Task-8

GO_STP_5247

Multiple Linear Regression

Predicting a Startups Profit/Success Rate using Multiple Linear Regression in Python-Download Data Set

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

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

```
[1]  import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import sklearn

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

✓ 0s completed at 11:15 PM

```
[1] from sklearn.linear_model import LinearRegression
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import r2_score, mean_squared_error

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

```
[2] print(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
```

```
[3] print(df.tail())
```

```
    R&D Spend  Administration  Marketing Spend       State    Profit
45    1000.23       124153.04          1903.93    New York  64926.08
46    1315.46       115816.21        297114.46     Florida  49490.75
47       0.00       135426.92             0.00  California  42559.73
48     542.05        51743.15             0.00    New York  35673.41
49       0.00       116983.80         45173.06  California  14681.40
```

```
[4] print(df.shape)
```

```
(50, 5)
```

```
[5] print(df.dtypes)
```

```
R&D Spend          float64
Administration     float64
Marketing Spend    float64
State               object
```

```
Profit            float64
dtype: object
```

```
[6] print(df.columns)
```

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

```
[7] df.nunique()
```

```
R&D Spend          49
Administration     50
Marketing Spend    48
State               3
Profit             50
dtype: int64
```

```
[8] df.isnull().sum()
```

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

```
[9] enc_df = pd.get_dummies(df, columns=["State"]) # get dummy values for State and converted it to binary values
```

```
[10] enc_df=enc_df.drop(['State_New York'],axis=1)
     # removed one of the encoded columns as it can be deduced on the basis of the values in the other 2 columns
     # if both State_California and State_Florida =0 means it is State_New York
```

```
[11] enc_df.head()
```

[11]

	R&D Spend	Administration	Marketing Spend	Profit	State_California	State_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

[12]
```
print(enc_df.corr())
```

```
                 R&D Spend  Administration  ...  State_California  State_Florida
R&D Spend         1.000000        0.241955  ...         -0.143165       0.105711
Administration    0.241955        1.000000  ...         -0.015478       0.010493
Marketing Spend   0.724248       -0.032154  ...         -0.168875       0.205685
Profit            0.972900        0.200717  ...         -0.145837       0.116244
State_California -0.143165       -0.015478  ...          1.000000      -0.492366
State_Florida     0.105711        0.010493  ...         -0.492366       1.000000

[6 rows x 6 columns]
```

[13]
```
enc_df.describe()
```

	R&D Spend	Administration	Marketing Spend	Profit	State_California	State_Florida
count	50.000000	50.000000	50.000000	50.000000	50.000000	50.000000
mean	73721.615600	121344.639600	211025.097800	112012.639200	0.340000	0.320000
std	45902.256482	28017.802755	122290.310726	40306.180338	0.478518	0.471212
min	0.000000	51283.140000	0.000000	14681.400000	0.000000	0.000000
25%	39936.370000	103730.875000	129300.132500	90138.902500	0.000000	0.000000

50%	73051.080000	122699.795000	212716.240000	107978.190000	0.000000	0.000000
75%	101602.800000	144842.180000	299469.085000	139765.977500	1.000000	1.000000
max	165349.200000	182645.560000	471784.100000	192261.830000	1.000000	1.000000

```
[14] type(enc_df)
```

```
pandas.core.frame.DataFrame
```

```
[15] y=enc_df['Profit']
```

```
[16] x=enc_df.drop(['Profit'],axis=1)
```

```
[17] print(x.head())
```

```
   R&D Spend  Administration  Marketing Spend  State_California  State_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
```

```
[18] print(y.head())
```

```
0    192261.83
1    191792.06
2    191050.39
3    182901.99
4    166187.94
Name: Profit, dtype: float64
```

```
[19] print(x.shape,y.shape)
```

```
(50, 5) (50,)
```

```
[20] x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=.25,random_state=5)
```

```
[21] print(x_train.shape)
```

```
(37, 5)
```

```
[22] print(x_test.shape)
```

```
(13, 5)
```

```
[23] print(y_train.shape)
```

```
(37,)
```

```
[24] print(y_test.shape)
```

```
(13,)
```

```
[25] mvr=LinearRegression()
```

```
[26] mvr.fit(x_train,y_train)
```

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

```
y_pred = mvr.predict(x_test)
print(y_pred)
```

```
y_pred = mvr.predict(x_test)
print(y_pred)
print(y_test)
print(x_test)
```

```
[ 71165.92121164  98991.88050568 155658.78699154 111109.64339249
 100180.12553441 128692.30008051 181975.81887461  57805.09379185
 172178.68837146 116394.29722332  94778.51265188 171783.69240153
  97372.02912334]
42     71498.49
29    101004.64
6     156122.51
19    122776.86
28    103282.38
17    125370.37
2     191050.39
43     69758.98
3     182901.99
21    111313.02
31     97483.56
4     166187.94
32     97427.84
Name: Profit, dtype: float64
    R&D Spend  Administration  Marketing Spend  State_California  State_Florida
42   23640.93        96189.63        148001.11                 1              0
29   65605.48       153032.06        107138.38                 0              0
6   134615.46       147198.87        127716.82                 1              0
19   86419.70       153514.11             0.00                 0              0
28   66051.52       182645.56        118148.20                 0              1
17   94657.16       145077.58        282574.31                 0              0
2   153441.51       101145.55        407934.54                 0              1
43   15505.73       127382.30         35534.17                 0              0
3   144372.41       118671.85        383199.62                 0              0
21   78389.47       153773.43        299737.29                 0              0
31   61136.38       152701.92         88218.23                 0              0
4   142107.34        91391.77        366168.42                 0              1
32   63408.86       129219.61         46085.25                 1              0
```

```
[28] mvr.coef_
```

```
array([ 7.81494897e-01, -2.84329206e-02,  3.85924982e-02,  1.77593125e+03,
        1.25676888e+03])
```

```
[29] mvr.intercept_
```

```
47937.94335938635
```

```
[30] print("Accuracy is: ", r2_score(y_test,y_pred))
```

```
Accuracy is:  0.9701993496424656
```

```
[31] print("MSE: ",mean_squared_error(y_test,y_pred))
```

```
MSE:  43526510.84446017
```