21bce5304-multipleregression

January 21, 2024

Objective: Implementation of Multiple Linear Regression

NAME: Rishikesh S

REGNO: 21BCE5304

[61]:
```
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```
```
import numpy as np
```

2.Dataset Descrption

The dataset you're referring to is related to temperature estimation.

[81]:
```
data=pd.read_csv("startup.csv")
```

3.Exploratory Analytics

[82]:
```
data.describe()
```

[82]:

	R&D Spend	Administration	Marketing Spend	Profit
count	50.000000	50.000000	50.000000	50.000000
mean	73721.615600	121344.639600	211025.097800	112012.639200
std	45902.256482	28017.802755	122290.310726	40306.180338
min	0.000000	51283.140000	0.000000	14681.400000
25%	39936.370000	103730.875000	129300.132500	90138.902500
50%	73051.080000	122699.795000	212716.240000	107978.190000
75%	101602.800000	144842.180000	299469.085000	139765.977500
max	165349.200000	182645.560000	471784.100000	192261.830000

[83]:
```
data.head()
```

[83]:

	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
```

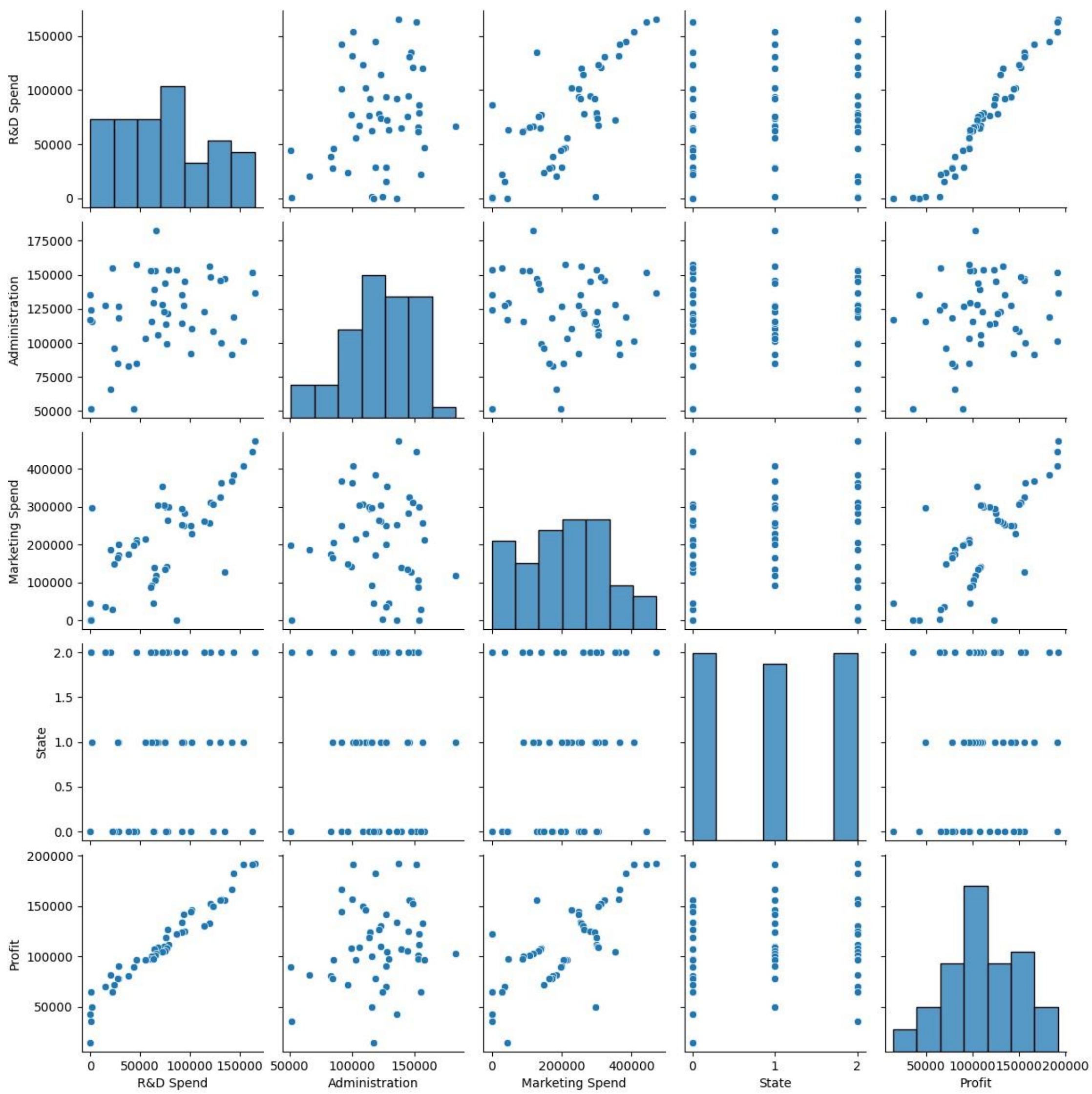
[84]:
```python
data["State"].unique()
```

[84]:
```
array(['New York', 'California', 'Florida'], dtype=object)
```

[85]:
```python
from sklearn.preprocessing import LabelEncoder
lbl=LabelEncoder()
data["State"]=lbl.fit_transform(data["State"])
```

[86]:
```python
sns.pairplot(data)
```

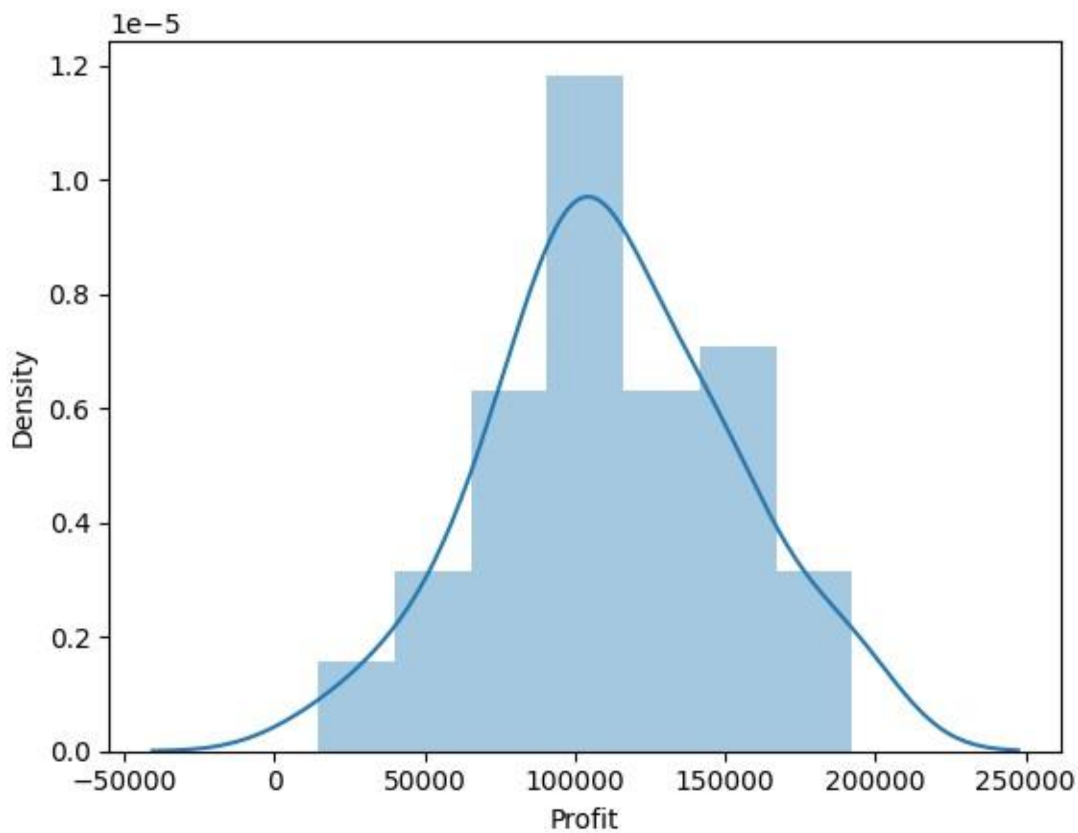
[86]:
```
<seaborn.axisgrid.PairGrid at 0x7fabe6addd20>
```



[87]:
```
sns.distplot(data['Profit'])
```

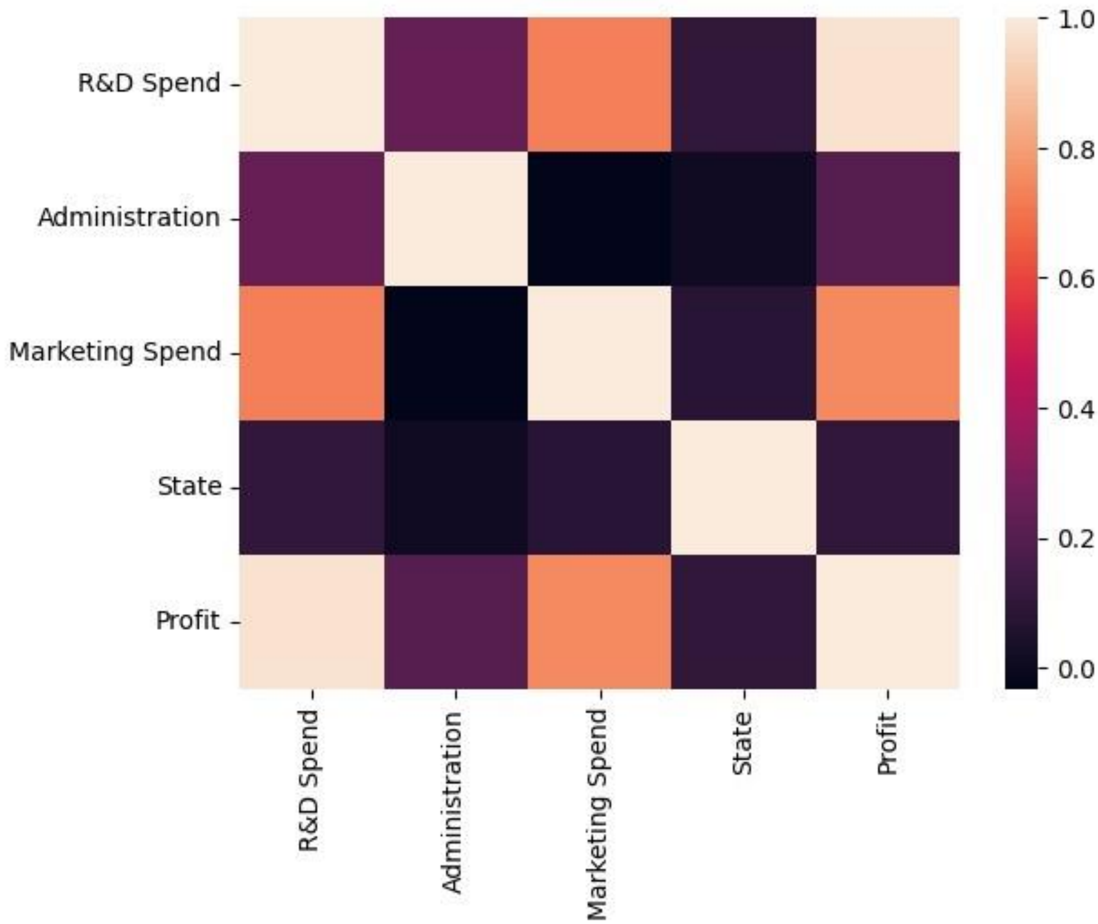
```
<ipython-input-87-5c9dc59bcdb4>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
                                                          v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level
function for histograms).

For a guide to updating your code to use the new functions, please

see https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

sns.distplot(data['Profit'])
```

[87]: <Axes: xlabel='Profit', ylabel='Density'>



[88]:
```
sns.heatmap(data.corr())
```

[88]: <Axes: >



[89]:
```
x = data.iloc[:, :-1].values
y=data.iloc[:,-1].values
```

[90]:
```
print(x)
print(y)
```

```
[[1.6534920e+05 1.3689780e+05 4.7178410e+05 2.0000000e+00]
 [1.6259770e+05 1.5137759e+05 4.4389853e+05 0.0000000e+00]
 [1.5344151e+05 1.0114555e+05 4.0793454e+05 1.0000000e+00]
 [1.4437241e+05 1.1867185e+05 3.8319962e+05 2.0000000e+00]
 [1.4210734e+05 9.1391770e+04 3.6616842e+05 1.0000000e+00]
 [1.3187690e+05 9.9814710e+04 3.6286136e+05 2.0000000e+00]
 [1.3461546e+05 1.4719887e+05 1.2771682e+05 0.0000000e+00]
 [1.3029813e+05 1.4553006e+05 3.2387668e+05 1.0000000e+00]
 [1.2054252e+05 1.4871895e+05 3.1161329e+05 2.0000000e+00]
 [1.2333488e+05 1.0867917e+05 3.0498162e+05 0.0000000e+00]
 [1.0191308e+05 1.1059411e+05 2.2916095e+05 1.0000000e+00]
```

```
 [1.0067196e+05 9.1790610e+04 2.4974455e+05 0.0000000e+00]
 [9.3863750e+04 1.2732038e+05 2.4983944e+05 1.0000000e+00]
 [9.1992390e+04 1.3549507e+05 2.5266493e+05 0.0000000e+00]
 [1.1994324e+05 1.5654742e+05 2.5651292e+05 1.0000000e+00]
 [1.1452361e+05 1.2261684e+05 2.6177623e+05 2.0000000e+00]
 [7.8013110e+04 1.2159755e+05 2.6434606e+05 0.0000000e+00]
 [9.4657160e+04 1.4507758e+05 2.8257431e+05 2.0000000e+00]
 [9.1749160e+04 1.1417579e+05 2.9491957e+05 1.0000000e+00]
 [8.6419700e+04 1.5351411e+05 0.0000000e+00 2.0000000e+00]
 [7.6253860e+04 1.1386730e+05 2.9866447e+05 0.0000000e+00]
 [7.8389470e+04 1.5377343e+05 2.9973729e+05 2.0000000e+00]
 [7.3994560e+04 1.2278275e+05 3.0331926e+05 1.0000000e+00]
 [6.7532530e+04 1.0575103e+05 3.0476873e+05 1.0000000e+00]
 [7.7044010e+04 9.9281340e+04 1.4057481e+05 2.0000000e+00]
 [6.4664710e+04 1.3955316e+05 1.3796262e+05 0.0000000e+00]
 [7.5328870e+04 1.4413598e+05 1.3405007e+05 1.0000000e+00]
 [7.2107600e+04 1.2786455e+05 3.5318381e+05 2.0000000e+00]
 [6.6051520e+04 1.8264556e+05 1.1814820e+05 1.0000000e+00]
 [6.5605480e+04 1.5303206e+05 1.0713838e+05 2.0000000e+00]
 [6.1994480e+04 1.5564128e+05 9.1131240e+04 1.0000000e+00]
 [6.1136380e+04 1.5270192e+05 8.8218230e+04 2.0000000e+00]
 [6.3408860e+04 1.2921961e+05 4.6085250e+04 0.0000000e+00]
 [5.5493950e+04 1.0305749e+05 2.1463481e+05 1.0000000e+00]
 [4.6426070e+04 1.5769392e+05 2.1079767e+05 0.0000000e+00]
 [4.6014020e+04 8.5047440e+04 2.0551764e+05 2.0000000e+00]
 [2.8663760e+04 1.2705621e+05 2.0112682e+05 1.0000000e+00]
 [4.4069950e+04 5.1283140e+04 1.9702942e+05 0.0000000e+00]
 [2.0229590e+04 6.5947930e+04 1.8526510e+05 2.0000000e+00]
 [3.8558510e+04 8.2982090e+04 1.7499930e+05 0.0000000e+00]
 [2.8754330e+04 1.1854605e+05 1.7279567e+05 0.0000000e+00]
 [2.7892920e+04 8.4710770e+04 1.6447071e+05 1.0000000e+00]
 [2.3640930e+04 9.6189630e+04 1.4800111e+05 0.0000000e+00]
 [1.5505730e+04 1.2738230e+05 3.5534170e+04 2.0000000e+00]
 [2.2177740e+04 1.5480614e+05 2.8334720e+04 0.0000000e+00]
 [1.0002300e+03 1.2415304e+05 1.9039300e+03 2.0000000e+00]
 [1.3154600e+03 1.1581621e+05 2.9711446e+05 1.0000000e+00]
 [0.0000000e+00 1.3542692e+05 0.0000000e+00 0.0000000e+00]
 [5.4205000e+02 5.1743150e+04 0.0000000e+00 2.0000000e+00]
 [0.0000000e+00 1.1698380e+05 4.5173060e+04 0.0000000e+00]]
[192261.83 191792.06 191050.39 182901.99 166187.94 156991.12
156122.51
 155752.6 152211.77 149759.96 146121.95 144259.4 141585.52 134307.35
 132602.65 129917.04 126992.93 125370.37 124266.9 122776.86 118474.03
 111313.02 110352.25 108733.99 108552.04 107404.34 105733.54
 105008.31 103282.38 101004.64 99937.59 97483.56 97427.84 96778.92
 96712.8
```

```
 96479.51 90708.19 89949.14 81229.06 81005.76 78239.91 77798.83
   71498.49 69758.98 65200.33 64926.08 49490.75 42559.73 35673.41
   14681.4 ]
```

[43]:
```python
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 = 1)
```

[44]:
```python
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)
```

[91]:
```python
data.head()
```

[91]:

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

[92]:
```python
data.rename(columns={'Marketing Spend': 'Marketing_Spend'}, inplace=True)

data.head()
```

[92]:

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

[93]:
```python
data.rename(columns={'R&D Spend': 'RD_Spend'}, inplace=True)
data.head()
```

[93]:

	RD_Spend	Administration	Marketing_Spend	State	Profit
0	165349.20	136897.80	471784.10	2	192261.83
1	162597.70	151377.59	443898.53	0	191792.06
2	153441.51	101145.55	407934.54	1	191050.39
3	144372.41	118671.85	383199.62	2	182901.99
4	142107.34	91391.77	366168.42	1	166187.94

Methodology AND Multiple Model Analysis

[94]:
```
import statsmodels.formula.api as smf
from sklearn.metrics import mean_squared_error, mean_absolute_error
import numpy as np
```

[95]:
```
# Initialize variables to store the results
results = []
```

```python
# Create a function to build a model and print summary
def build_model(features, data, model_number):
    formula = f'Profit ~ {" + ".join(features)}'
    model = smf.ols(formula=formula, data=data).fit()

    # Get model performance metrics
    y_pred = model.predict(data[features])
    mse = mean_squared_error(data['Profit'], y_pred)
    mae = mean_absolute_error(data['Profit'], y_pred)
    rmse = np.sqrt(mse)
    r_squared = model.rsquared

    # Append results to the list
    p_values = model.pvalues[1:]  # Exclude intercept
    results.append([model_number, formula, features, mse, mae, rmse, r_squared,␣
↪p_values])

    # Print model summary
    print(f"Model {model_number} - {formula}")
    print("MSE:", mse)
    print("MAE:", mae)
    print("RMSE:", rmse)
    print("R-squared:", r_squared)
    print("P-values:")
    print(p_values)
    print(model.summary())
    print("\n")
```

[97]:
```python
# Consider all features initially
all_features = [ 'RD_Spend','Administration','Marketing_Spend','State' ]
# Set a threshold for p-value
p_value_threshold = 0.05

# Build models and print summary based on p-values
model_number = 1
while all_features:
    build_model(all_features, data, model_number)
    model_number += 1

    # Get p-values for all features in the current model
    p_values = smf.ols(formula=f'Profit ~ {" + ".join(all_features)}',␣
↪data=data).fit().pvalues[1:]  # Exclude intercept

    # Identify the feature with the highest p-value
    max_p_value_feature = p_values.idxmax()
    max_p_value = p_values[max_p_value_feature]
```

```python
    # Check if the highest p-value is above the threshold
    if max_p_value > p_value_threshold:
        # Remove the feature with the highest p-value
        all_features.remove(max_p_value_feature)
    else:
        # Break the loop if all features have p-values below the threshold
        break
```

```
Model 1 - Profit ~ RD_Spend + Administration + Marketing_Spend +
State
MSE: 78416791.01666646
MAE: 6468.105695552672
RMSE: 8855.325573724913 R-
squared: 0.9507462044842656
P-values:
RD_Spend          8.249206e-
                  22
Administration    6.056771e-
                  01
Marketing_Spend   1.086131e-
                  01
State dtype:      9.889988e-
float64           01
                        OLS Regression Results
======================================================================
=========
Dep. Variable:              Profit   R-squared:                   0.951
Model:                         OLS   Adj. R-squared:              0.946
Method:              Least Squares   F-statistic:                 217.2
Date:             Sun, 21 Jan 2024Prob (F-statistic):           8.51e-
                                                                29
Time:                     13:33:05   Log-Likelihood:               -
                                                                525.39
No. Observations:               50   AIC:                         1061.
Df Residuals:                   45   BIC:                         1070.
Df Model:                        4

Covariance Type:         nonrobust

======================================================================
=========== ===
                   coef    std err          t      P>|t|     [0.025
0.975] ---------------------------------------------------------------
------------------
---
Intercept      5.014e+04 6804.555      7.369     0.000   3.64e+04
```

```
6.38e+04

RD_Spend         0.8058     0.046    17.609    0.000     0.714
0.898
Administration -0.0268      0.052    -0.520    0.606    -0.131
0.077
Marketing_Spend  0.0272     0.017     1.637    0.109    -0.006
0.061
State          -22.3206 1609.829    -0.014    0.989        -
3220.041                                   3264.682
======================================================================
=========
Omnibus:                      14.864  Durbin-Watson:              1.282
Prob(Omnibus):                 0.001  Jarque-Bera (JB):          21.542
Skew:                         -0.949  Prob(JB):                2.10e-05
Kurtosis:                      5.596  Cond. No.                1.44e+06
======================================================================
=========

Notes:
[1]  Standard Errors assume that the covariance matrix of the errors
is correctly specified.
[2]  The condition number is large, 1.44e+06. This might indicate
that there are strong multicollinearity or other numerical problems.


Model 2 - Profit ~ RD_Spend + Administration + Marketing_Spend
MSE: 78417126.01913084
MAE: 6471.45039610481
RMSE: 8855.344489015142 R-
squared: 0.9507459940683246
P-values:
RD_Spend        2.634968e-22
Administration 6.017551e-01
Marketing_Spend 1.047168e-01
dtype: float64
                       OLS Regression Results
======================================================================
=========
Dep. Variable:               Profit  R-squared:                  0.951
Model:                          OLS  Adj. R-squared:             0.948
Method:               Least Squares  F-statistic:                296.0
Date:              Sun, 21 Jan 2024Prob (F-statistic):        4.53e-
                                                              30
Time:                      13:33:05  Log-Likelihood:             -
                                                              525.39
No. Observations:                50  AIC:                        1059.
```

```
Df Residuals:                    46 BIC:                          1066.
Df Model:                         3

Covariance Type:          nonrobust

==============================================================================
=========== ===
                   coef    std err          t      P>|t|      [0.025
0.975] ----------------------------------------------------------------------
------------------
---
Intercept       5.012e+04 6572.353     7.626     0.000    3.69e+04
6.34e+04
RD_Spend          0.8057     0.045    17.846     0.000       0.715
0.897
Administration -0.0268      0.051    -0.526     0.602      -0.130
0.076
Marketing_Spend  0.0272     0.016     1.655     0.105      -0.006
0.060
==============================================================================
=========
Omnibus:                     14.838  Durbin-Watson:                   1.282
Prob(Omnibus):                0.001  Jarque-Bera (JB):               21.442
Skew:                        -0.949  Prob(JB):                     2.21e-05
Kurtosis:                     5.586  Cond. No.                     1.40e+06
==============================================================================
=========

Notes:
[1]  Standard Errors assume that the covariance matrix of the errors
is correctly specified.
[2]  The condition number is large, 1.4e+06. This might indicate that
there are strong multicollinearity or other numerical problems.


Model 3 - Profit ~ RD_Spend + Marketing_Spend
MSE: 78887897.00648756
MAE: 6499.319940113646
RMSE: 8881.885892449169 R-
squared: 0.9504503015559763
P-values:
RD_Spend          6.040433e-24
Marketing_Spend   6.003040e-02
dtype: float64
                        OLS Regression Results
==============================================================================
=========
```

```
Dep. Variable:              Profit  R-squared:                       0.950
Model:                         OLS  Adj. R-squared:                  0.948
Method:              Least Squares F-statistic:                      450.8
Date:             Sun, 21 Jan 2024Prob (F-statistic):             2.16e-
                                                                  31
Time:                     13:33:05 Log-Likelihood:                 -
                                                                   525.54
No. Observations:               50 AIC:                            1057.
Df Residuals:                   47 BIC:                            1063.
Df Model:                        2

Covariance Type:         nonrobust

======================================================================
=========== ===
                   coef   std err         t     P>|t|     [0.025
0.975] ---------------------------------------------------------------
------------------
---
Intercept      4.698e+04 2689.933    17.464     0.000   4.16e+04
5.24e+04
RD_Spend          0.7966     0.041    19.266     0.000      0.713
0.880
Marketing_Spend   0.0299     0.016     1.927     0.060     -0.001
0.061
======================================================================
=========
Omnibus:                    14.677 Durbin-Watson:                    1.257
Prob(Omnibus):               0.001 Jarque-Bera (JB):                21.161
Skew:                       -0.939  Prob(JB):                     2.54e-05
Kurtosis:                    5.575 Cond. No.                      5.32e+05
======================================================================
=========
```

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.32e+05. This might indicate that there are strong multicollinearity or other numerical problems.

```
Model 4 - Profit ~ RD_Spend
MSE: 85120931.32706906
MAE: 6910.984354579612
RMSE: 9226.100548285232 R-
squared: 0.9465353160804393
P-values:
```

```
RD_Spend   3.500322e-32
dtype: float64
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Profit   R-squared:                       0.947
Model:                            OLS   Adj. R-squared:                  0.945
Method:                 Least Squares   F-statistic:                     849.8
Date:                Sun, 21 Jan 2024   Prob (F-statistic):           3.50e-32
Time:                        13:33:06   Log-Likelihood:                -527.44
No. Observations:                  50   AIC:                             1059.
Df Residuals:                      48   BIC:                             1063.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept   4.903e+04   2537.897     19.320      0.000    4.39e+04    5.41e+04
RD_Spend       0.8543      0.029     29.151      0.000       0.795       0.913
==============================================================================
Omnibus:                       13.727   Durbin-Watson:                   1.116
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               18.536
Skew:                          -0.911   Prob(JB):                     9.44e-05
Kurtosis:                       5.361   Cond. No.                     1.65e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.
[2] The condition number is large, 1.65e+05. This might indicate
that there are strong multicollinearity or other numerical problems.
```

Multiple regression on the best model obtained above using Scikt Learn

[98]:
```
# Create a DataFrame from the results list results_df =
pd.DataFrame(results, columns=['No', 'Regression Model name',
'Independent variables chosen', 'MSE', 'MAE', 'RMSE', 'R-
square', 'P-values'])
```

```
[100]: # Choose the best model based on the lowest RMSE
best_model =
results_df.loc[results_df['RMSE'].idxmin()]

# Extract features and target variable for the
best model best_features = best_model['Independent
variables chosen']
X =
data[best_features]
y = data['Profit']
```

```
[101]: from sklearn.model_selection import
train_test_split from sklearn.linear_model
import LinearRegression
from sklearn.metrics import mean_squared_error,
mean_absolute_error import numpy as np

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
 test_size=0.2,␣ ↪random_state=42)

# Train a Linear Regression model using scikit-
learn
model_sklearn = LinearRegression()
model_sklearn.fit(X_train, y_train)
```

```
[101]: LinearRegression()
```

Results

```
[102]: # Make predictions on the test
set y_pred =
model_sklearn.predict(X_test)

# Evaluate the model performance on the test
set mse_test = mean_squared_error(y_test,
y_pred) mae_test =
mean_absolute_error(y_test, y_pred) rmse_test
= np.sqrt(mse_test)
r_squared_test = model_sklearn.score(X_test, y_test)

# Print model performance on the test set
print("Best Model -", best_model['Regression Model name'])
print("Test Set Metrics:")
```

```python
print("MSE:", mse_test)
print("MAE:", mae_test)
print("RMSE:", rmse_test)
print("R-squared:", r_squared_test)
```

```
Best Model - Profit ~ RD_Spend + Administration + Marketing_Spend
+ State Test Set Metrics:
MSE: 59510962.80787997
MAE: 6077.363300620398
RMSE: 7714.334890830185
R-squared: 0.9265108109341951
```

[103]:

```python
results_df
```

[103]:

```
[103] results_df
```

	No	Regression Model name	Independent variables chosen	MSE	MAE	RMSE	R-square	P-values
0	1	Profit ~ RD_Spend + Administration + Marketing...	[RD_Spend]	7.841679e+07	6468.105696	8855.325574	0.950746	RD_Spend 8.249206e-22 Administration...
1	2	Profit ~ RD_Spend + Administration + Marketing...	[RD_Spend]	7.841713e+07	6471.450396	8855.344489	0.950746	RD_Spend 2.634968e-22 Administration...
2	3	Profit ~ RD_Spend + Marketing_Spend	[RD_Spend]	7.888790e+07	6499.319940	8881.885892	0.950450	RD_Spend 6.040433e-24 Marketing_Spen...
3	4	Profit ~ RD_Spend	[RD_Spend]	8.512093e+07	6910.984355	9226.100548	0.946535	RD_Spend 3.500322e-32 dtype: float64