Title: Predicting Company Profit Based on Expenditure and State Information

Objective:

The goal of this analysis is to build a predictive model that can accurately forecast a company's profit based on its expenditures in productivity, management, and promotions, as well as its state of operation. This model will aid in identifying the key drivers of profit and help companies optimize their budget allocations to maximize profitability.

Dataset Description:

The dataset consists of the following columns:

1. Productivity_Exp: Amount spent on productivity-related expenses (in thousands of dollars).
2. Management_Exp: Amount spent on management-related expenses (in thousands of dollars).
3. Promotions_Exp: Amount spent on promotional activities (in thousands of dollars).
4. State: The state in which the company operates.
5. Profit: The profit of the company (in thousands of dollars)

Setting Up EDA Environment :

> setting up the environment for data analysis using pandas, numpy, and matplotlib.

In [1]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
import matplotlib
matplotlib.rcParams['figure.figsize']=(20,10)
import warnings
warnings.filterwarnings("ignore")
```

Data Collection :

> Collecting data on productivity, management, promotions, state, and profit.

In [2]:
```
#Reading Data from a csv File`
df1=pd.read_csv('companies.csv')
df1.head()
```

Out[2]:

	Productivity_Exp	Management_Exp	Promotions_Exp	State	Profit
0	220349.20	236897.80	521784.10	Texas	242261.83
1	217597.70	251377.59	493898.53	Illinois	241792.06
2	208441.51	201145.55	457934.54	Washington	241050.39
3	199372.41	218671.85	433199.62	Texas	232901.99
4	197107.34	191391.77	416168.42	Washington	216187.94

Asking Questions :

1. What are some common features to explore during data analysis ?
2. What is the correlation coefficient between productivity expenses and profit, and how does it indicate the strength of their relationship?
3. Does an increase in promotional expenses lead to higher profits?
4. What do the box plots reveal about the profit distribution across Texas, Illinois, and Washington?
5. what relationship can be observed between 'Features' and 'Profit'?
6. How does one-hot encoding handle categorical variables in a dataset?
7. What is the purpose of min-max scaling in data preprocessing?
8. How can the effectiveness of each expenditure category be quantified in terms of its impact on profit?
9. How can we divide data into training and testing sets in machine learning?
10. Which model is being used as the starting point for training in this project?
11. What metrics can be used to evaluate model performance ?
12. How can the coefficients of a linear regression model be used to evaluate its performance in predicting profit?

Initial Checkup :

> Exploring the dataset to understand its structure, null values, and potential features.

In [3]:
```
df1.shape
```
Out[3]:
```
(50, 5)
```

In [4]:
```
df1.isnull().sum()
```
Out[4]:
```
Productivity_Exp    0
Management_Exp      0
Promotions_Exp      0
State               0
Profit              0
dtype: int64
```

> There is no null values

In [5]:
```
# Statistical Summary of Columns:
print("Statistical summary of the DataFrame columns:")
print(df1.describe())
```

```
Statistical summary of the DataFrame columns:
       Productivity_Exp  Management_Exp  Promotions_Exp         Profit
count         50.000000       50.000000       50.000000      50.000000
mean      128721.615600   221344.639600   261025.097800  162012.639200
std        45902.256482    28017.802755   122290.310726   40306.180338
min        55000.000000   151283.140000    50000.000000   64681.400000
25%        94936.370000   203730.875000   179300.132500  140138.902500
50%       128051.080000   222699.795000   262716.240000  157978.190000
75%       156602.800000   244842.180000   349469.085000  189765.977500
max       220349.200000   282645.560000   521784.100000  242261.830000
```

In [6]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Productivity_Exp  50 non-null     float64
 1   Management_Exp    50 non-null     float64
 2   Promotions_Exp    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
```

1.What are some common features to explore during data analysis ?

In [7]:

```
df1.columns
```

Out[7]:

```
Index(['Productivity_Exp', 'Management_Exp', 'Promotions_Exp', 'State',
       'Profit'],
      dtype='object')
```

In [8]:

```
df2=df1.drop(['Management_Exp'],axis='columns')
df2.head()
```

Out[8]:

	Productivity_Exp	Promotions_Exp	State	Profit
0	220349.20	521784.10	Texas	242261.83
1	217597.70	493898.53	Illinois	241792.06
2	208441.51	457934.54	Washington	241050.39
3	199372.41	433199.62	Texas	232901.99
4	197107.34	416168.42	Washington	216187.94

In [9]:

```
df2['State'].unique()
```

Out[9]:

```
array(['Texas', 'Illinois', 'Washington'], dtype=object)
```

Data Visualization :

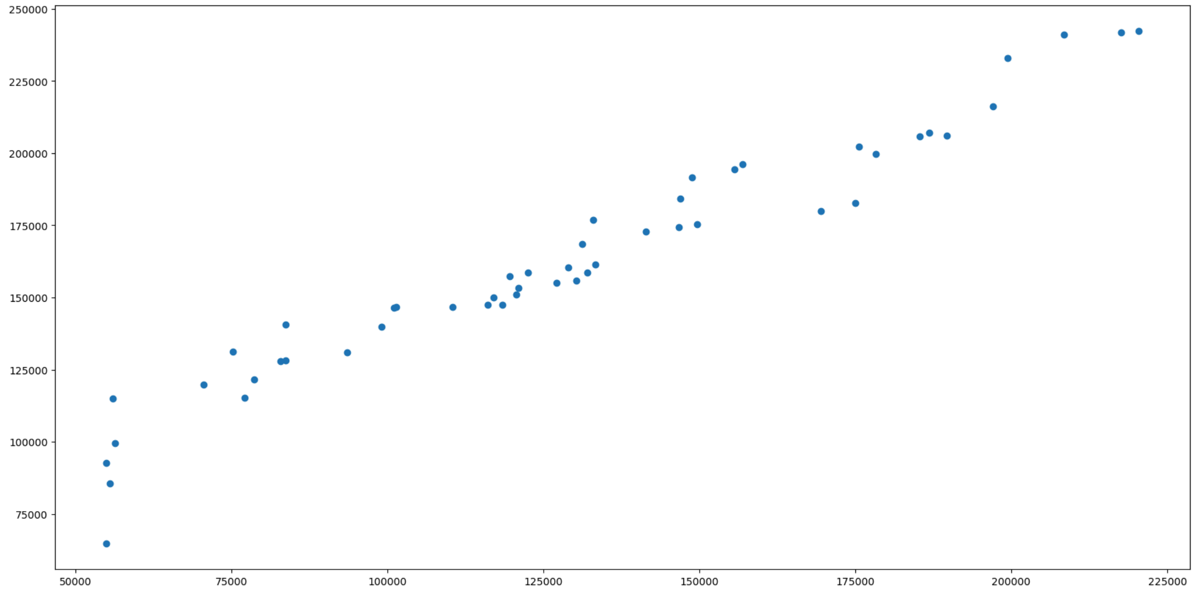
2. What is the correlation coefficient between productivity expenses and profit?

In [10]:

```
# Scatter plot: Productivity Expenses vs. Profit
plt.scatter(df2["Productivity_Exp"], df2["Profit"])
```

Out[10]:

```
<matplotlib.collections.PathCollection at 0x212857ed450>
```

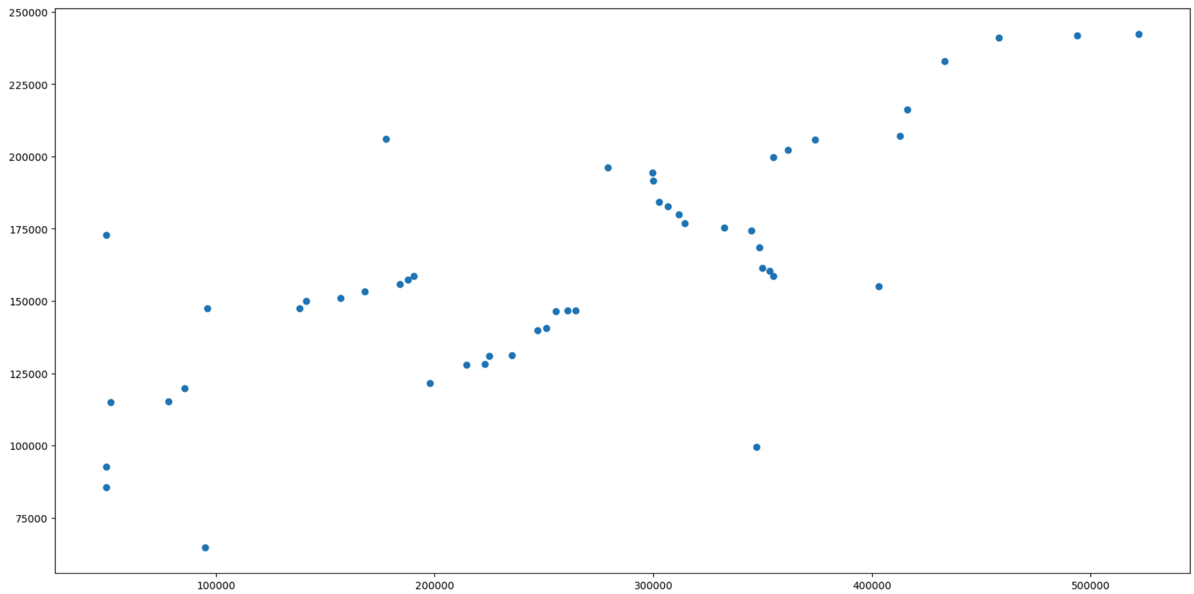


> Relationship between productivity expenses and profit. Each dot represents a company's data point, showing how their productivity expenses impact their profit.

3. Does an increase in promotional expenses lead to higher profits?

In [11]:
```
plt.scatter(df2["Promotions_Exp"], df2["Profit"])
```

Out[11]:
```
<matplotlib.collections.PathCollection at 0x21287b181c0>
```

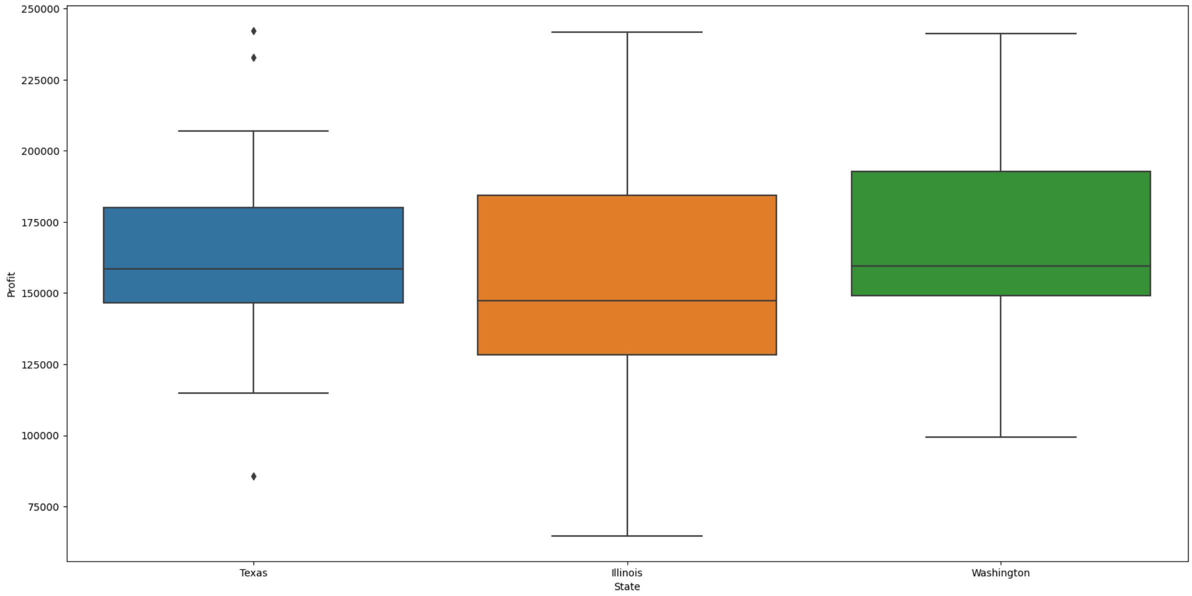


> The above scatter plot shows how promotional expenses relate to profit for each data point. and visualizes how promotional expenses impact profit across different data points.

4. What do the box plots reveal about the profit distribution across Texas, Illinois, and Washington?

In [12]:
```
import seaborn as sns
sns.boxplot(x=df2['State'],y=df2['Profit'])
```

```
Out[12]: <Axes: xlabel='State', ylabel='Profit'>
```



- Texas: The blue box shows Texas companies have a median profit around 17500, with some variability.
- Florida: The orange box indicates Florida companies' median profit is slightly higher than Texas.
- Washington: The green box reveals Washington companies have the highest median profit, around 22500, compared to the other states.

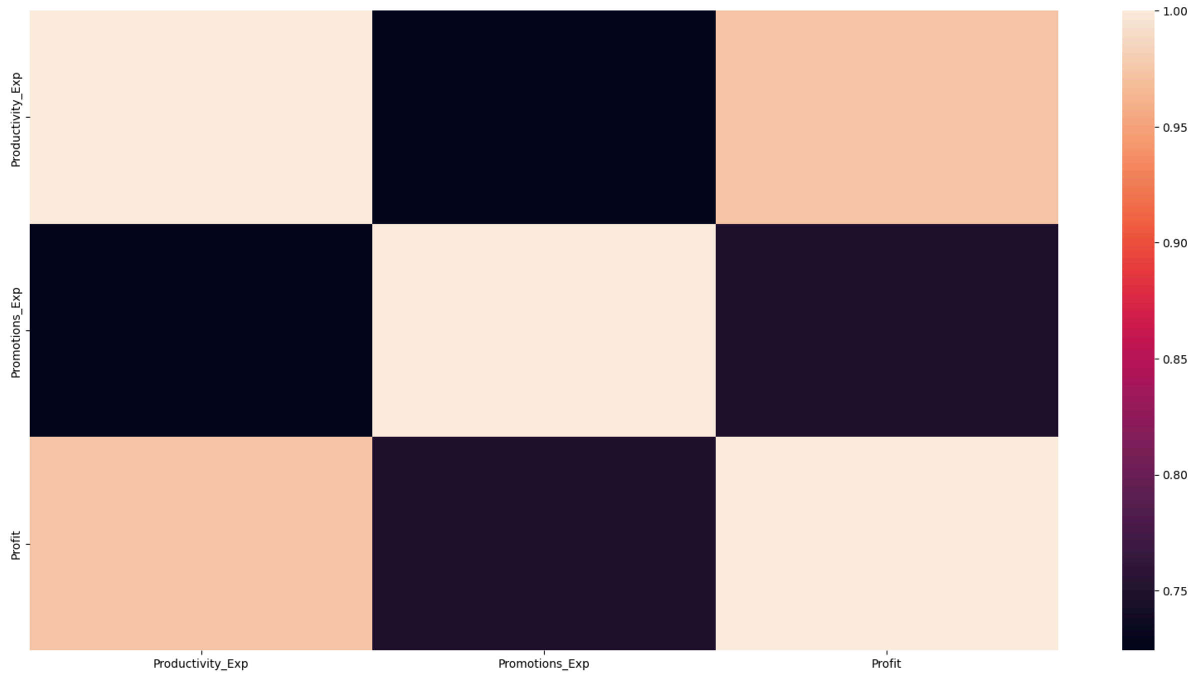
Exploring Correlations Between Features:

> Visualize correlations using heatmaps or scatter plots.

5. what relationship can be observed between 'Features' and 'Profit'?

```
In [16]: correlation_matrix = df2[["Productivity_Exp", "Promotions_Exp", "Profit"]].corr()
         sns.heatmap(correlation_matrix)
```

```
Out[16]: <Axes: >
```



- The colors range from light to dark, where darker shades mean a stronger relationship.
- Relationship: If the color in the cell where ‘Productivity_Exp’ or ‘Promotions_Exp’ meets ‘Profit’ is darker, it means that feature strongly influences profit. Conversely, a lighter color means a weaker influence.

Feature Engineering:

6.How does one-hot encoding handle categorical variables in a dataset?

In [18]:
```
one_hot_encoded_df = pd.get_dummies(df2, columns=['State'])
one_hot_encoded_df
```

Out[18]:

	Productivity_Exp	Promotions_Exp	Profit	State_Illinois	State_Texas	State_Washington
0	220349.20	521784.10	242261.83	0	1	0
1	217597.70	493898.53	241792.06	1	0	0
2	208441.51	457934.54	241050.39	0	0	1
3	199372.41	433199.62	232901.99	0	1	0
4	197107.34	416168.42	216187.94	0	0	1
5	186876.90	412861.36	206991.12	0	1	0
6	189615.46	177716.82	206122.51	1	0	0
7	185298.13	373876.68	205752.60	0	0	1
8	175542.52	361613.29	202211.77	0	1	0
9	178334.88	354981.62	199759.96	1	0	0
10	156913.08	279160.95	196121.95	0	0	1
11	155671.96	299744.55	194259.40	1	0	0
12	148863.75	299839.44	191585.52	0	0	1
13	146992.39	302664.93	184307.35	1	0	0
14	174943.24	306512.92	182602.65	0	0	1
15	169523.61	311776.23	179917.04	0	1	0
16	133013.11	314346.06	176992.93	1	0	0
17	149657.16	332574.31	175370.37	0	1	0
18	146749.16	344919.57	174266.90	0	0	1
19	141419.70	50000.00	172776.86	0	1	0
20	131253.86	348664.47	168474.03	1	0	0
21	133389.47	349737.29	161313.02	0	1	0
22	128994.56	353319.26	160352.25	0	0	1
23	122532.53	354768.73	158733.99	0	0	1
24	132044.01	190574.81	158552.04	0	1	0
25	119664.71	187962.62	157404.34	1	0	0
26	130328.87	184050.07	155733.54	0	0	1
27	127107.60	403183.81	155008.31	0	1	0
28	121051.52	168148.20	153282.38	0	0	1
29	120605.48	157138.38	151004.64	0	1	0
30	116994.48	141131.24	149937.59	0	0	1
31	116136.38	138218.23	147483.56	0	1	0
32	118408.86	96085.25	147427.84	1	0	0
33	110493.95	264634.81	146778.92	0	0	1
34	101426.07	260797.67	146712.80	1	0	0
35	101014.02	255517.64	146479.51	0	1	0

	Productivity_Exp	Promotions_Exp	Profit	State_Illinois	State_Texas	State_Washington
36	83663.76	251126.82	140708.19	0	0	1
37	99069.95	247029.42	139949.14	1	0	0
38	75229.59	235265.10	131229.06	0	1	0
39	93558.51	224999.30	131005.76	1	0	0
40	83754.33	222795.67	128239.91	1	0	0
41	82892.92	214470.71	127798.83	0	0	1
42	78640.93	198001.11	121498.49	1	0	0
43	70505.73	85534.17	119758.98	0	1	0
44	77177.74	78334.72	115200.33	1	0	0
45	56000.23	51903.93	114926.08	0	1	0
46	56315.46	347114.46	99490.75	0	0	1
47	55000.00	50000.00	92559.73	1	0	0
48	55542.05	50000.00	85673.41	0	1	0
49	55000.00	95173.06	64681.40	1	0	0

7.What is the purpose of min-max scaling in data preprocessing?

Scaling Numerical Features (Min-Max Scaling) :

> Min-max scaling transforms features to a specified range (typically between 0 and 1).Ensuring that numerical features are on a similar scale to improve model performance.

In [19]:

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaler.fit_transform(df2[["Productivity_Exp", "Promotions_Exp"]])
```

Out[19]:
```
array([[1.        , 1.        ],
       [0.98335946, 0.94089337],
       [0.92798459, 0.8646636 ],
       [0.87313643, 0.81223513],
       [0.85943772, 0.77613557],
       [0.797566  , 0.76912588],
       [0.81412828, 0.27071031],
       [0.7880179 , 0.68649342],
       [0.72901786, 0.66049977],
       [0.74590551, 0.64644319],
       [0.61635061, 0.48573267],
       [0.60884455, 0.52936195],
       [0.56766982, 0.52956308],
       [0.55635219, 0.53555202],
       [0.72539353, 0.54370828],
       [0.69261666, 0.55486446],
       [0.47180821, 0.56031151],
       [0.57246821, 0.59894835],
       [0.55488118, 0.62511553],
       [0.52264964, 0.        ],
       [0.46116861, 0.63305328],
       [0.47408436, 0.63532724],
       [0.4475048 , 0.64291963],
       [0.40842369, 0.64599195],
       [0.46594728, 0.29796428],
       [0.39107967, 0.29242745],
       [0.45557444, 0.28413435],
       [0.43609283, 0.74861321],
       [0.39946683, 0.25042853],
       [0.39676926, 0.22709197],
       [0.37493063, 0.19316302],
       [0.36974101, 0.18698856],
       [0.38348453, 0.09768292],
       [0.33561668, 0.45494286],
       [0.2807759 , 0.44680961],
       [0.2782839 , 0.43561799],
       [0.17335288, 0.42631115],
       [0.26652654, 0.41762624],
       [0.12234465, 0.39269043],
       [0.23319442, 0.3709309 ],
       [0.17390063, 0.36626005],
       [0.16869099, 0.34861436],
       [0.14297577, 0.31370517],
       [0.09377566, 0.07531871],
       [0.13412668, 0.06005866],
       [0.0060492 , 0.0040356 ],
       [0.00795565, 0.62976785],
       [0.        , 0.        ],
       [0.00327821, 0.        ],
       [0.        , 0.09574943]])
```

8.How can the effectiveness of each expenditure category be quantified in terms of its impact on profit?

Profit per Expenditure Category :

> This helps quantify how effectively each expenditure impacts profit.

In [20]:
```
one_hot_encoded_df["Profit_per_Productivity"] = one_hot_encoded_df["Profit"] / one_
one_hot_encoded_df["Profit_per_Promotions"] = one_hot_encoded_df["Profit"] / one_hc
one_hot_encoded_df["Profit_per_Productivity"]
```

Out[20]:

```
0     1.099445
1     1.111188
2     1.156441
3     1.168176
4     1.096803
5     1.107634
6     1.087055
7     1.110387
8     1.151925
9     1.120140
10    1.249876
11    1.247877
12    1.286986
13    1.253856
14    1.043782
15    1.061310
16    1.330643
17    1.171814
18    1.187515
19    1.221731
20    1.283574
21    1.209338
22    1.243093
23    1.295444
24    1.200751
25    1.315378
26    1.194927
27    1.219505
28    1.266257
29    1.252055
30    1.281578
31    1.269917
32    1.245074
33    1.328389
34    1.446500
35    1.450091
36    1.681830
37    1.412630
38    1.744381
39    1.400255
40    1.531144
41    1.541734
42    1.544978
43    1.698571
44    1.492663
45    2.052243
46    1.766669
47    1.682904
48    1.542496
49    1.176025
Name: Profit_per_Productivity, dtype: float64
```

In [21]:

```
one_hot_encoded_df["Profit_per_Promotions"]
```

Out[21]:
```
0     0.464295
1     0.489558
2     0.526386
3     0.537632
4     0.519472
5     0.501357
6     1.159837
7     0.550322
8     0.559193
9     0.562733
10    0.702541
11    0.648083
12    0.638960
13    0.608948
14    0.595742
15    0.577071
16    0.563051
17    0.527312
18    0.505239
19    3.455537
20    0.483198
21    0.461241
22    0.453845
23    0.447429
24    0.831967
25    0.837424
26    0.846148
27    0.384461
28    0.911591
29    0.960966
30    1.062398
31    1.067034
32    1.534344
33    0.554647
34    0.562554
35    0.573266
36    0.560307
37    0.566528
38    0.557792
39    0.582250
40    0.575594
41    0.595880
42    0.613625
43    1.400130
44    1.470616
45    2.214208
46    0.286622
47    1.851195
48    1.713468
49    0.679619
Name: Profit_per_Promotions, dtype: float64
```

9.How can we divide data into training and testing sets in machine learning?

Spliting Data into Training and Testing Sets :

- **Training set:** Used to train the ML model.
- **Validation (or test) set:** Used to evaluate model accuracy.

In [22]:
```
from sklearn.model_selection import train_test_split
X = one_hot_encoded_df.drop(columns=["Profit"])
y = one_hot_encoded_df["Profit"]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.8)
X_test
```

Out[22]:

	Productivity_Exp	Promotions_Exp	State_Illinois	State_Texas	State_Washington	Profit_per_Pro
10	156913.08	279160.95	0	0	1	
13	146992.39	302664.93	1	0	0	
19	141419.70	50000.00	0	1	0	
12	148863.75	299839.44	0	0	1	
5	186876.90	412861.36	0	1	0	
3	199372.41	433199.62	0	1	0	
31	116136.38	138218.23	0	1	0	
36	83663.76	251126.82	0	0	1	
26	130328.87	184050.07	0	0	1	
0	220349.20	521784.10	0	1	0	
17	149657.16	332574.31	0	1	0	
25	119664.71	187962.62	1	0	0	
37	99069.95	247029.42	1	0	0	
48	55542.05	50000.00	0	1	0	
43	70505.73	85534.17	0	1	0	
21	133389.47	349737.29	0	1	0	
8	175542.52	361613.29	0	1	0	
40	83754.33	222795.67	1	0	0	
24	132044.01	190574.81	0	1	0	
6	189615.46	177716.82	1	0	0	
47	55000.00	50000.00	1	0	0	
27	127107.60	403183.81	0	1	0	
46	56315.46	347114.46	0	0	1	
45	56000.23	51903.93	0	1	0	
34	101426.07	260797.67	1	0	0	
23	122532.53	354768.73	0	0	1	
14	174943.24	306512.92	0	0	1	
44	77177.74	78334.72	1	0	0	
15	169523.61	311776.23	0	1	0	
32	118408.86	96085.25	1	0	0	
2	208441.51	457934.54	0	0	1	
35	101014.02	255517.64	0	1	0	
39	93558.51	224999.30	1	0	0	
20	131253.86	348664.47	1	0	0	
42	78640.93	198001.11	1	0	0	
9	178334.88	354981.62	1	0	0	

	Productivity_Exp	Promotions_Exp	State_Illinois	State_Texas	State_Washington	Profit_per_Pro
30	116994.48	141131.24	0	0	1	
33	110493.95	264634.81	0	0	1	
22	128994.56	353319.26	0	0	1	
38	75229.59	235265.10	0	1	0	

10.Which model is being used as the starting point for training in this project?

Model Training

> For this project, I am using Linear Regression as a starting point. and Training the model using the training data.

In [31]:
```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
y_pred
```

Out[31]:
```
array([190078.66899907, 181465.16898111, 201061.22371932, 183318.40468316,
       206129.95299943, 227500.21941205, 148966.39425225, 146636.59372054,
       155563.1851013 , 243176.47267334, 169884.75534383, 159506.42329699,
       141198.62826042, 113335.8399412 , 143716.13330148, 153357.48290133,
       198659.63658463, 135788.40299316, 157397.74204426, 222912.98178202,
       133690.84623668, 145209.33562877, 118304.64493129, 172678.93025377,
       147306.86043126, 149735.61347283, 188457.09882209, 135768.657665  ,
       182948.69096512, 159548.79122609, 237177.26691378, 142587.7902348 ,
       133827.99280123, 163644.84035999, 131927.63611725, 203617.1165641 ,
       151693.4637437 , 141393.64120521, 152072.20247879, 142760.99748219])
```

> The y_pred array contains predicted profit values for the validation set. Each value corresponds to a company's predicted profit based on the model's learned relationships between features (expenditure categories) and profit.

11.What metrics can be used to evaluate model performance ?

Model Evaluation

> Evaluating the model's performance on the validation set. By using metrics like **Mean Squared Error (MSE), R-squared**

In [34]:
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")
```

```
Mean Squared Error: 234076510.43
R-squared: 0.83
```

12.How can the coefficients of a linear regression model be used to evaluate its performance in predicting profit?

Interpretation:

> Interpreting the model coefficients (for linear regression). and understanding which features contribute most to profit prediction.

In [107…

```
print("Model Coefficients:")
for feature, coef in zip(X.columns, model.coef_):
    print(f"{feature}: {coef:.2f}")
```

```
Model Coefficients:
Productivity_Exp: 1.21
Promotions_Exp: -0.06
Profit_per_Productivity: 77921.52
Profit_per_Promotions: -15875.09
State_Illinois: 550.63
State_Texas: -453.36
State_Washington: -97.27
```

Conclusion :

> The model is quite effective in predicting profit, with productivity being a key positive driver. Promotions do not seem to contribute significantly to profit, and there are notable differences based on the state.