Multiple Linear Regression

0.Data Preprocessing

0.1 Importing the libraries

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

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
path= '/content/drive/My Drive/50_Startups.csv/'
```

0.2 Importing the dataset

```
dataset = pd.read_csv('/content/drive/My Drive/50_Startups.csv')
dataset
```

10	101913.08	110594.11	229160.95	Florida	146121.95
11	100671.96	91790.61	249744.55	California	144259.40
12	93863.75	127320.38	249839.44	Florida	141585.52
13	91992.39	135495.07	252664.93	California	134307.35
14	119943.24	156547.42	256512.92	Florida	132602.65
15	114523.61	122616.84	261776.23	New York	129917.04
16	78013.11	121597.55	264346.06	California	126992.93
17	94657.16	145077.58	282574.31	New York	125370.37
18	91749.16	114175.79	294919.57	Florida	124266.90
19	86419.70	153514.11	0.00	New York	122776.86
20	76253.86	113867.30	298664.47	California	118474.03
21	78389.47	153773.43	299737.29	New York	111313.02
22	73994.56	122782.75	303319.26	Florida	110352.25
23	67532.53	105751.03	304768.73	Florida	108733.99
24	77044.01	99281.34	140574.81	New York	108552.04
25	64664.71	139553.16	137962.62	California	107404.34
26	75328.87	144135.98	134050.07	Florida	105733.54
27	72107.60	127864.55	353183.81	New York	105008.31
28	66051.52	182645.56	118148.20	Florida	103282.38
29	65605.48	153032.06	107138.38	New York	101004.64
30	61994.48	115641.28	91131.24	Florida	99937.59
31	61136.38	152701.92	88218.23	New York	97483.56
32	63408.86	129219.61	46085.25	California	97427.84
33	55493.95	103057.49	214634.81	Florida	96778.92
34	46426.07	157693.92	210797.67	California	96712.80

0.3 Check if any null value

```
dataset.isna().sum()
```

```
R&D Spend          0
Administration     0
Marketing Spend    0
State              0
Profit             0
dtype: int64
```

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   R&D Spend        50 non-null     float64
 1   Administration   50 non-null     float64
 2   Marketing Spend  50 non-null     float64
 3   State            50 non-null     object 
 4   Profit           50 non-null     float64
dtypes: float64(4), object(1)
memory usage: 2.1+ KB
```

```
### 0.4 Split into X & y
```

```
X = dataset.drop('Profit', axis=1)
X
```

```
10  101913.08  110594.11  229160.95     Florida
11  100671.96   91790.61  249744.55  California
12   93863.75  127320.38  249839.44     Florida
13   91992.39  135495.07  252664.93  California
14  119943.24  156547.42  256512.92     Florida
15  114523.61  122616.84  261776.23    New York
16   78013.11  121597.55  264346.06  California
17   94657.16  145077.58  282574.31    New York
18   91749.16  114175.79  294919.57     Florida
19   86419.70  153514.11       0.00    New York
20   76253.86  113867.30  298664.47  California
21   78389.47  153773.43  299737.29    New York
22   73994.56  122782.75  303319.26     Florida
23   67532.53  105751.03  304768.73     Florida
24   77044.01   99281.34  140574.81    New York
25   64664.71  139553.16  137962.62  California
26   75328.87  144135.98  134050.07     Florida
27   72107.60  127864.55  353183.81    New York
28   66051.52  182645.56  118148.20     Florida
29   65605.48  153032.06  107138.38    New York
30   61994.48  115641.28   91131.24     Florida
31   61136.38  152701.92   88218.23    New York
32   63408.86  129219.61   46085.25  California
33   55493.95  103057.49  214634.81     Florida
34   46426.07  157693.92  210797.67  California
35   46014.02   85047.44  205517.64    New York
36   28663.76  127056.21  201126.82     Florida
37   44069.95   51283.14  197029.42  California
38   20229.59   65947.93  185265.10    New York
39   38558.51   82982.09  174999.30  California
40   28754.33  118546.05  172795.67  California
41   27892.92   84710.77  164470.71     Florida
42   23640.93   96189.63  148001.11  California
43   15505.73  127382.30   35534.17    New York
44   22177.74  154806.14   28334.72  California
45    1000.23  124153.04    1903.93    New York
46    1315.46  115816.21  297114.46     Florida
47       0.00  135426.92       0.00  California
48     542.05   51743.15       0.00    New York
49       0.00  116983.80   45173.06  California
```

```
y = dataset['Profit']
y
```

```
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
```

0.5 Encoding categorical data

```
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer

categorical_feature = ["State"]
one_hot = OneHotEncoder()
transformer = ColumnTransformer([("one_hot",
                                  one_hot,
                                  categorical_feature)],
                                 remainder="passthrough")

transformed_X = transformer.fit_transform(X)
```

```
pd.DataFrame(transformed_X).head()
```

	0	1	2	3	4	5
0	0.0	0.0	1.0	165349.20	136897.80	471784.10
1	1.0	0.0	0.0	162597.70	151377.59	443898.53
3	0.0	0.0	1.0	144372.41	118671.85	383199.62

0.6 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(transformed_X, y, test_size = 0.25, random_state = 2509)
```

1. Training the Multiple Linear Regression model on the Training set

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

```
LinearRegression()
```

1.1 Score

```
regressor.score(X_test,y_test)
```

```
0.9840064291741644
```

2. Predicting the Test set results

```
y_pred = regressor.predict(X_test)
```

```
d = {'y_pred': y_pred, 'y_test': y_test}
```

2.1 Compare Predicted results

```
pd.DataFrame(d)
```

	y_pred	y_test
32	98884.371543	97427.84
33	100047.235184	96778.92
47	47766.247901	42559.73
9	154976.558305	149759.96
37	91129.087779	89949.14
8	151755.926389	152211.77
23	112436.195860	108733.99
24	113375.898676	108552.04
17	130706.106786	125370.37
1	189141.730655	191792.06
39	85217.422839	81005.76
22	116952.737156	110352.25
46	60343.602070	49490.75