Experiment No:3

Course Outcome: CO2 **Blooms Level: L3**

Aim: To implement Linear Regression

Abstract:

Linear Regression is a fundamental statistical and machine learning technique used to model the relationship between a dependent variable and one or more independent variables. It aims to fit a linear equation to observed data, enabling predictions and insights into data trends.

Multiple Linear Regression (MLR) is a statistical technique used to model the relationship between one dependent variable and two or more independent variables.

Sample Input and Output:

Case 1:

Temperat	Revenue
24.56688	534.799
26.00519	625.1901
27.79055	660.6323
20.59534	487.707
11.5035	316.2402
14.35251	367.9407
13.70778	308.8945
30.83398	696.7166
0.97687	55.39034
31.66946	737.8008
11.45525	325.9684
3.66467	71.16015

Sample Output

The linear regression model learns a relationship of the form:

Revenue=m·Temperature+c

Example

$$\text{Revenue} = 25.6 \cdot \text{Temperature} + 100$$

Case 2:

Area (sq ft)	Price ($)
800	150,000
950	175,000
1100	200,000
1300	230,000
1500	260,000
1700	295,000
2000	340,000
2200	370,000
2500	410,000
2800	450,000

This table can be used as a training dataset for a **Simple Linear Regression** model to learn the relationship:

$$\text{Price} = m \cdot \text{Area} + c$$

Case 3:

Experience (X)	Salary (y) (in lakhs)
2	3
6	10
5	4
7	13

$$y = -2.14 + 1.928x$$

Theory:

LINEAR REGRESSION

Regression falls under supervised learning where the system tries to predict a value for an input based on previous information.

Simple Linear regression involves finding the "best" line to fit two attributes (or variables) so that one attribute (x) can be used to predict the other (y).

Regression involves two variables:

i. First variable : denoted by x : predictor, explanatory or independent variable

ii.Second variable : denoted by y : response, outcome or dependent variable

Mathematical Representation –

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

$$y = \beta 0 + \beta 1 x$$

β values are called **model coefficients**. These values are "learned" during the model fitting/training step, where :

$$\beta_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$$

and

$$\beta_0 = \frac{1}{n}(\sum y_i - \beta_1 \sum x_i) = \bar{y} - \beta_1 \bar{x}$$

Multiple Linear Regression: A linear regression model with more than one independent variable and one dependent variable. Polynomial Regression is a type of Regression analysis that models the relationship of values of the Dependent variable "x" and Independent variables "y'' as non linear. It is a special case of Multiple Linear Regression.

Program:

1. You own an ice cream business and you would like to create a model that could predict the daily revenue in dollars based on the outside air temperature (degC). You decide that a Linear Regression model might be a good candidate to solve this problem.
   Independent variable X: Outside Air Temperature
   Dependant Variable Y: Overall daily revenue generated in dollars.

Output:

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
C:\Users\Ramya\Anaconda3\lib\site-packages\statsmodels\tools\_testing.py:19: FutureWarning: pandas.ut
il.testing is deprecated. Use the functions in the public API at pandas.testing instead.
  import pandas.util.testing as tm
```

In [2]:
```
IceCream = pd.read_csv(r'F:\ML\datasets\IceCream.csv')
```

In [3]:
```
IceCream.head(100)
```

Out[3]:

	Temperature	Revenue
0	24.566884	534.799028
1	26.005191	625.190122
2	27.790554	660.632289
3	20.595335	487.706960
4	11.503498	316.240194
...	...	...
95	9.018860	212.591740
96	20.265012	474.749392
97	19.363153	460.402500
98	14.685944	343.362905
99	9.954357	283.834327

100 rows × 2 columns

In [4]:
```
IceCream.tail()
```

Out[4]:

	Temperature	Revenue
495	22.274899	524.746364
496	32.893092	755.818399
497	12.588157	306.090719
498	22.362402	566.217304
499	28.957736	655.660388

In [5]:
```
IceCream.describe()
```

Out[5]:

	Temperature	Revenue
count	500.000000	500.000000
mean	22.232225	521.570777
std	8.096388	175.404751
min	0.000000	10.000000
25%	17.122258	405.558681
50%	22.392791	529.368565
75%	27.740674	642.257922
max	45.000000	1000.000000

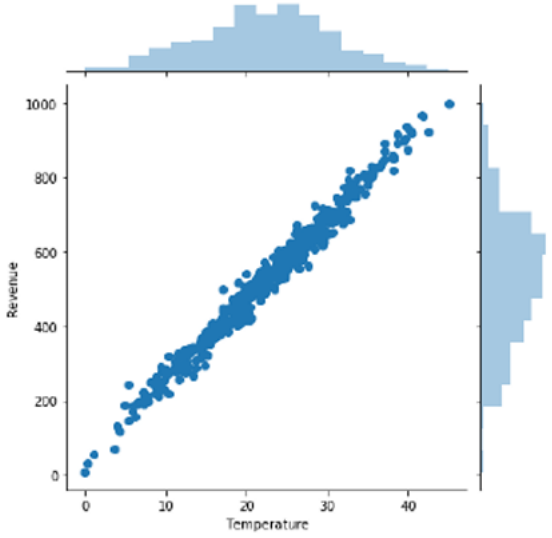
In [6]:
```
IceCream.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Temperature  500 non-null    float64
 1   Revenue      500 non-null    float64
dtypes: float64(2)
memory usage: 7.9 KB
```

VISUALIZE DATASET

In [7]:
```
sns.jointplot(x='Temperature', y='Revenue', data = IceCream)
```

Out[7]: <seaborn.axisgrid.JointGrid at 0x1d6f3e72a08>



CREATE TESTING AND TRAINING DATASET

```
In [10]: y = IceCream['Revenue']
```

```
In [11]: X = IceCream[['Temperature']]
```

```
In [12]: X
```

Out[12]:

	Temperature
0	24.566884
1	26.005191
2	27.790554
3	20.595335
4	11.503498
...	...
495	22.274899
496	32.893092
497	12.588157
498	22.362402
499	28.957736

500 rows × 1 columns

```
In [13]: from sklearn.model_selection import train_test_split
```

```
In [14]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)
```

TRAIN THE MODEL

```
In [15]: X_train.shape
```

```
Out[15]: (375, 1)
```

```
In [16]: from sklearn.linear_model import LinearRegression
```

```
In [17]: regressor = LinearRegression(fit_intercept =True)
```

```
In [18]: regressor.fit(X_train,y_train)
```

```
Out[18]: LinearRegression()
```

```
In [19]: print('Linear Model Coefficient (m): ', regressor.coef_)
         print('Linear Model Coefficient (b): ', regressor.intercept_)

         Linear Model Coefficient (m):  [21.60370979]
         Linear Model Coefficient (b):  41.23376038174729
```

TEST THE MODEL

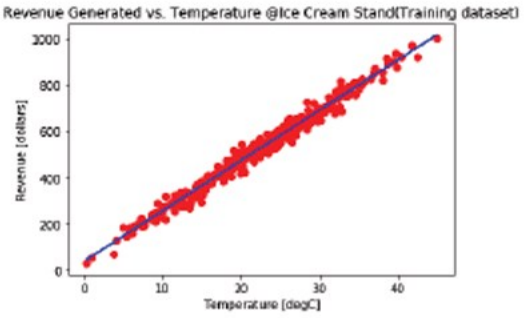
```
In [20]: y_pred = regressor.predict( X_test)

In [21]: y_test

Out[21]: 171    366.247714
         30     528.380417
         228    850.246982
         349    482.571988
         215    562.792463
                   ...
         74     586.150568
         131    417.354839
         87     344.688765
         440    684.158444
         462    297.025414
         Name: Revenue, Length: 125, dtype: float64

In [22]: plt.scatter(X_train, y_train, color = 'red')
         plt.plot(X_train, regressor.predict(X_train), color = 'blue')
         plt.ylabel('Revenue [dollars]')
         plt.xlabel('Temperature [degC]')
         plt.title('Revenue Generated vs. Temperature @Ice Cream Stand(Training dataset)')

Out[22]: Text(0.5, 1.0, 'Revenue Generated vs. Temperature @Ice Cream Stand(Training dataset)')
```



A Multiple Linear Regression model for predicting profit of a company according to various factors. Dataset consists of five features(first 4 are independent features and remaining last is dependent)

In [1]:
```
#importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
#importing dataset and declaring independent and dependent variable
#the dependent variable is in last column, hence y is assigned with [:,-1] (meaning all rows of last co
lumn)
ds = pd.read_csv('C:/Users/Ramya/Desktop/ML/datasets/50_Startups.csv')
x = ds.iloc[:,:-1].values
y = ds.iloc[:,-1].values
```

In [3]:
```
#checking our dataset
ds.head(10)
```

Out[3]:

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94
5	131876.90	99814.71	362861.36	New York	156991.12
6	134615.46	147198.87	127716.82	California	156122.51
7	130298.13	145530.06	323876.68	Florida	155752.60
8	120542.52	148718.95	311613.29	New York	152211.77
9	123334.88	108679.17	304981.62	California	149759.96

In [4]:
```
#checking for missing values
ds.isnull().any()
```

Out[4]:
```
R&D Spend          False
Administration     False
Marketing Spend    False
State              False
Profit             False
dtype: bool
```

no missing data

In [5]:
```
#categorical data found in column [3]
#encoding categorical data using OneHotEncoding
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
ct = ColumnTransformer(transformers = [('encoding',OneHotEncoder(),[3])], remainder = 'passthrough')
x = np.array(ct.fit_transform(x))
```

```
In [7]: #splitting dataset in training and testing set
        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)

In [8]: #training the model
        from sklearn.linear_model import LinearRegression
        lr = LinearRegression()
        lr.fit(x_train,y_train)

Out[8]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)

In [9]: #comparing the predicted values of the model with the actual values of the model
        y_pred = lr.predict(x_test)
        np.set_printoptions(precision = 2)
        print(np.concatenate((y_pred.reshape(-1,1),y_test.reshape(-1,1)),1))

        [[103015.2  103282.38]
         [132582.28 144259.4 ]
         [132447.74 146121.95]
         [ 71976.1   77798.83]
         [178537.48 191050.39]
         [116161.24 105008.31]
         [ 67851.69  81229.06]
         [ 98791.73  97483.56]
         [113969.44 110352.25]
         [167921.07 166187.94]]
```

AS YOU CAN SEE THE COLUMN [0] HAS PREDICTED VALUES AND COLUMN [1] HAS ACTUAL VALUES, BOTH THE VALUES ARE PRTTY CLOSE WHICH SUGGESTS THAT OUR MODEL DID A GOOD JOB IN PREDICTING VALUES.

```
In [10]: #getting accuracy of the model
         from sklearn.metrics import r2_score
         r2_score(y_test,lr.predict(x_test))

Out[10]: 0.9347068473282303
```

THE ACCURACY FOR THE MODEL CAME OUT TO BE 93.4% WHICH IS PRETTY GOOD.

Conclusion: Linear Regression is implemented on the Ice cream Dataset where temperature is the independent variable and Revenue is the dependent variable. A Multiple Linear Regression model is built on the Startup dataset with four independent variables namely R&D Spend, Administration, Marketing Spend and State and the dependent variable Profit. State being a categorical variable is one hot encoded before model building.

Exercise 1:

You work in the real estate sector and you would like to create a model that could predict the selling price of a property based on its area in square feet. You decide that a Linear Regression model might be a good candidate to solve this problem.
Data set:

- Independent variable X: Area of the property (in square feet)
- Dependent variable Y: Selling price of the property (in dollars)

Students shall draw flowchart of exercise question in the writeup and submit.

Exercise 2:
Predict the total fare amount of a Chicago taxi trip based on trip distance and duration using a Linear Regression model

Students shall draw flowchart of exercise question in the writeup and submit.