Profit estmation of companies with Mulitple Linear Regression Model

In [12]:
```
#importing necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [13]:
```
df = pd.read_csv('data/1000_Companies.csv')

df.head()
```

Out[13]:

	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 [14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   R&D Spend        1000 non-null   float64
 1   Administration   1000 non-null   float64
 2   Marketing Spend  1000 non-null   float64
 3   State            1000 non-null   object 
 4   Profit           1000 non-null   float64
dtypes: float64(4), object(1)
memory usage: 39.2+ KB
```

In [15]:
```
#seprating features and target

X = df.drop('Profit', axis=1).values
y = df['Profit'].values
```

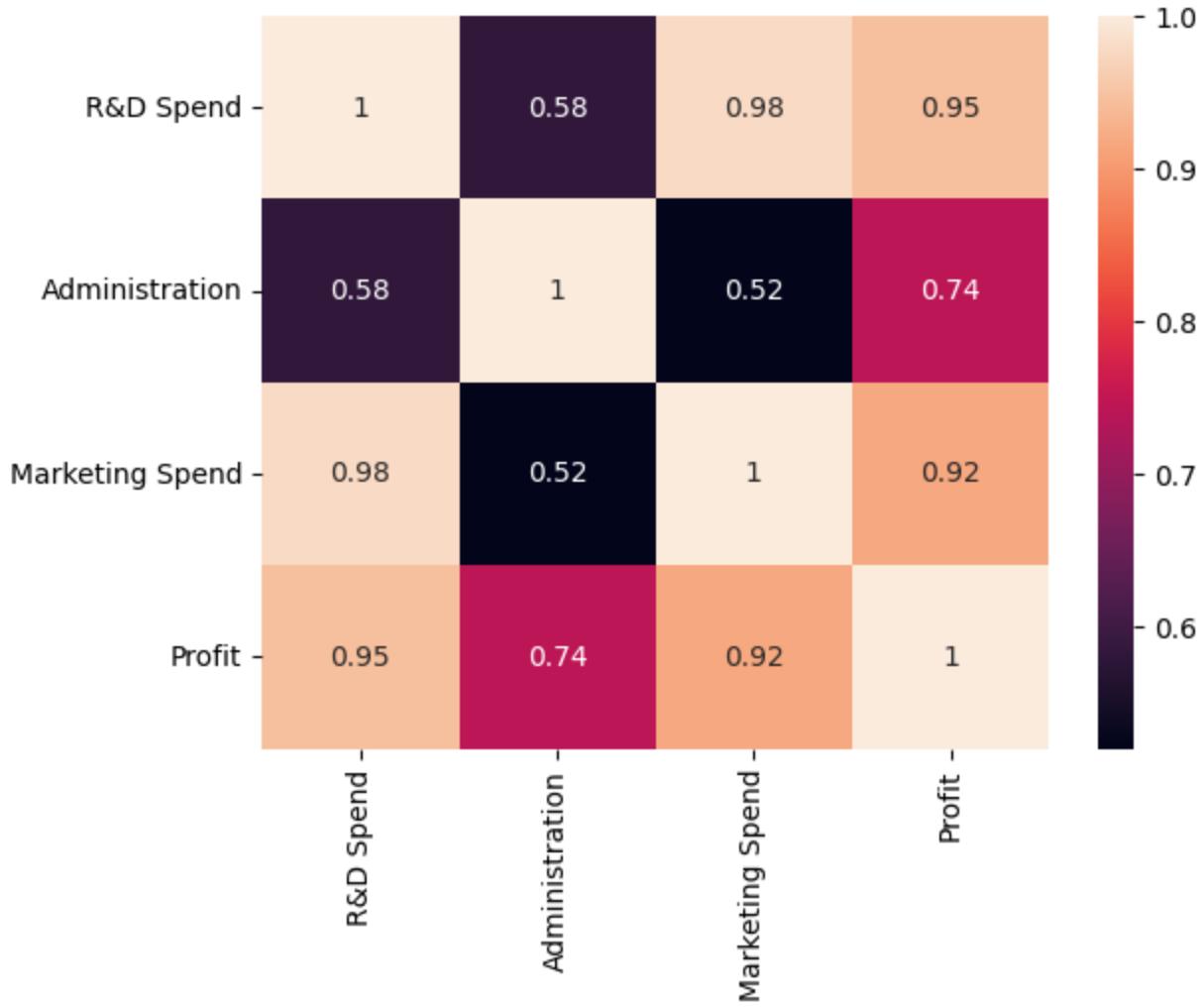
In [16]:
```
X
```

Out[16]:
```
array([[165349.2, 136897.8, 471784.1, 'New York'],
       [162597.7, 151377.59, 443898.53, 'California'],
       [153441.51, 101145.55, 407934.54, 'Florida'],
       ...,
       [100275.47, 241926.31, 227142.82, 'California'],
       [128456.23, 321652.14, 281692.32, 'California'],
       [161181.72, 270939.86, 295442.17, 'New York']], dtype=object)
```

In [17]:
```
#plotting heatmap to anayze correlation

sns.heatmap(df.corr(numeric_only=True), annot=True)
```

Out[17]:
```
<AxesSubplot: >
```



Data preprocessing

In [19]:
```
#encoding categorical data
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.compose import ColumnTransformer

le = LabelEncoder()
X[:, 3] = le.fit_transform(X[:, 3])
ct = ColumnTransformer([('State', OneHotEncoder(), [3])], remainder='passthrough')
X = ct.fit_transform(X)
X
```

Out[19]:
```
array([[0.0, 0.0, 1.0, 165349.2, 136897.8, 471784.1],
       [1.0, 0.0, 0.0, 162597.7, 151377.59, 443898.53],
       [0.0, 1.0, 0.0, 153441.51, 101145.55, 407934.54],
       ...,
       [1.0, 0.0, 0.0, 100275.47, 241926.31, 227142.82],
       [1.0, 0.0, 0.0, 128456.23, 321652.14, 281692.32],
       [0.0, 0.0, 1.0, 161181.72, 270939.86, 295442.17]], dtype=object)
```

In [20]:
```
#avoid dummy variable trap
X = X[:, 1:]
X
```

Out[20]:
```
array([[0.0, 1.0, 165349.2, 136897.8, 471784.1],
       [0.0, 0.0, 162597.7, 151377.59, 443898.53],
       [1.0, 0.0, 153441.51, 101145.55, 407934.54],
       ...,
       [0.0, 0.0, 100275.47, 241926.31, 227142.82],
       [0.0, 0.0, 128456.23, 321652.14, 281692.32],
       [0.0, 1.0, 161181.72, 270939.86, 295442.17]], dtype=object)
```

In [22]:
```
#splitting the dataset into train and test set
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)
```

Model Training

In [23]:
```
#fitting data into machine learning model
from sklearn.linear_model import LinearRegression

lr = LinearRegression()
lr.fit(X_train, y_train)
```

Out[23]:
```
▾ LinearRegression

LinearRegression()
```

Model Evaluation

In [24]:
```
y_pred = lr.predict(X_test)
```

In [28]:
```
#calculating r2 value for the model
from sklearn import metrics
r2 = metrics.r2_score(y_test, y_pred)
r2
```

Out[28]:
```
0.9112695892268834
```

In [30]:
```
#calculating the coefficiens
lr.coef_
```

Out[30]:
```
array([-8.80536598e+02, -6.98169073e+02,  5.25845857e-01,  8.44390881e-01,
        1.07574255e-01])
```

In [31]:
```
#calculating the intercept
lr.intercept_
```

Out[31]:
```
-51035.22972403464
```