

```
In [2]: import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt
```

```
In [3]: df = pd.read_csv("C:\\\\Users\\\\Vishal\\\\Downloads\\\\All Codes\\\\Regression\\\\50_Startups.csv")
```

```
In [4]: df.head()
```

Out[4]:

| | 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 [5]: df.info()
```

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

```
In [6]: # divide data into x and y i.e independent and dependent variables
```

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

In [7]: X

```
Out[7]: array([[165349.2, 136897.8, 471784.1, 'New York'],
   [162597.7, 151377.59, 443898.53, 'California'],
   [153441.51, 101145.55, 407934.54, 'Florida'],
   [144372.41, 118671.85, 383199.62, 'New York'],
   [142107.34, 91391.77, 366168.42, 'Florida'],
   [131876.9, 99814.71, 362861.36, 'New York'],
   [134615.46, 147198.87, 127716.82, 'California'],
   [130298.13, 145530.06, 323876.68, 'Florida'],
   [120542.52, 148718.95, 311613.29, 'New York'],
   [123334.88, 108679.17, 304981.62, 'California'],
   [101913.08, 110594.11, 229160.95, 'Florida'],
   [100671.96, 91790.61, 249744.55, 'California'],
   [93863.75, 127320.38, 249839.44, 'Florida'],
   [91992.39, 135495.07, 252664.93, 'California'],
   [119943.24, 156547.42, 256512.92, 'Florida'],
   [114523.61, 122616.84, 261776.23, 'New York'],
   [78013.11, 121597.55, 264346.06, 'California'],
   [94657.16, 145077.58, 282574.31, 'New York'],
   [91749.16, 114175.79, 294919.57, 'Florida'],
   [86419.7, 153514.11, 0.0, 'New York'],
   [76253.86, 113867.3, 298664.47, 'California'],
   [78389.47, 153773.43, 299737.29, 'New York'],
   [73994.56, 122782.75, 303319.26, 'Florida'],
   [67532.53, 105751.03, 304768.73, 'Florida'],
   [77044.01, 99281.34, 140574.81, 'New York'],
   [64664.71, 139553.16, 137962.62, 'California'],
   [75328.87, 144135.98, 134050.07, 'Florida'],
   [72107.6, 127864.55, 353183.81, 'New York'],
   [66051.52, 182645.56, 118148.2, 'Florida'],
   [65605.48, 153032.06, 107138.38, 'New York'],
   [61994.48, 115641.28, 91131.24, 'Florida'],
   [61136.38, 152701.92, 88218.23, 'New York'],
   [63408.86, 129219.61, 46085.25, 'California'],
   [55493.95, 103057.49, 214634.81, 'Florida'],
   [46426.07, 157693.92, 210797.67, 'California'],
   [46014.02, 85047.44, 205517.64, 'New York'],
   [28663.76, 127056.21, 201126.82, 'Florida'],
   [44069.95, 51283.14, 197029.42, 'California'],
   [20229.59, 65947.93, 185265.1, 'New York'],
   [38558.51, 82982.09, 174999.3, 'California'],
   [28754.33, 118546.05, 172795.67, 'California'],
   [27892.92, 84710.77, 164470.71, 'Florida'],
   [23640.93, 96189.63, 148001.11, 'California'],
   [15505.73, 127382.3, 35534.17, 'New York'],
   [22177.74, 154806.14, 28334.72, 'California'],
   [1000.23, 124153.04, 1903.93, 'New York'],
   [1315.46, 115816.21, 297114.46, 'Florida'],
   [0.0, 135426.92, 0.0, 'California'],
   [542.05, 51743.15, 0.0, 'New York'],
   [0.0, 116983.8, 45173.06, 'California]]], dtype=object)
```

In [8]: y

```
Out[8]: array([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 ])
```

In [9]: `from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer`

```
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [3])], remainder='passthrough')
X = np.array(ct.fit_transform(X))
```

In [10]: X

```
Out[10]: 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],  
[0.0, 0.0, 1.0, 144372.41, 118671.85, 383199.62],  
[0.0, 1.0, 0.0, 142107.34, 91391.77, 366168.42],  
[0.0, 0.0, 1.0, 131876.9, 99814.71, 362861.36],  
[1.0, 0.0, 0.0, 134615.46, 147198.87, 127716.82],  
[0.0, 1.0, 0.0, 130298.13, 145530.06, 323876.68],  
[0.0, 0.0, 1.0, 120542.52, 148718.95, 311613.29],  
[1.0, 0.0, 0.0, 123334.88, 108679.17, 304981.62],  
[0.0, 1.0, 0.0, 101913.08, 110594.11, 229160.95],  
[1.0, 0.0, 0.0, 100671.96, 91790.61, 249744.55],  
[0.0, 1.0, 0.0, 93863.75, 127320.38, 249839.44],  
[1.0, 0.0, 0.0, 91992.39, 135495.07, 252664.93],  
[0.0, 1.0, 0.0, 119943.24, 156547.42, 256512.92],  
[0.0, 0.0, 1.0, 114523.61, 122616.84, 261776.23],  
[1.0, 0.0, 0.0, 78013.11, 121597.55, 264346.06],  
[0.0, 0.0, 1.0, 94657.16, 145077.58, 282574.31],  
[0.0, 1.0, 0.0, 91749.16, 114175.79, 294919.57],  
[0.0, 0.0, 1.0, 86419.7, 153514.11, 0.0],  
[1.0, 0.0, 0.0, 76253.86, 113867.3, 298664.47],  
[0.0, 0.0, 1.0, 78389.47, 153773.43, 299737.29],  
[0.0, 1.0, 0.0, 73994.56, 122782.75, 303319.26],  
[0.0, 1.0, 0.0, 67532.53, 105751.03, 304768.73],  
[0.0, 0.0, 1.0, 77044.01, 99281.34, 140574.81],  
[1.0, 0.0, 0.0, 64664.71, 139553.16, 137962.62],  
[0.0, 1.0, 0.0, 75328.87, 144135.98, 134050.07],  
[0.0, 0.0, 1.0, 72107.6, 127864.55, 353183.81],  
[0.0, 1.0, 0.0, 66051.52, 182645.56, 118148.2],  
[0.0, 0.0, 1.0, 65605.48, 153032.06, 107138.38],  
[0.0, 1.0, 0.0, 61994.48, 115641.28, 91131.24],  
[0.0, 0.0, 1.0, 61136.38, 152701.92, 88218.23],  
[1.0, 0.0, 0.0, 63408.86, 129219.61, 46085.25],  
[0.0, 1.0, 0.0, 55493.95, 103057.49, 214634.81],  
[1.0, 0.0, 0.0, 46426.07, 157693.92, 210797.67],  
[0.0, 0.0, 1.0, 46014.02, 85047.44, 205517.64],  
[0.0, 1.0, 0.0, 28663.76, 127056.21, 201126.82],  
[1.0, 0.0, 0.0, 44069.95, 51283.14, 197029.42],  
[0.0, 0.0, 1.0, 20229.59, 65947.93, 185265.1],  
[1.0, 0.0, 0.0, 38558.51, 82982.09, 174999.3],  
[1.0, 0.0, 0.0, 28754.33, 118546.05, 172795.67],  
[0.0, 1.0, 0.0, 27892.92, 84710.77, 164470.71],  
[1.0, 0.0, 0.0, 23640.93, 96189.63, 148001.11],  
[0.0, 0.0, 1.0, 15505.73, 127382.3, 35534.17],  
[1.0, 0.0, 0.0, 22177.74, 154806.14, 28334.72],  
[0.0, 0.0, 1.0, 1000.23, 124153.04, 1903.93],  
[0.0, 1.0, 0.0, 1315.46, 115816.21, 297114.46],  
[1.0, 0.0, 0.0, 0.0, 135426.92, 0.0],  
[0.0, 0.0, 1.0, 542.05, 51743.15, 0.0],  
[1.0, 0.0, 0.0, 0.0, 116983.8, 45173.06]], dtype=object)
```

```
In [11]: # train_test_split
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.1)
```

```
In [12]: len(X_train)
```

```
Out[12]: 45
```

```
In [13]: len(X_test)
```

```
Out[13]: 5
```

```
In [14]: # feature scaling
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.transform(X_test)
```

```
In [15]: # create linear model
```

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train,y_train)
```

```
Out[15]: LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.

On GitHub, the HTML representation is unable to render, please try loading this page with [nbviewer.org](#).

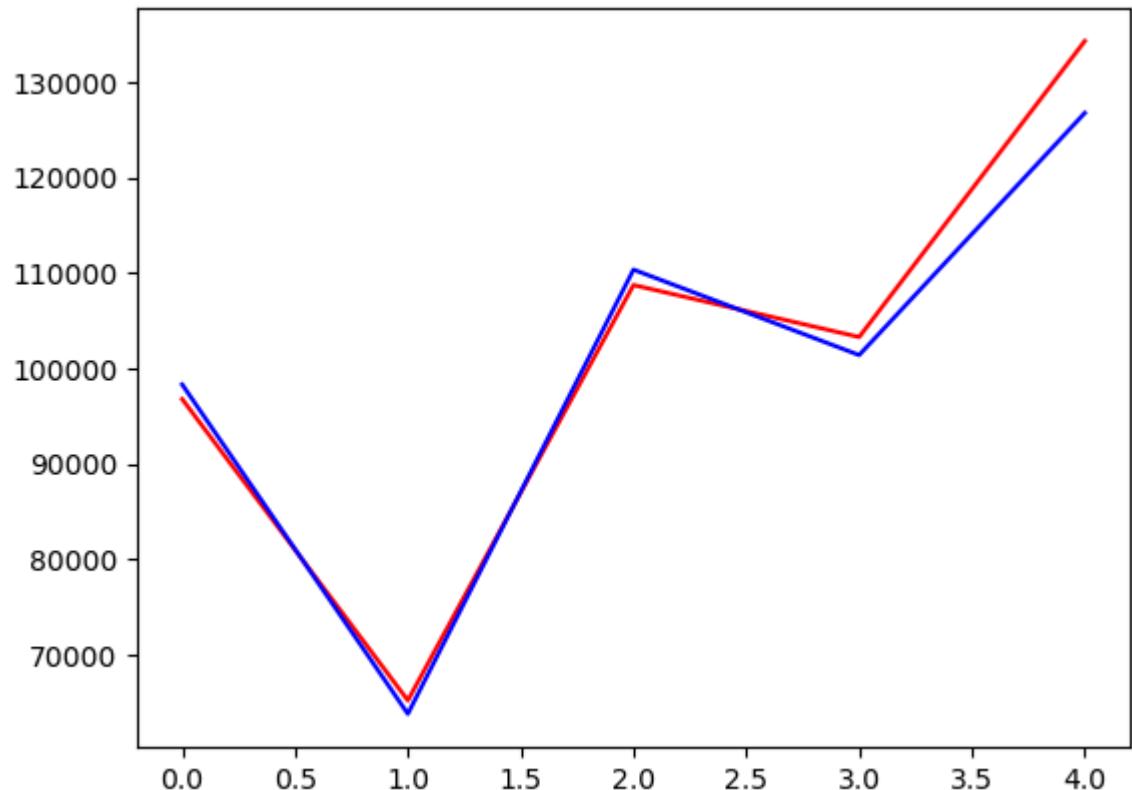
```
In [16]: # predictions
```

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

```
In [17]: # plotting graph y_test vs y_pred
```

```
plt.plot (y_test, color ='red',label='test')
plt.plot (y_pred, color ='blue',label='prediction')

plt.show()
```



```
In [18]: data=[[0.0, 0.0, 1.0, 165349.2, 136897.8, 471784.1]]
new_df=pd.DataFrame(data)
```

```
new_df=sc.transform(new_df)

single=regressor.predict(new_df)

print(single)
```

```
[192224.37606177]
```

```
In [ ]:
```

```
In [ ]:
```