Profit Prediction of companies Linear Regression

In [2]:
```python
#importing libraries
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
%matplotlib inline
```

loading the dataset

In [3]:
```python
companies = pd.read_csv('1000_Companies.csv')
companies.head()
```

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

In [4]:
```python
#Research and Development (R&D)
```

Observing Correlation between Columns

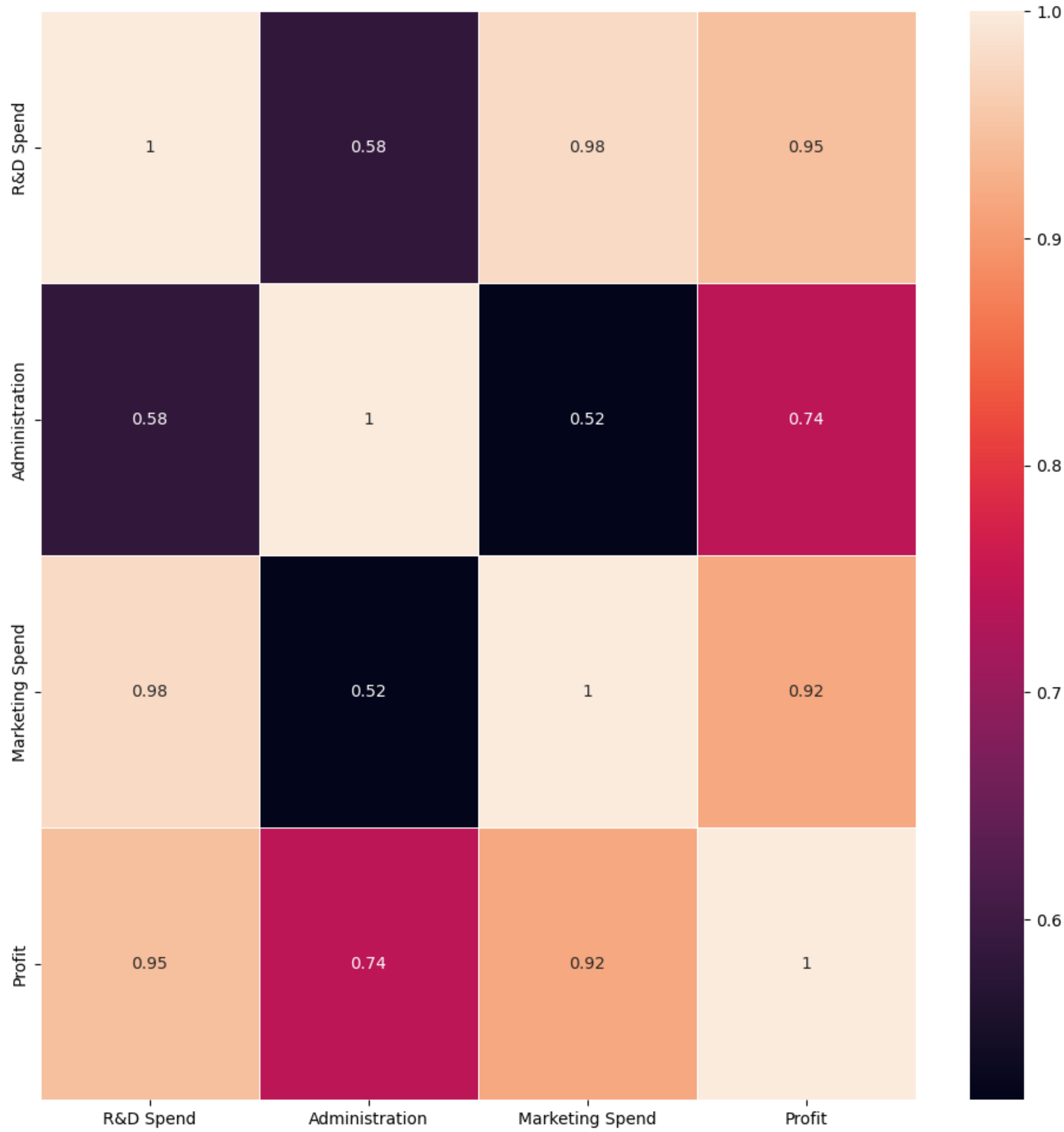
In [5]:
```python
companies[['R&D Spend','Administration','Marketing Spend']].corr()
```

Out[5]:

	R&D Spend	Administration	Marketing Spend
R&D Spend	1.000000	0.582434	0.978407
Administration	0.582434	1.000000	0.520465
Marketing Spend	0.978407	0.520465	1.000000

In [6]:
```
plt.figure(figsize=(12,12))
sns.heatmap(companies.corr(),annot = True, linewidths=.5)
```

Out[6]: <AxesSubplot:>



removing column due to high correlation

In [7]:
```
companies.drop(['Marketing Spend'],axis=1,inplace=True)
```

In [8]:
```
companies
```

Out[8]:

	R&D Spend	Administration	State	Profit
0	165349.20	136897.800	New York	192261.83000
1	162597.70	151377.590	California	191792.06000
2	153441.51	101145.550	Florida	191050.39000
3	144372.41	118671.850	New York	182901.99000
4	142107.34	91391.770	Florida	166187.94000
...	...	...	...	...
995	54135.00	118451.999	California	95279.96251
996	134970.00	130390.080	California	164336.60550
997	100275.47	241926.310	California	413956.48000
998	128456.23	321652.140	California	333962.19000
999	161181.72	270939.860	New York	476485.43000

1000 rows × 4 columns

Encoding Data

In [9]:
```
dummies=pd.get_dummies(companies.State,drop_first=True)
```

In [10]:
```
companies=pd.concat([companies,dummies],axis=1)
```

In [11]:
```
companies
```

Out[11]:

	R&D Spend	Administration	State	Profit	Florida	New York
0	165349.20	136897.800	New York	192261.83000	0	1
1	162597.70	151377.590	California	191792.06000	0	0
2	153441.51	101145.550	Florida	191050.39000	1	0
3	144372.41	118671.850	New York	182901.99000	0	1
4	142107.34	91391.770	Florida	166187.94000	1	0
...	...	...	...	...	...	...
995	54135.00	118451.999	California	95279.96251	0	0
996	134970.00	130390.080	California	164336.60550	0	0
997	100275.47	241926.310	California	413956.48000	0	0
998	128456.23	321652.140	California	333962.19000	0	0
999	161181.72	270939.860	New York	476485.43000	0	1

1000 rows × 6 columns

In [12]:
```
companies.drop(['State'],axis=1,inplace=True)
companies
```

Out[12]:

	R&D Spend	Administration	Profit	Florida	New York
0	165349.20	136897.800	192261.83000	0	1
1	162597.70	151377.590	191792.06000	0	0
2	153441.51	101145.550	191050.39000	1	0
3	144372.41	118671.850	182901.99000	0	1
4	142107.34	91391.770	166187.94000	1	0
...	...	...	...	...	...
995	54135.00	118451.999	95279.96251	0	0
996	134970.00	130390.080	164336.60550	0	0
997	100275.47	241926.310	413956.48000	0	0
998	128456.23	321652.140	333962.19000	0	0
999	161181.72	270939.860	476485.43000	0	1

1000 rows × 5 columns

Scaling

In [13]:
```
from sklearn.preprocessing import MinMaxScaler
scale=MinMaxScaler()
companies[['R&D Spend','Administration']]=scale.fit_transform(companies[['
```

In [14]:
```
companies
```

Out[14]:

	R&D Spend	Administration	Profit	Florida	New York
0	1.000000	0.316659	192261.83000	0	1
1	0.983359	0.370214	191792.06000	0	0
2	0.927985	0.184424	191050.39000	1	0
3	0.873136	0.249247	182901.99000	0	1
4	0.859438	0.148348	166187.94000	1	0
...	...	...	...	...	...
995	0.327398	0.248434	95279.96251	0	0
996	0.816272	0.292589	164336.60550	0	0
997	0.606447	0.705122	413956.48000	0	0
998	0.776878	1.000000	333962.19000	0	0
999	0.974796	0.812433	476485.43000	0	1

1000 rows × 5 columns

Specifying X and y

In [15]:

```
y=companies.iloc[:,2].values
y

```

```
       65743.69265,  96381.04826,  70133.5447 ,  93606.40571,
      151891.2888 ,  79299.58812,  83553.10486,  60650.40749,
      123228.959  ,  52481.67341, 161467.0408 , 107682.5646 ,
      140522.3792 , 102118.5649 ,  57143.54142, 159227.0888 ,
      163673.6754 , 157493.7316 ,  91623.59544, 157949.9232 ,
       94974.98049, 148975.5923 , 158516.3184 , 110682.8359 ,
      146690.3629 ,  58605.23395, 120412.3603 , 161783.1286 ,
       76487.26093,  95178.30183, 104231.2275 ,  58963.18204,
       76017.40068,  59803.80475, 129642.9786 ,  51003.74933,
       77362.05529, 185502.5285 , 180753.5228 , 172495.0881 ,
      140251.5689 ,  63093.68082, 171416.9724 , 111814.772  ,
      123671.4819 ,  92903.32391, 105457.9899 ,  74425.00156,
      173861.9543 ,  62223.15791,  60869.96038, 110395.794  ,
      161076.6296 , 107704.7762 , 141344.2075 , 168760.9805 ,
       97599.36358,  89558.7732 ,  99322.46927,  60065.21791,
      102489.3274 ,  94400.89669, 154569.4922 ,  90808.60147,
      138855.6568 , 103378.6447 , 134808.0242 ,  84305.73556,
       83178.92524,  86221.9111 , 165330.1463 , 161035.6236 ,
      138841.9881 ,  89012.02672, 132077.709  ,  95279.96251,
      164336.6055 , 413956.48   , 333962.19   , 476485.43   ])
```

In [16]:

```
companies.drop(['Profit'],axis=1,inplace=True)
companies

```

Out[16]:

	R&D Spend	Administration	Florida	New York
0	1.000000	0.316659	0	1
1	0.983359	0.370214	0	0
2	0.927985	0.184424	1	0
3	0.873136	0.249247	0	1
4	0.859438	0.148348	1	0
...	...	...	...	...
995	0.327398	0.248434	0	0
996	0.816272	0.292589	0	0
997	0.606447	0.705122	0	0
998	0.776878	1.000000	0	0
999	0.974796	0.812433	0	1

1000 rows × 4 columns

In [17]:
```
X=companies.iloc[:,:].values
X.shape
```

Out[17]:
```
(1000, 4)
```

splitting dataset into train and test

In [18]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.25,random

```

fitting model to training set

In [24]:
```
from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(X_train,y_train)
```

Out[24]:
```
LinearRegression()
```

predicting the test dataset

In [29]:

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

Out[29]:
```
array([ 89500.55415283,  88138.8446888 ,  94597.75216055, 175653.71089275,
        83382.18053249, 110505.82130012, 116493.19253044,  91181.05268779,
       164458.30850173,  53360.37811062,  66690.85003053, 150573.06276522,
       126575.21430958,  59088.75792558, 177105.8602847 ,  75297.62004058,
       118298.23553328, 164183.76056061, 175819.41530767, 181906.87097512,
       129596.30184748,  85385.4322966 , 180958.29325073,  83862.61435084,
       104695.31182842, 100928.11561916,  61332.5861039 ,  56887.54851565,
        68634.43531861, 203117.71769303, 120825.4105066 , 111405.30851453,
       101434.04281794, 137982.60653888,  63714.62794184, 108542.20517781,
       185847.17294968, 171421.18541889, 174618.50317275, 117722.35835103,
        96683.91356914, 165060.43063323, 107410.16102518,  50336.16629218,
       116566.31937606,  58384.33681526, 158412.53276962,  78644.9826959 ,
       159721.96763826, 131170.63158137, 184841.17896123, 174583.36786983,
        93450.28274676,  78318.40396649, 180585.43865278,  84557.14973158,
       142917.8391768 , 170471.59683568,  84334.26313478, 105248.37178381,
       141678.67499905,  52539.33345548, 141482.89032463, 139071.7257301 ,
        97993.0276453 , 113643.2288559 , 126666.37280783, 152013.53549595,
        58867.9830529 , 174260.29393326, 124287.13118198, 168669.0891951 ,
        91365.69688163, 156042.96140013,  84024.57980569,  77855.64562128,
       120461.20254026,  93337.48478727, 138664.3489102 , 143242.22919672,
       171127.24814345, 139725.26539881, 106309.28818736, 155262.80705272,
       139924.40089532, 110107.33491894,  69409.98409295,  88112.59973955,
       139775.82604795, 148362.02895806, 157786.87501105,  58284.92946325,
        93663.9865313 , 112704.00068045,  56562.76063282, 107353.06546855,
       147363.74755766, 152022.10514219, 167665.05278989, 118421.24001484,
       120809.98524544, 139052.09258867, 157398.67213732, 122007.15833523,
        87056.82500466, 104861.10278352,  95289.63963652, 178232.22650296,
       181528.63081635, 109766.26480182, 164862.47290234, 167411.07186366,
       157872.5709631 , 174630.42364108, 169798.56242833,  52062.54291783,
       176530.84016064, 104441.65160961,  82429.77722279, 138786.35801584,
       144289.83206322, 161670.69468006, 169642.99347819, 120498.05178944,
       158876.14808797, 109896.20196873, 169528.94076771,  60509.59759993,
       158653.79775688, 157787.4111917 , 174359.6242999 , 156167.22063268,
       103405.58700153,  85603.42086172, 141243.79847892, 165777.16998974,
       121228.18175343, 169654.38266736, 100579.73069184,  82196.22490865,
       178233.61965674, 101484.2826746 ,  70495.21619638,  90211.29478249,
        61247.76094445,  68902.12755974,  72887.06852701, 176506.84525324,
        89793.17525987, 123529.59951589,  92757.39989486,  98644.27708714,
       172434.57137303,  60642.42639366, 168997.70257818, 166057.07507274,
       165412.18195059, 102242.23304624, 181432.97206428,  73812.5853025 ,
        90975.38228378, 135439.60852655,  64918.12055097,  71761.80308899,
        60605.04096384, 184249.87651782, 176258.72465107, 158776.66375058,
       141251.83194451, 154485.86512459,  58553.47867751,  90705.83769359,
       152762.51852856, 168380.69136617,  72470.50897991, 116117.27234575,
        80189.22462966, 149710.02702213, 116732.49262613, 129847.08753491,
       174508.81231329, 303337.1082258 , 145904.80388315, 150526.7099113 ,
        86200.7989323 ,  70082.23842748,  70211.63941375,  68771.54889292,
       120587.09864863,  89639.70246589, 166809.0268026 , 125176.65559637,
        67201.91898303, 136345.46932987, 118215.56961084, 165505.19288295,
       168637.77939244, 146642.64643519, 141288.68119368, 109022.42352293,
        82290.94970221, 156538.36128437, 136225.44080833,  60463.32173139,
       143138.8578566 , 176586.5425635 ,  87257.35365494, 136276.00145747,
       169642.13650506,  95036.8364759 , 136923.00621867, 126217.32577751,
        31292.94904833,  71168.00679663, 166700.97281897, 138985.24970524,
       127155.69697984, 137810.35757659,  77306.2013945 , 130506.08983488,
       169984.92048338, 108946.23109056, 123767.35418044, 117420.63150216,
        60379.33964054, 126455.5606501 , 179321.10188708,  53997.95633751,
```

```
       119158.47580472, 173187.37999096,  69979.32628264, 152176.4349093 ,
       134453.5683019 ,  60163.38572909,  85820.30873167,  84812.44682622,
        98469.81818296,  76232.56335038, 128063.75384049, 110337.85711842,
        59738.33360618, 101645.39115632, 147543.31141521, 158864.6099146 ,
       161303.83880896, 185322.71347476,  55771.78659731, 153192.89868631,
        67257.62138589, 163875.25499714])
```

finding coefficients and intercepts

In [26]:

```
# coefficient
print(lin_reg.coef_)
```

```
[120926.28381048 229977.19020541   -500.92384786   -445.29843037]
```

In [27]:

```
# intercepts
print(lin_reg.intercept_)
```

```
-937.1148480373813
```

evaluating the model

In [23]:

```
# calculating the  R squared error
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

Out[23]:

```
0.9227500022763352
```

In [ ]:

```

```