----- MLR - Multi Linear Regression Model ------

In [2]:
```
# importing libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [3]:
```
# Load the dataset

dataset = pd.read_csv(r"C:\Users\Jan Saida\OneDrive\Documents\Desktop\Excel sheets\Investment.csv")
dataset
```

Out[3]:

	DigitalMarketing	Promotion	Research	State	Profit
0	165349.20	136897.80	471784.10	Hyderabad	192261.83
1	162597.70	151377.59	443898.53	Bangalore	191792.06
2	153441.51	101145.55	407934.54	Chennai	191050.39
3	144372.41	118671.85	383199.62	Hyderabad	182901.99
4	142107.34	91391.77	366168.42	Chennai	166187.94
5	131876.90	99814.71	362861.36	Hyderabad	156991.12
6	134615.46	147198.87	127716.82	Bangalore	156122.51
7	130298.13	145530.06	323876.68	Chennai	155752.60
8	120542.52	148718.95	311613.29	Hyderabad	152211.77
9	123334.88	108679.17	304981.62	Bangalore	149759.96
10	101913.08	110594.11	229160.95	Chennai	146121.95
11	100671.96	91790.61	249744.55	Bangalore	144259.40
12	93863.75	127320.38	249839.44	Chennai	141585.52
13	91992.39	135495.07	252664.93	Bangalore	134307.35
14	119943.24	156547.42	256512.92	Chennai	132602.65
15	114523.61	122616.84	261776.23	Hyderabad	129917.04
16	78013.11	121597.55	264346.06	Bangalore	126992.93
17	94657.16	145077.58	282574.31	Hyderabad	125370.37
18	91749.16	114175.79	294919.57	Chennai	124266.90
19	86419.70	153514.11	0.00	Hyderabad	122776.86
20	76253.86	113867.30	298664.47	Bangalore	118474.03
21	78389.47	153773.43	299737.29	Hyderabad	111313.02

	DigitalMarketing	Promotion	Research	State	Profit
22	73994.56	122782.75	303319.26	Chennai	110352.25
23	67532.53	105751.03	304768.73	Chennai	108733.99
24	77044.01	99281.34	140574.81	Hyderabad	108552.04
25	64664.71	139553.16	137962.62	Bangalore	107404.34
26	75328.87	144135.98	134050.07	Chennai	105733.54
27	72107.60	127864.55	353183.81	Hyderabad	105008.31
28	66051.52	182645.56	118148.20	Chennai	103282.38
29	65605.48	153032.06	107138.38	Hyderabad	101004.64
30	61994.48	115641.28	91131.24	Chennai	99937.59
31	61136.38	152701.92	88218.23	Hyderabad	97483.56
32	63408.86	129219.61	46085.25	Bangalore	97427.84
33	55493.95	103057.49	214634.81	Chennai	96778.92
34	46426.07	157693.92	210797.67	Bangalore	96712.80
35	46014.02	85047.44	205517.64	Hyderabad	96479.51
36	28663.76	127056.21	201126.82	Chennai	90708.19
37	44069.95	51283.14	197029.42	Bangalore	89949.14
38	20229.59	65947.93	185265.10	Hyderabad	81229.06
39	38558.51	82982.09	174999.30	Bangalore	81005.76
40	28754.33	118546.05	172795.67	Bangalore	78239.91
41	27892.92	84710.77	164470.71	Chennai	77798.83
42	23640.93	96189.63	148001.11	Bangalore	71498.49
43	15505.73	127382.30	35534.17	Hyderabad	69758.98

	DigitalMarketing	Promotion	Research	State	Profit
44	22177.74	154806.14	28334.72	Bangalore	65200.33
45	1000.23	124153.04	1903.93	Hyderabad	64926.08
46	1315.46	115816.21	297114.46	Chennai	49490.75
47	0.00	135426.92	0.00	Bangalore	42559.73
48	542.05	51743.15	0.00	Hyderabad	35673.41
49	0.00	116983.80	45173.06	Bangalore	14681.40

In [4]:
```
# Selecting the independent variables (X) and dependent variable (y)

x = dataset.iloc[:, :-1]  # independent variable
y = dataset.iloc[:, 4]    # dependent variable
```

In [5]:
```
x
```

Out[5]:

	DigitalMarketing	Promotion	Research	State
0	165349.20	136897.80	471784.10	Hyderabad
1	162597.70	151377.59	443898.53	Bangalore
2	153441.51	101145.55	407934.54	Chennai
3	144372.41	118671.85	383199.62	Hyderabad
4	142107.34	91391.77	366168.42	Chennai
5	131876.90	99814.71	362861.36	Hyderabad
6	134615.46	147198.87	127716.82	Bangalore
7	130298.13	145530.06	323876.68	Chennai
8	120542.52	148718.95	311613.29	Hyderabad
9	123334.88	108679.17	304981.62	Bangalore
10	101913.08	110594.11	229160.95	Chennai
11	100671.96	91790.61	249744.55	Bangalore
12	93863.75	127320.38	249839.44	Chennai
13	91992.39	135495.07	252664.93	Bangalore
14	119943.24	156547.42	256512.92	Chennai
15	114523.61	122616.84	261776.23	Hyderabad
16	78013.11	121597.55	264346.06	Bangalore
17	94657.16	145077.58	282574.31	Hyderabad
18	91749.16	114175.79	294919.57	Chennai
19	86419.70	153514.11	0.00	Hyderabad
20	76253.86	113867.30	298664.47	Bangalore
21	78389.47	153773.43	299737.29	Hyderabad

	DigitalMarketing	Promotion	Research	State
22	73994.56	122782.75	303319.26	Chennai
23	67532.53	105751.03	304768.73	Chennai
24	77044.01	99281.34	140574.81	Hyderabad
25	64664.71	139553.16	137962.62	Bangalore
26	75328.87	144135.98	134050.07	Chennai
27	72107.60	127864.55	353183.81	Hyderabad
28	66051.52	182645.56	118148.20	Chennai
29	65605.48	153032.06	107138.38	Hyderabad
30	61994.48	115641.28	91131.24	Chennai
31	61136.38	152701.92	88218.23	Hyderabad
32	63408.86	129219.61	46085.25	Bangalore
33	55493.95	103057.49	214634.81	Chennai
34	46426.07	157693.92	210797.67	Bangalore
35	46014.02	85047.44	205517.64	Hyderabad
36	28663.76	127056.21	201126.82	Chennai
37	44069.95	51283.14	197029.42	Bangalore
38	20229.59	65947.93	185265.10	Hyderabad
39	38558.51	82982.09	174999.30	Bangalore
40	28754.33	118546.05	172795.67	Bangalore
41	27892.92	84710.77	164470.71	Chennai
42	23640.93	96189.63	148001.11	Bangalore
43	15505.73	127382.30	35534.17	Hyderabad

	DigitalMarketing	Promotion	Research	State
44	22177.74	154806.14	28334.72	Bangalore
45	1000.23	124153.04	1903.93	Hyderabad
46	1315.46	115816.21	297114.46	Chennai
47	0.00	135426.92	0.00	Bangalore
48	542.05	51743.15	0.00	Hyderabad
49	0.00	116983.80	45173.06	Bangalore

In [6]:
```
y
```

Out[6]:
```
0     192261.83
1     191792.06
2     191050.39
3     182901.99
4     166187.94
5     156991.12
6     156122.51
7     155752.60
8     152211.77
9     149759.96
10    146121.95
11    144259.40
12    141585.52
13    134307.35
14    132602.65
15    129917.04
16    126992.93
17    125370.37
18    124266.90
19    122776.86
20    118474.03
21    111313.02
22    110352.25
23    108733.99
24    108552.04
25    107404.34
26    105733.54
27    105008.31
28    103282.38
29    101004.64
30     99937.59
31     97483.56
32     97427.84
33     96778.92
34     96712.80
35     96479.51
36     90708.19
37     89949.14
38     81229.06
39     81005.76
```

```
40     78239.91
41     77798.83
42     71498.49
43     69758.98
44     65200.33
45     64926.08
46     49490.75
47     42559.73
48     35673.41
49     14681.40
Name: Profit, dtype: float64
```

In [7]:

```
# Apply one-hot encoding to the features (categorical variables)

X = pd.get_dummies(x, dtype=int)
X
```

Out[7]:

	DigitalMarketing	Promotion	Research	State_Bangalore	State_Chennai	State_Hyderabad
0	165349.20	136897.80	471784.10	0	0	1
1	162597.70	151377.59	443898.53	1	0	0
2	153441.51	101145.55	407934.54	0	1	0
3	144372.41	118671.85	383199.62	0	0	1
4	142107.34	91391.77	366168.42	0	1	0
5	131876.90	99814.71	362861.36	0	0	1
6	134615.46	147198.87	127716.82	1	0	0
7	130298.13	145530.06	323876.68	0	1	0
8	120542.52	148718.95	311613.29	0	0	1
9	123334.88	108679.17	304981.62	1	0	0
10	101913.08	110594.11	229160.95	0	1	0
11	100671.96	91790.61	249744.55	1	0	0
12	93863.75	127320.38	249839.44	0	1	0
13	91992.39	135495.07	252664.93	1	0	0
14	119943.24	156547.42	256512.92	0	1	0
15	114523.61	122616.84	261776.23	0	0	1
16	78013.11	121597.55	264346.06	1	0	0
17	94657.16	145077.58	282574.31	0	0	1
18	91749.16	114175.79	294919.57	0	1	0
19	86419.70	153514.11	0.00	0	0	1
20	76253.86	113867.30	298664.47	1	0	0
21	78389.47	153773.43	299737.29	0	0	1

	DigitalMarketing	Promotion	Research	State_Bangalore	State_Chennai	State_Hyderabad
22	73994.56	122782.75	303319.26	0	1	0
23	67532.53	105751.03	304768.73	0	1	0
24	77044.01	99281.34	140574.81	0	0	1
25	64664.71	139553.16	137962.62	1	0	0
26	75328.87	144135.98	134050.07	0	1	0
27	72107.60	127864.55	353183.81	0	0	1
28	66051.52	182645.56	118148.20	0	1	0
29	65605.48	153032.06	107138.38	0	0	1
30	61994.48	115641.28	91131.24	0	1	0
31	61136.38	152701.92	88218.23	0	0	1
32	63408.86	129219.61	46085.25	1	0	0
33	55493.95	103057.49	214634.81	0	1	0
34	46426.07	157693.92	210797.67	1	0	0
35	46014.02	85047.44	205517.64	0	0	1
36	28663.76	127056.21	201126.82	0	1	0
37	44069.95	51283.14	197029.42	1	0	0
38	20229.59	65947.93	185265.10	0	0	1
39	38558.51	82982.09	174999.30	1	0	0
40	28754.33	118546.05	172795.67	1	0	0
41	27892.92	84710.77	164470.71	0	1	0
42	23640.93	96189.63	148001.11	1	0	0
43	15505.73	127382.30	35534.17	0	0	1

	DigitalMarketing	Promotion	Research	State_Bangalore	State_Chennai	State_Hyderabad
44	22177.74	154806.14	28334.72	1	0	0
45	1000.23	124153.04	1903.93	0	0	1
46	1315.46	115816.21	297114.46	0	1	0
47	0.00	135426.92	0.00	1	0	0
48	542.05	51743.15	0.00	0	0	1
49	0.00	116983.80	45173.06	1	0	0

In [8]:
```
# Train-test split

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 [9]:
```
x_train
```

Out[9]:

	DigitalMarketing	Promotion	Research	State_Bangalore	State_Chennai	State_Hyderabad
33	55493.95	103057.49	214634.81	0	1	0
35	46014.02	85047.44	205517.64	0	0	1
26	75328.87	144135.98	134050.07	0	1	0
34	46426.07	157693.92	210797.67	1	0	0
18	91749.16	114175.79	294919.57	0	1	0
7	130298.13	145530.06	323876.68	0	1	0
14	119943.24	156547.42	256512.92	0	1	0
45	1000.23	124153.04	1903.93	0	0	1
48	542.05	51743.15	0.00	0	0	1
29	65605.48	153032.06	107138.38	0	0	1
15	114523.61	122616.84	261776.23	0	0	1
30	61994.48	115641.28	91131.24	0	1	0
32	63408.86	129219.61	46085.25	1	0	0
16	78013.11	121597.55	264346.06	1	0	0
42	23640.93	96189.63	148001.11	1	0	0
20	76253.86	113867.30	298664.47	1	0	0
43	15505.73	127382.30	35534.17	0	0	1
8	120542.52	148718.95	311613.29	0	0	1
13	91992.39	135495.07	252664.93	1	0	0
25	64664.71	139553.16	137962.62	1	0	0
5	131876.90	99814.71	362861.36	0	0	1
17	94657.16	145077.58	282574.31	0	0	1

	DigitalMarketing	Promotion	Research	State_Bangalore	State_Chennai	State_Hyderabad
40	28754.33	118546.05	172795.67	1	0	0
49	0.00	116983.80	45173.06	1	0	0
1	162597.70	151377.59	443898.53	1	0	0
12	93863.75	127320.38	249839.44	0	1	0
37	44069.95	51283.14	197029.42	1	0	0
24	77044.01	99281.34	140574.81	0	0	1
6	134615.46	147198.87	127716.82	1	0	0
23	67532.53	105751.03	304768.73	0	1	0
36	28663.76	127056.21	201126.82	0	1	0
21	78389.47	153773.43	299737.29	0	0	1
19	86419.70	153514.11	0.00	0	0	1
9	123334.88	108679.17	304981.62	1	0	0
39	38558.51	82982.09	174999.30	1	0	0
46	1315.46	115816.21	297114.46	0	1	0
3	144372.41	118671.85	383199.62	0	0	1
0	165349.20	136897.80	471784.10	0	0	1
47	0.00	135426.92	0.00	1	0	0
44	22177.74	154806.14	28334.72	1	0	0

In [10]:
```
x_test
```

Out[10]:

	DigitalMarketing	Promotion	Research	State_Bangalore	State_Chennai	State_Hyderabad
28	66051.52	182645.56	118148.20	0	1	0
11	100671.96	91790.61	249744.55	1	0	0
10	101913.08	110594.11	229160.95	0	1	0
41	27892.92	84710.77	164470.71	0	1	0
2	153441.51	101145.55	407934.54	0	1	0
27	72107.60	127864.55	353183.81	0	0	1
38	20229.59	65947.93	185265.10	0	0	1
31	61136.38	152701.92	88218.23	0	0	1
22	73994.56	122782.75	303319.26	0	1	0
4	142107.34	91391.77	366168.42	0	1	0

In [11]:
```
y_train
```

Out[11]:

33	96778.92
35	96479.51
26	105733.54
34	96712.80
18	124266.90
7	155752.60
14	132602.65
45	64926.08
48	35673.41
29	101004.64
15	129917.04
30	99937.59
32	97427.84
16	126992.93
42	71498.49
20	118474.03
43	69758.98
8	152211.77
13	134307.35
25	107404.34
5	156991.12
17	125370.37
40	78239.91
49	14681.40
1	191792.06
12	141585.52
37	89949.14
24	108552.04
6	156122.51
23	108733.99
36	90708.19
21	111313.02
19	122776.86
9	149759.96
39	81005.76
46	49490.75
3	182901.99
0	192261.83
47	42559.73

```
44     65200.33
Name: Profit, dtype: float64
```

In [12]:
```
y_test
```

Out[12]:
```
28    103282.38
11    144259.40
10    146121.95
41     77798.83
2     191050.39
27    105008.31
38     81229.06
31     97483.56
22    110352.25
4     166187.94
Name: Profit, dtype: float64
```

In [13]:
```
# Create and train the regressor model

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train, y_train)
```

Out[13]:

▾ LinearRegression

```
LinearRegression()
```

In [14]:
```
y_pred = regressor.predict(x_test)
y_pred
```

Out[14]:
```
array([103015.20159796, 132582.27760816, 132447.73845174,  71976.09851258,
       178537.48221055, 116161.24230165,  67851.69209676,  98791.73374687,
       113969.43533012, 167921.0656955 ])
```

In [15]:
```
m = regressor.coef_
print(m)
```

```
[ 7.73467193e-01  3.28845975e-02  3.66100259e-02  8.66383692e+01
 -8.72645791e+02  7.86007422e+02]
```

In [16]:
```
c = regressor.intercept_
print(c)
```

```
42467.52924855314
```

In [17]:
```
X = np.append(arr = np.ones((50,1)).astype(int), values=X, axis=1)
X
```

Out[17]:
```
array([[1.0000000e+00, 1.6534920e+05, 1.3689780e+05, 4.7178410e+05,
        0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
       [1.0000000e+00, 1.6259770e+05, 1.5137759e+05, 4.4389853e+05,
        1.0000000e+00, 0.0000000e+00, 0.0000000e+00],
       [1.0000000e+00, 1.5344151e+05, 1.0114555e+05, 4.0793454e+05,
        0.0000000e+00, 1.0000000e+00, 0.0000000e+00],
       [1.0000000e+00, 1.4437241e+05, 1.1867185e+05, 3.8319962e+05,
        0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
       [1.0000000e+00, 1.4210734e+05, 9.1391770e+04, 3.6616842e+05,
        0.0000000e+00, 1.0000000e+00, 0.0000000e+00],
       [1.0000000e+00, 1.3187690e+05, 9.9814710e+04, 3.6286136e+05,
        0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
       [1.0000000e+00, 1.3461546e+05, 1.4719887e+05, 1.2771682e+05,
        1.0000000e+00, 0.0000000e+00, 0.0000000e+00],
       [1.0000000e+00, 1.3029813e+05, 1.4553006e+05, 3.2387668e+05,
        0.0000000e+00, 1.0000000e+00, 0.0000000e+00],
       [1.0000000e+00, 1.2054252e+05, 1.4871895e+05, 3.1161329e+05,
        0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
       [1.0000000e+00, 1.2333488e+05, 1.0867917e+05, 3.0498162e+05,
        1.0000000e+00, 0.0000000e+00, 0.0000000e+00],
       [1.0000000e+00, 1.0191308e+05, 1.1059411e+05, 2.2916095e+05,
        0.0000000e+00, 1.0000000e+00, 0.0000000e+00],
       [1.0000000e+00, 1.0067196e+05, 9.1790610e+04, 2.4974455e+05,
        1.0000000e+00, 0.0000000e+00, 0.0000000e+00],
       [1.0000000e+00, 9.3863750e+04, 1.2732038e+05, 2.4983944e+05,
        0.0000000e+00, 1.0000000e+00, 0.0000000e+00],
       [1.0000000e+00, 9.1992390e+04, 1.3549507e+05, 2.5266493e+05,
        1.0000000e+00, 0.0000000e+00, 0.0000000e+00],
       [1.0000000e+00, 1.1994324e+05, 1.5654742e+05, 2.5651292e+05,
        0.0000000e+00, 1.0000000e+00, 0.0000000e+00],
       [1.0000000e+00, 1.1452361e+05, 1.2261684e+05, 2.6177623e+05,
        0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
       [1.0000000e+00, 7.8013110e+04, 1.2159755e+05, 2.6434606e+05,
        1.0000000e+00, 0.0000000e+00, 0.0000000e+00],
       [1.0000000e+00, 9.4657160e+04, 1.4507758e+05, 2.8257431e+05,
        0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
       [1.0000000e+00, 9.1749160e+04, 1.1417579e+05, 2.9491957e+05,
        0.0000000e+00, 1.0000000e+00, 0.0000000e+00],
       [1.0000000e+00, 8.6419700e+04, 1.5351411e+05, 0.0000000e+00,
        0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
```

```
[1.0000000e+00, 7.6253860e+04, 1.1386730e+05, 2.9866447e+05,
 1.0000000e+00, 0.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 7.8389470e+04, 1.5377343e+05, 2.9973729e+05,
 0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
[1.0000000e+00, 7.3994560e+04, 1.2278275e+05, 3.0331926e+05,
 0.0000000e+00, 1.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 6.7532530e+04, 1.0575103e+05, 3.0476873e+05,
 0.0000000e+00, 1.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 7.7044010e+04, 9.9281340e+04, 1.4057481e+05,
 0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
[1.0000000e+00, 6.4664710e+04, 1.3955316e+05, 1.3796262e+05,
 1.0000000e+00, 0.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 7.5328870e+04, 1.4413598e+05, 1.3405007e+05,
 0.0000000e+00, 1.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 7.2107600e+04, 1.2786455e+05, 3.5318381e+05,
 0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
[1.0000000e+00, 6.6051520e+04, 1.8264556e+05, 1.1814820e+05,
 0.0000000e+00, 1.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 6.5605480e+04, 1.5303206e+05, 1.0713838e+05,
 0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
[1.0000000e+00, 6.1994480e+04, 1.1564128e+05, 9.1131240e+04,
 0.0000000e+00, 1.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 6.1136380e+04, 1.5270192e+05, 8.8218230e+04,
 0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
[1.0000000e+00, 6.3408860e+04, 1.2921961e+05, 4.6085250e+04,
 1.0000000e+00, 0.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 5.5493950e+04, 1.0305749e+05, 2.1463481e+05,
 0.0000000e+00, 1.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 4.6426070e+04, 1.5769392e+05, 2.1079767e+05,
 1.0000000e+00, 0.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 4.6014020e+04, 8.5047440e+04, 2.0551764e+05,
 0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
[1.0000000e+00, 2.8663760e+04, 1.2705621e+05, 2.0112682e+05,
 0.0000000e+00, 1.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 4.4069950e+04, 5.1283140e+04, 1.9702942e+05,
 1.0000000e+00, 0.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 2.0229590e+04, 6.5947930e+04, 1.8526510e+05,
 0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
[1.0000000e+00, 3.8558510e+04, 8.2982090e+04, 1.7499930e+05,
 1.0000000e+00, 0.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 2.8754330e+04, 1.1854605e+05, 1.7279567e+05,
```

```
 1.0000000e+00, 0.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 2.7892920e+04, 8.4710770e+04, 1.6447071e+05,
 0.0000000e+00, 1.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 2.3640930e+04, 9.6189630e+04, 1.4800111e+05,
 1.0000000e+00, 0.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 1.5505730e+04, 1.2738230e+05, 3.5534170e+04,
 0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
[1.0000000e+00, 2.2177740e+04, 1.5480614e+05, 2.8334720e+04,
 1.0000000e+00, 0.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 1.0002300e+03, 1.2415304e+05, 1.9039300e+03,
 0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
[1.0000000e+00, 1.3154600e+03, 1.1581621e+05, 2.9711446e+05,
 0.0000000e+00, 1.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 0.0000000e+00, 1.3542692e+05, 0.0000000e+00,
 1.0000000e+00, 0.0000000e+00, 0.0000000e+00],
[1.0000000e+00, 5.4205000e+02, 5.1743150e+04, 0.0000000e+00,
 0.0000000e+00, 0.0000000e+00, 1.0000000e+00],
[1.0000000e+00, 0.0000000e+00, 1.1698380e+05, 4.5173060e+04,
 1.0000000e+00, 0.0000000e+00, 0.0000000e+00]])
```

In [18]:

```python
# --- 1
import statsmodels.api as sm
X_opt  = X[:,[0,1,2,3,4,5]]

# Ordinary Least Sqaures

regressor_OLS = sm.OLS(endog=y , exog=X_opt).fit()
regressor_OLS.summary()

# --- 2
import statsmodels.api as sm
X_opt = X[:,[0,1,2,3,5]]

regressor_OLS =sm.OLS(endog=y,exog=X_opt).fit()
regressor_OLS.summary()

# --- 3
import statsmodels.api as sm
X_opt = X[:,[0,1,2,3]]

regressor_OLS =sm.OLS(endog=y,exog=X_opt).fit()
```

```
regressor_OLS.summary()

# --- 4
import statsmodels.api as sm
X_opt = X[:,[0,1,3]]

regressor_OLS =sm.OLS(endog=y,exog=X_opt).fit()
regressor_OLS.summary()

# --- 5
import statsmodels.api as sm
X_opt = X[:,[0,1]]

regressor_OLS =sm.OLS(endog=y,exog=X_opt).fit()
regressor_OLS.summary()
```

Out[18]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.947
Model:	OLS	**Adj. R-squared:**	0.945
Method:	Least Squares	**F-statistic:**	849.8
Date:	Sat, 25 Jan 2025	**Prob (F-statistic):**	3.50e-32
Time:	23:41:53	**Log-Likelihood:**	-527.44
No. Observations:	50	**AIC:**	1059.
Df Residuals:	48	**BIC:**	1063.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	4.903e+04	2537.897	19.320	0.000	4.39e+04	5.41e+04
x1	0.8543	0.029	29.151	0.000	0.795	0.913

Omnibus:	13.727	**Durbin-Watson:**	1.116
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	18.536
Skew:	-0.911	**Prob(JB):**	9.44e-05
Kurtosis:	5.361	**Cond. No.**	1.65e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.65e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

In [19]:
```
bias = regressor.score(x_train,y_train)
bias
```

Out[19]:
```
0.9501847627493607
```

In [20]:
```
variance = regressor.score(x_test,y_test)
variance
```

Out[20]:
```
0.9347068473282424
```