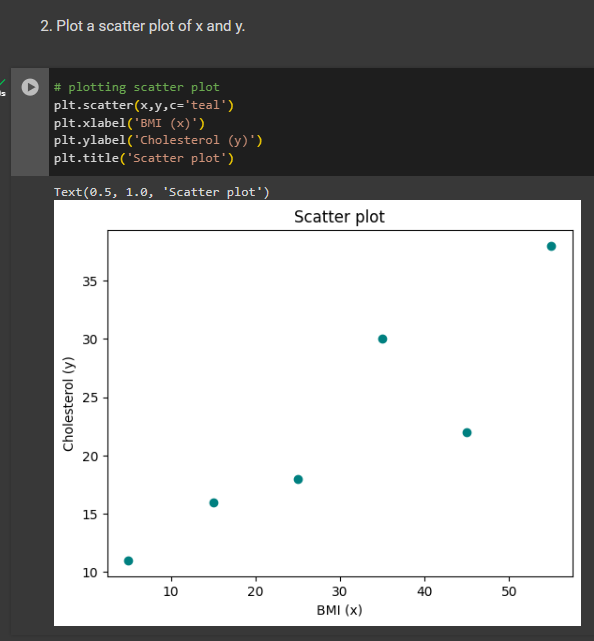
|  |  |
| --- | --- |
| Roll No. N052 | Name: Pratyush Kumar |
| Class : MBA Tech CE (div. D) | Batch : B2 |
| Date of Experiment: 20-01-2024 | Date of Submission: 27-01-2024 |
| Grade : |  |

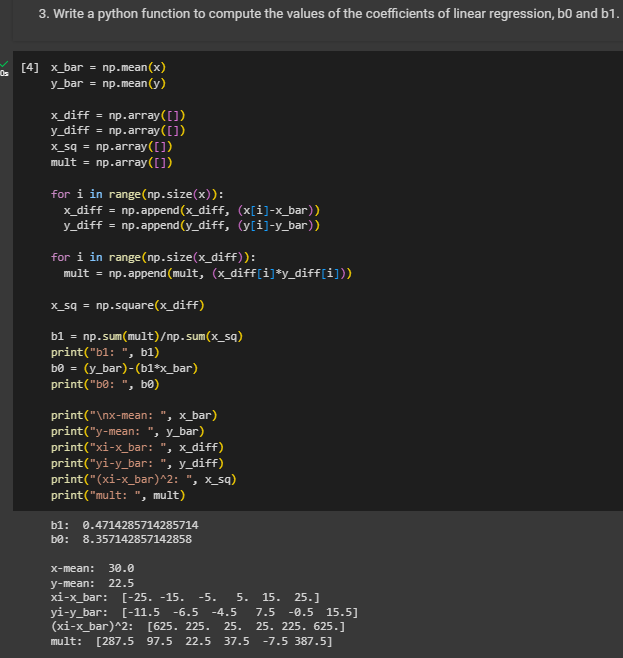
**B.1 Tasks 1**

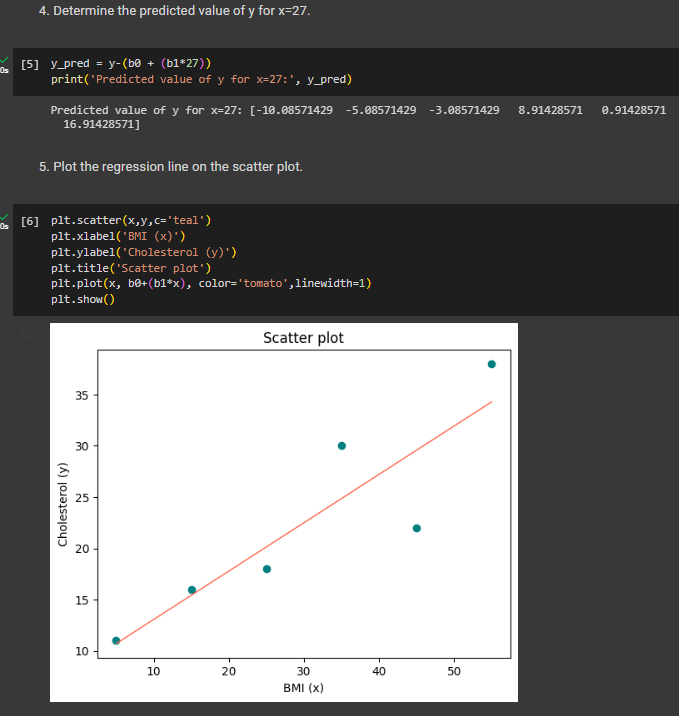
**Colab link:** <https://colab.research.google.com/drive/1wWsVDblqw_R0zST1RmabmnWRMq458xCj?usp=sharing>

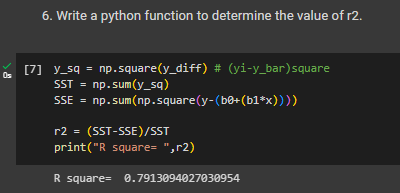
* **Source Code**
* *"""  
   \* This file contains code snippets to implement numpy  
   \* ML-E4-Task1  
   \*  
   \* Original file is located at: https://colab.research.google.com/drive/1wWsVDblqw\_R0zST1RmabmnWRMq458xCj?usp=sharing  
   \* @author Pratyush Kumar (github.com/pratyushgta)  
  """*"""  
  ## EXPERIMENT 4 - Task 1  
  """  
    
  import numpy as np  
  import pandas as pd  
  from scipy import stats  
  # for min\_max scaling  
  from mlxtend.preprocessing import minmax\_scaling  
  # plotting modules  
  import seaborn as sns  
  import matplotlib.pyplot as plt  
  # set seed for reproducibility  
  #np.random.seed(0)  
    
  """1. Define two Numpy arrays x and y which represents BMI and Cholesterol  
  x= [5,15,25,35,45,55] and y= [11,16,18,30,22,38]  
  """  
    
  x= np.array([5,15,25,35,45,55])  
  y= np.array([11,16,18,30,22,38])  
    
  """2. Plot a scatter plot of x and y."""  
    
  # plotting scatter plot  
  plt.scatter(x,y,c='teal')  
  plt.xlabel('BMI (x)')  
  plt.ylabel('Cholesterol (y)')  
  plt.title('Scatter plot')  
    
  """3. Write a python function to compute the values of the coefficients of linear regression, b0 and b1."""  
    
  x\_bar = np.mean(x)  
  y\_bar = np.mean(y)  
    
  x\_diff = np.array([])  
  y\_diff = np.array([])  
  x\_sq = np.array([])  
  mult = np.array([])  
    
  for i in range(np.size(x)):  
   x\_diff = np.append(x\_diff, (x[i]-x\_bar))  
   y\_diff = np.append(y\_diff, (y[i]-y\_bar))  
    
  for i in range(np.size(x\_diff)):  
   mult = np.append(mult, (x\_diff[i]\*y\_diff[i]))  
    
  x\_sq = np.square(x\_diff)  
    
  b1 = np.sum(mult)/np.sum(x\_sq)  
  print("b1: ", b1)  
  b0 = (y\_bar)-(b1\*x\_bar)  
  print("b0: ", b0)  
    
  print("\nx-mean: ", x\_bar)  
  print("y-mean: ", y\_bar)  
  print("xi-x\_bar: ", x\_diff)  
  print("yi-y\_bar: ", y\_diff)  
  print("(xi-x\_bar)^2: ", x\_sq)  
  print("mult: ", mult)  
    
  """4. Determine the predicted value of y for x=27."""  
    
  y\_pred = y-(b0 + (b1\*27))  
  print('Predicted value of y for x=27:', y\_pred)  
    
  """5. Plot the regression line on the scatter plot."""  
    
  plt.scatter(x,y,c='teal')  
  plt.xlabel('BMI (x)')  
  plt.ylabel('Cholesterol (y)')  
  plt.title('Scatter plot')  
  plt.plot(x, b0+(b1\*x), color='tomato',linewidth=1)  
  plt.show()  
    
  """6. Write a python function to determine the value of r2."""  
    
  y\_sq = np.square(y\_diff) # (yi-y\_bar)square  
  SST = np.sum(y\_sq)  
  SSE = np.sum(np.square(y-(b0+(b1\*x))))  
    
  r2 = (SST-SSE)/SST  
  print("R square= ",r2)
* **Input/ Output**

****

****

****

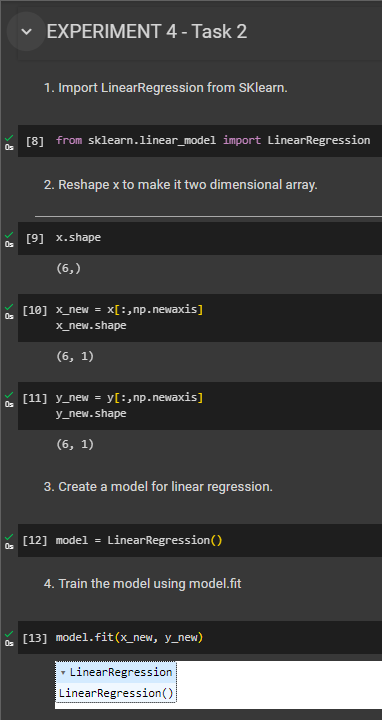
****

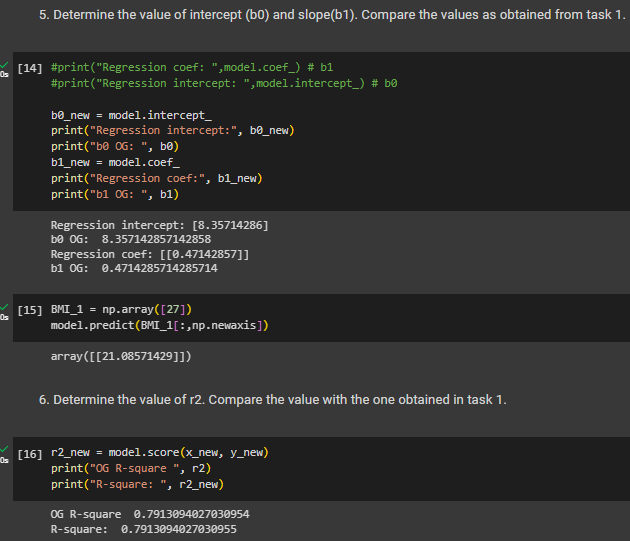
****

**B.2 Tasks 2**

**Colab link:** <https://colab.research.google.com/drive/1wWsVDblqw_R0zST1RmabmnWRMq458xCj?usp=sharing>

* **Source Code**
* *"""  
   \* This file contains code snippets to implement Linear Regression from sklearn  
   \* ML-E4-Task2  
   \*  
   \* Original file is located at: https://colab.research.google.com/drive/1wWsVDblqw\_R0zST1RmabmnWRMq458xCj?usp=sharing  
   \* @author Pratyush Kumar (github.com/pratyushgta)  
  """*"""## EXPERIMENT 4 - Task 2  
    
  1. Import LinearRegression from SKlearn.  
  """  
  import numpy as np  
  from sklearn.linear\_model import LinearRegression  
    
  """2. Reshape x to make it two dimensional array."""  
    
  x.shape  
    
  x\_new = x[:,np.newaxis]  
  x\_new.shape  
    
  y\_new = y[:,np.newaxis]  
  y\_new.shape  
    
  """3. Create a model for linear regression."""  
    
  model = LinearRegression()  
    
  """4. Train the model using model.fit"""  
    
  model.fit(x\_new, y\_new)  
    
  """5. Determine the value of intercept (b0) and slope(b1). Compare the values as obtained from task 1."""  
    
  #print("Regression coef: ",model.coef\_) # b1  
  #print("Regression intercept: ",model.intercept\_) # b0  
    
  b0\_new = model.intercept\_  
  print("Regression intercept:", b0\_new)  
  print("b0 OG: ", b0)  
  b1\_new = model.coef\_  
  print("Regression coef:", b1\_new)  
  print("b1 OG: ", b1)  
  BMI\_1 = np.array([27])  
  model.predict(BMI\_1[:,np.newaxis])  
    
  """6. Determine the value of r2. Compare the value with the one obtained in task 1"""  
  r2\_new = model.score(x\_new, y\_new)  
  print("OG R-square ", r2)  
  print("R-square: ", r2\_new)
* **Input/ Output**

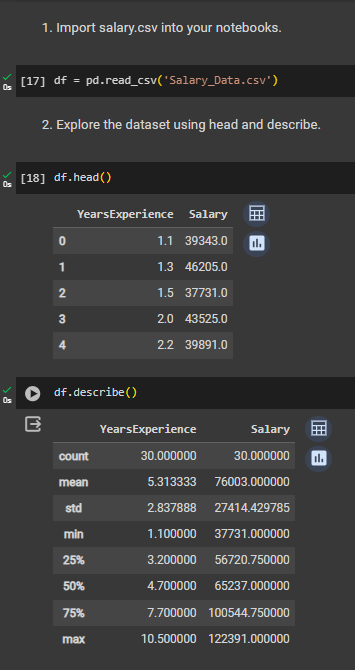
****

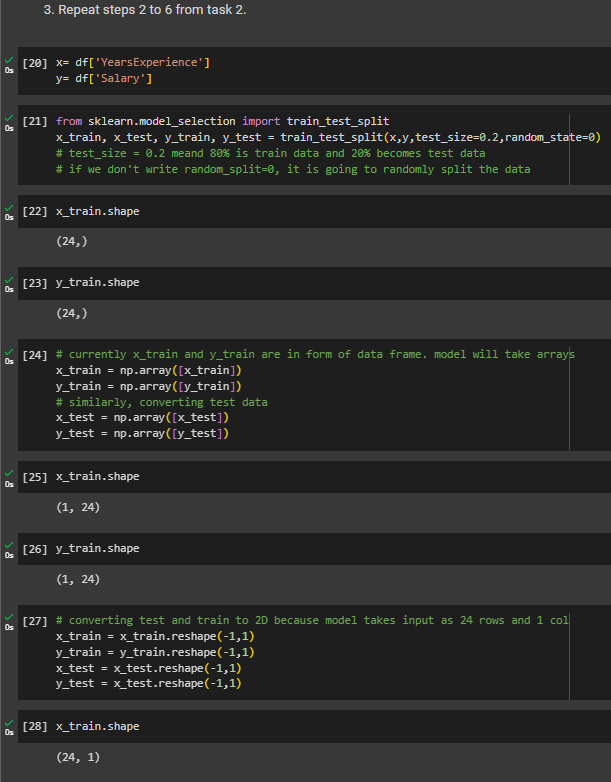
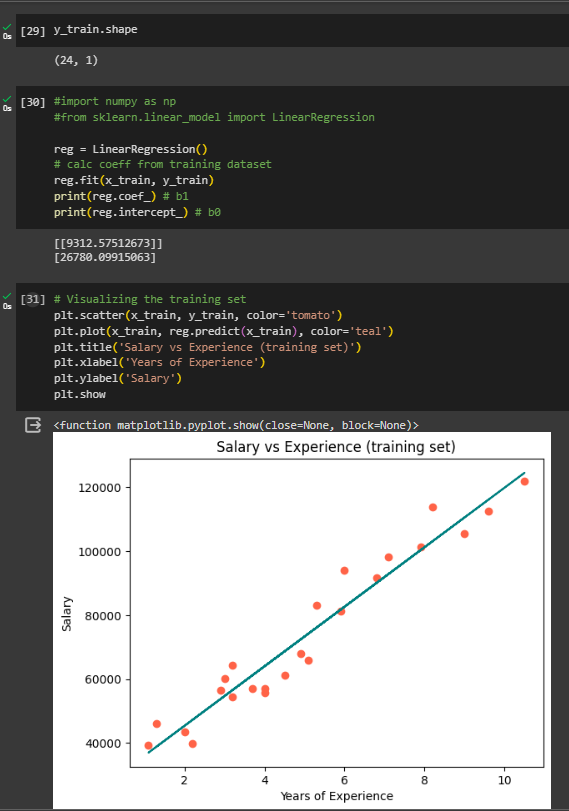
****

**B.3 Tasks 3**

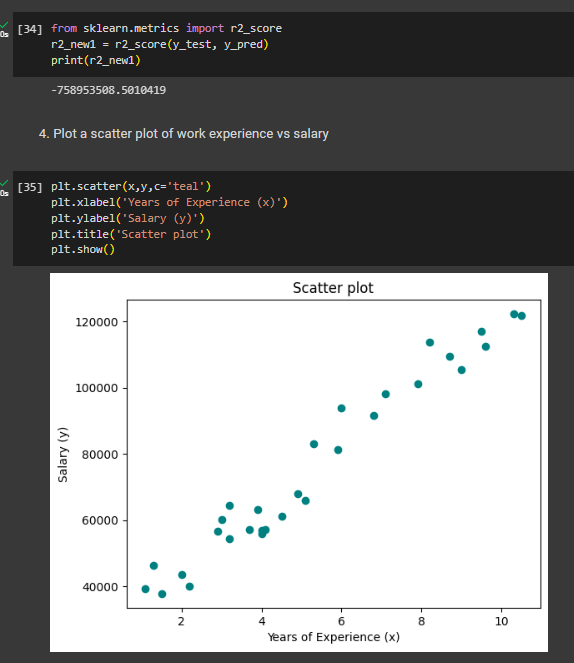
**Colab link:** <https://colab.research.google.com/drive/1wWsVDblqw_R0zST1RmabmnWRMq458xCj?usp=sharing>

* **Source Code**
* *"""  
   \* This file contains code snippets for performing exploratory data analysis on salary dataset  
   \* ML-E4-Task3  
   \*  
   \* Original file is located at: https://colab.research.google.com/drive/1wWsVDblqw\_R0zST1RmabmnWRMq458xCj?usp=sharing  
   \* @author Pratyush Kumar (github.com/pratyushgta)  
  """*"""## EXPERIMENT 4 - Task 3  
    
  1. Import salary.csv into your notebooks.  
  """  
  import numpy as np  
  import pandas as pd  
  import matplotlib.pyplot as plt  
    
  df = pd.read\_csv('Salary\_Data.csv')  
    
  """2. Explore the dataset using head and describe."""  
    
  df.head()  
    
  df.describe()  
    
  """3. Repeat steps 2 to 6 from task 2."""  
    
  x= df['YearsExperience']  
  y= df['Salary']  
    
  from sklearn.model\_selection import train\_test\_split  
  x\_train, x\_test, y\_train, y\_test = train\_test\_split(x,y,test\_size=0.2,random\_state=0)  
  # test\_size = 0.2 meand 80% is train data and 20% becomes test data  
  # if we don't write random\_split=0, it is going to randomly split the data  
    
  x\_train.shape  
    
  y\_train.shape  
    
  # currently x\_train and y\_train are in form of data frame. model will take arrays  
  x\_train = np.array([x\_train])  
  y\_train = np.array([y\_train])  
  # similarly, converting test data  
  x\_test = np.array([x\_test])  
  y\_test = np.array([y\_test])  
    
  x\_train.shape  
    
  y\_train.shape  
    
  # converting test and train to 2D because model takes input as 24 rows and 1 col  
  x\_train = x\_train.reshape(-1,1)  
  y\_train = y\_train.reshape(-1,1)  
  x\_test = x\_test.reshape(-1,1)  
  y\_test = x\_test.reshape(-1,1)  
    
  x\_train.shape  
    
  y\_train.shape  
    
  #import numpy as np  
  #from sklearn.linear\_model import LinearRegression  
    
  reg = LinearRegression()  
  # calc coeff from training dataset  
  reg.fit(x\_train, y\_train)  
  print(reg.coef\_) # b1  
  print(reg.intercept\_) # b0  
    
  # Visualizing the training set  
  plt.scatter(x\_train, y\_train, color='tomato')  
  plt.plot(x\_train, reg.predict(x\_train), color='teal')  
  plt.title('Salary vs Experience (training set)')  
  plt.xlabel('Years of Experience')  
  plt.ylabel('Salary')  
  plt.show  
    
  # predicting the test set results  
  y\_pred = reg.predict(x\_test)  
  y\_pred  
    
  # Visualizing the test set  
  plt.scatter(x\_test, y\_test, color='tomato')  
  plt.plot(x\_test, reg.predict(x\_test), color='teal')  
  plt.title('Salary vs Experience (testing set)')  
  plt.xlabel('Years of Experience')  
  plt.ylabel('Salary')  
  plt.show  
    
  from sklearn.metrics import r2\_score  
  r2\_new1 = r2\_score(y\_test, y\_pred)  
  print(r2\_new1)  
    
  """4. Plot a scatter plot of work experience vs salary"""  
    
  plt.scatter(x,y,c='teal')  
  plt.xlabel('Years of Experience (x)')  
  plt.ylabel('Salary (y)')  
  plt.title('Scatter plot')  
  plt.show()
* **Input/ Output**

****

****

****

****

**B.4 Conclusion:**

*(Students must write the conclusion in their own words.)*

Implemented simple linear regression using NumPy arrays and validated the results with the Sklearn package.