ASSIGNMENT / TASK 8

Task- Predicting a Startups Profit/Success Rate using Multiple Linear Regression in Python.

Here 50 startups dataset containing 5 columns like “R&D Spend”, “Administration”, “Marketing Spend”, “State”, “Profit”.

In this dataset first 3 columns provides you spending on Research , Administration and Marketing respectively. State indicates startup based on that state. Profit indicates how much profits earned by a startup.

Clearly, we can understand that it is a multiple linear regression problem, as the independent variables are more than one.

Prepare a prediction model for profit of 50_Startups data in Python

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn
from sklearn.linear_model import LinearRegression
```

```
df=pd.read_csv('/content/50_Startups.csv')
df.head()
```

	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

```
df.keys()
```

```
Index(['R&D Spend', 'Administration', 'Marketing Spend', 'State', 'Profit'], dtype='obje
```

```
dummies=pd.get_dummies(df.State)
dummies
```

	California	Florida	New York
0	0	0	1
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0
5	0	0	1
6	1	0	0
7	0	1	0
8	0	0	1
9	1	0	0
10	0	1	0
11	1	0	0
12	0	1	0
13	1	0	0
14	0	1	0
15	0	0	1
16	1	0	0
17	0	0	1
18	0	1	0
19	0	0	1
20	1	0	0
21	0	0	1
22	0	1	0
23	0	1	0
24	0	0	1
25	1	0	0
26	0	1	0
27	0	0	1
28	0	1	0
29	0	0	1

30	0	1	0
31	0	0	1
32	1	0	0
33	0	1	0
34	1	0	0
35	0	0	1
36	0	1	0
37	1	0	0
38	0	0	1
39	1	0	0
40	1	0	0
41	0	1	0
42	1	0	0
43	0	0	1
44	1	0	0
45	0	0	1
46	0	1	0
47	1	0	0
48	0	0	1
49	1	0	0

```
merge=pd.concat([df,dummies],axis='columns')
merge.head()
```

	R&D Spend	Administration	Marketing Spend	State	Profit	California	Florida
0	165349.20	136897.80	471784.10	New York	192261.83	0	0
1	162597.70	151377.59	443898.53	California	191792.06	1	0
2	153441.51	101145.55	407934.54	Florida	191050.39	0	1
3	144372.41	118671.85	383199.62	New York	182901.99	0	0
4	142107.34	91391.77	366168.42	Florida	166187.94	0	1

```
new_df=merge.drop(columns='State')
new_df.head()
```

	R&D Spend	Administration	Marketing Spend	Profit	California	Florida	New York
0	165349.20	136897.80	471784.10	192261.83	0	0	1
1	162597.70	151377.59	443898.53	191792.06	1	0	0
2	153441.51	101145.55	407934.54	191050.39	0	1	0
3	144372.41	118671.85	383199.62	182901.99	0	0	1

```
new_df=new_df.drop(["New York"],axis="columns")
new_df.head()
```

	R&D Spend	Administration	Marketing Spend	Profit	California	Florida
0	165349.20	136897.80	471784.10	192261.83	0	0
1	162597.70	151377.59	443898.53	191792.06	1	0
2	153441.51	101145.55	407934.54	191050.39	0	1
3	144372.41	118671.85	383199.62	182901.99	0	0
4	142107.34	91391.77	366168.42	166187.94	0	1

```
y=new_df['Profit']
print("Values of y","\n",y,"\n")
```

```
Values of 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
```

```
x=new_df.loc[:,["R&D Spend",  "Administration", "Marketing Spend", "California" ,"Florida"]]
print("Values of x \n",x)
```

```
Values of x
     R&D Spend  Administration  Marketing Spend  California  Florida
0   165349.20       136897.80        471784.10           0        0
1   162597.70       151377.59        443898.53           1        0
2   153441.51       101145.55        407934.54           0        1
3   144372.41       118671.85        383199.62           0        0
4   142107.34        91391.77        366168.42           0        1
5   131876.90        99814.71        362861.36           0        0
6   134615.46       147198.87        127716.82           1        0
7   130298.13       145530.06        323876.68           0        1
8   120542.52       148718.95        311613.29           0        0
9   123334.88       108679.17        304981.62           1        0
10  101913.08       110594.11        229160.95           0        1
11  100671.96        91790.61        249744.55           1        0
12   93863.75       127320.38        249839.44           0        1
13   91992.39       135495.07        252664.93           1        0
14  119943.24       156547.42        256512.92           0        1
15  114523.61       122616.84        261776.23           0        0
16   78013.11       121597.55        264346.06           1        0
17   94657.16       145077.58        282574.31           0        0
18   91749.16       114175.79        294919.57           0        1
19   86419.70       153514.11             0.00           0        0
20   76253.86       113867.30        298664.47           1        0
21   78389.47       153773.43        299737.29           0        0
22   73994.56       122782.75        303319.26           0        1
23   67532.53       105751.03        304768.73           0        1
```

24	77044.01	99281.34	140574.81	0	0
25	64664.71	139553.16	137962.62	1	0
26	75328.87	144135.98	134050.07	0	1
27	72107.60	127864.55	353183.81	0	0
28	66051.52	182645.56	118148.20	0	1
29	65605.48	153032.06	107138.38	0	0
30	61994.48	115641.28	91131.24	0	1
31	61136.38	152701.92	88218.23	0	0
32	63408.86	129219.61	46085.25	1	0
33	55493.95	103057.49	214634.81	0	1
34	46426.07	157693.92	210797.67	1	0
35	46014.02	85047.44	205517.64	0	0
36	28663.76	127056.21	201126.82	0	1
37	44069.95	51283.14	197029.42	1	0
38	20229.59	65947.93	185265.10	0	0
39	38558.51	82982.09	174999.30	1	0
40	28754.33	118546.05	172795.67	1	0
41	27892.92	84710.77	164470.71	0	1
42	23640.93	96189.63	148001.11	1	0
43	15505.73	127382.30	35534.17	0	0
44	22177.74	154806.14	28334.72	1	0
45	1000.23	124153.04	1903.93	0	0
46	1315.46	115816.21	297114.46	0	1
47	0.00	135426.92	0.00	1	0
48	542.05	51743.15	0.00	0	0
49	0.00	116983.80	45173.06	1	0

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=5)
model=LinearRegression()
model.fit(x_train,y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
y_pred_train=model.predict(x_train)
y_pred_test=model.predict(x_test)
data=pd.DataFrame(y_pred_test,y_test)
data.head()
```

	0
Profit	
71498.49	72026.062615
101004.64	100303.969440
156122.51	156099.674725
122776.86	113558.712493
103282.38	98681.808832

```
from sklearn.metrics import r2_score
```

```
score=r2_score(y_test,y_pred_test)
score
```

```
0.9760590128066435
```

✓ 0s completed at 10:57