In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
dataset = pd.read_csv('50_startups.csv')
```

In [3]:

```
dataset
```

Out[3]:

	R&D Spend	Administration	Marketing Spend	City	Profit
0	165349.20	136897.80	471784.10	Chennai	192261.83
1	162597.70	151377.59	443898.53	Madurai	191792.06
2	153441.51	101145.55	407934.54	Coimbatore	191050.39
3	144372.41	118671.85	383199.62	Chennai	182901.99
4	142107.34	91391.77	366168.42	Coimbatore	166187.94
5	131876.90	99814.71	362861.36	Chennai	156991.12
6	134615.46	147198.87	127716.82	Madurai	156122.51
7	130298.13	145530.06	323876.68	Coimbatore	155752.60
8	120542.52	148718.95	311613.29	Chennai	152211.77
9	123334.88	108679.17	304981.62	Madurai	149759.96
10	101913.08	110594.11	229160.95	Coimbatore	146121.95
11	100671.96	91790.61	249744.55	Madurai	144259.40
12	93863.75	127320.38	249839.44	Coimbatore	141585.52
13	91992.39	135495.07	252664.93	Madurai	134307.35
14	119943.24	156547.42	256512.92	Coimbatore	132602.65
15	114523.61	122616.84	261776.23	Chennai	129917.04
16	78013.11	121597.55	264346.06	Madurai	126992.93
17	94657.16	145077.58	282574.31	Chennai	125370.37
18	91749.16	114175.79	294919.57	Coimbatore	124266.90
19	86419.70	153514.11	0.00	Chennai	122776.86
20	76253.86	113867.30	298664.47	Madurai	118474.03
21	78389.47	153773.43	299737.29	Chennai	111313.02
22	73994.56	122782.75	303319.26	Coimbatore	110352.25
23	67532.53	105751.03	304768.73	Coimbatore	108733.99
24	77044.01	99281.34	140574.81	Chennai	108552.04
25	64664.71	139553.16	137962.62	Madurai	107404.34
26	75328.87	144135.98	134050.07	Coimbatore	105733.54
27	72107.60	127864.55	353183.81	Chennai	105008.31
28	66051.52	182645.56	118148.20	Coimbatore	103282.38
29	65605.48	153032.06	107138.38	Chennai	101004.64
30	61994.48	115641.28	91131.24	Coimbatore	99937.59
31	61136.38	152701.92	88218.23	Chennai	97483.56
32	63408.86	129219.61	46085.25	Madurai	97427.84
33	55493.95	103057.49	214634.81	Coimbatore	96778.92

	R&D Spend	Administration	Marketing Spend	City	Profit
34	46426.07	157693.92	210797.67	Madurai	96712.80
35	46014.02	85047.44	205517.64	Chennai	96479.51
36	28663.76	127056.21	201126.82	Coimbatore	90708.19
37	44069.95	51283.14	197029.42	Madurai	89949.14
38	20229.59	65947.93	185265.10	Chennai	81229.06
39	38558.51	82982.09	174999.30	Madurai	81005.76
40	28754.33	118546.05	172795.67	Madurai	78239.91
41	27892.92	84710.77	164470.71	Coimbatore	77798.83
42	23640.93	96189.63	148001.11	Madurai	71498.49
43	15505.73	127382.30	35534.17	Chennai	69758.98
44	22177.74	154806.14	28334.72	Madurai	65200.33
45	1000.23	124153.04	1903.93	Chennai	64926.08
46	1315.46	115816.21	297114.46	Coimbatore	49490.75
47	0.00	135426.92	0.00	Madurai	42559.73
48	542.05	51743.15	0.00	Chennai	35673.41
49	0.00	116983.80	45173.06	Madurai	14681.40

In [4]:

```
dataset.isnull().sum()
```

Out[4]:

```
R&D Spend          0
Administration     0
Marketing Spend    0
City               0
Profit             0
dtype: int64
```

In [5]:

```
#dependent & independent variable
x =dataset.iloc[:,:-1].values
y= dataset.iloc[:,4].values
```

In [6]:

```
# convert categorical to numerical
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.compose import ColumnTransformer
le_x = LabelEncoder()
x[:,3] = le_x.fit_transform(x[:,3])
ohe_x = ColumnTransformer(
    [('one_hot_encoder', OneHotEncoder(),[3])],
    remainder = 'passthrough'
)
x = np.array(ohe_x.fit_transform(x), dtype = np.int)
```

In [7]:

```
x
```

Out[7]:

```
array([[     1,      0,      0, 165349, 136897, 471784],
       [     0,      0,      1, 162597, 151377, 443898],
       [     0,      1,      0, 153441, 101145, 407934],
       [     1,      0,      0, 144372, 118671, 383199],
       [     0,      1,      0, 142107,  91391, 366168],
       [     1,      0,      0, 131876,  99814, 362861],
       [     0,      0,      1, 134615, 147198, 127716],
       [     0,      1,      0, 130298, 145530, 323876],
       [     1,      0,      0, 120542, 148718, 311613],
       [     0,      0,      1, 123334, 108679, 304981],
       [     0,      1,      0, 101913, 110594, 229160],
       [     0,      0,      1, 100671,  91790, 249744],
       [     0,      1,      0,  93863, 127320, 249839],
       [     0,      0,      1,  91992, 135495, 252664],
       [     0,      1,      0, 119943, 156547, 256512],
       [     1,      0,      0, 114523, 122616, 261776],
       [     0,      0,      1,  78013, 121597, 264346],
       [     1,      0,      0,  94657, 145077, 282574],
       [     0,      1,      0,  91749, 114175, 294919],
       [     1,      0,      0,  86419, 153514,      0],
       [     0,      0,      1,  76253, 113867, 298664],
       [     1,      0,      0,  78389, 153773, 299737],
       [     0,      1,      0,  73994, 122782, 303319],
       [     0,      1,      0,  67532, 105751, 304768],
       [     1,      0,      0,  77044,  99281, 140574],
       [     0,      0,      1,  64664, 139553, 137962],
       [     0,      1,      0,  75328, 144135, 134050],
       [     1,      0,      0,  72107, 127864, 353183],
       [     0,      1,      0,  66051, 182645, 118148],
       [     1,      0,      0,  65605, 153032, 107138],
       [     0,      1,      0,  61994, 115641,  91131],
       [     1,      0,      0,  61136, 152701,  88218],
       [     0,      0,      1,  63408, 129219,  46085],
       [     0,      1,      0,  55493, 103057, 214634],
       [     0,      0,      1,  46426, 157693, 210797],
       [     1,      0,      0,  46014,  85047, 205517],
       [     0,      1,      0,  28663, 127056, 201126],
       [     0,      0,      1,  44069,  51283, 197029],
       [     1,      0,      0,  20229,  65947, 185265],
       [     0,      0,      1,  38558,  82982, 174999],
       [     0,      0,      1,  28754, 118546, 172795],
       [     0,      1,      0,  27892,  84710, 164470],
       [     0,      0,      1,  23640,  96189, 148001],
       [     1,      0,      0,  15505, 127382,  35534],
       [     0,      0,      1,  22177, 154806,  28334],
       [     1,      0,      0,   1000, 124153,   1903],
       [     0,      1,      0,   1315, 115816, 297114],
       [     0,      0,      1,      0, 135426,      0],
       [     1,      0,      0,    542,  51743,      0],
       [     0,      0,      1,      0, 116983,  45173]])
```

In [8]:

```
#dummy variable trap
x = x[:,1:]
```

In [9]:

```
x
```

Out[9]:

```
array([[     0,      0, 165349, 136897, 471784],
       [     0,      1, 162597, 151377, 443898],
       [     1,      0, 153441, 101145, 407934],
       [     0,      0, 144372, 118671, 383199],
       [     1,      0, 142107,  91391, 366168],
       [     0,      0, 131876,  99814, 362861],
       [     0,      1, 134615, 147198, 127716],
       [     1,      0, 130298, 145530, 323876],
       [     0,      0, 120542, 148718, 311613],
       [     0,      1, 123334, 108679, 304981],
       [     1,      0, 101913, 110594, 229160],
       [     0,      1, 100671,  91790, 249744],
       [     1,      0,  93863, 127320, 249839],
       [     0,      1,  91992, 135495, 252664],
       [     1,      0, 119943, 156547, 256512],
       [     0,      0, 114523, 122616, 261776],
       [     0,      1,  78013, 121597, 264346],
       [     0,      0,  94657, 145077, 282574],
       [     1,      0,  91749, 114175, 294919],
       [     0,      0,  86419, 153514,      0],
       [     0,      1,  76253, 113867, 298664],
       [     0,      0,  78389, 153773, 299737],
       [     1,      0,  73994, 122782, 303319],
       [     1,      0,  67532, 105751, 304768],
       [     0,      0,  77044,  99281, 140574],
       [     0,      1,  64664, 139553, 137962],
       [     1,      0,  75328, 144135, 134050],
       [     0,      0,  72107, 127864, 353183],
       [     1,      0,  66051, 182645, 118148],
       [     0,      0,  65605, 153032, 107138],
       [     1,      0,  61994, 115641,  91131],
       [     0,      0,  61136, 152701,  88218],
       [     0,      1,  63408, 129219,  46085],
       [     1,      0,  55493, 103057, 214634],
       [     0,      1,  46426, 157693, 210797],
       [     0,      0,  46014,  85047, 205517],
       [     1,      0,  28663, 127056, 201126],
       [     0,      1,  44069,  51283, 197029],
       [     0,      0,  20229,  65947, 185265],
       [     0,      1,  38558,  82982, 174999],
       [     0,      1,  28754, 118546, 172795],
       [     1,      0,  27892,  84710, 164470],
       [     0,      1,  23640,  96189, 148001],
       [     0,      0,  15505, 127382,  35534],
       [     0,      1,  22177, 154806,  28334],
       [     0,      0,   1000, 124153,   1903],
       [     1,      0,   1315, 115816, 297114],
       [     0,      1,      0, 135426,      0],
       [     0,      0,    542,  51743,      0],
       [     0,      1,      0, 116983,  45173]])
```

In [10]:

```
#split and train data

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=.30,random_state = 0)
```

In [11]:

```
# MLR model from sklearn
from sklearn.linear_model import LinearRegression
regress = LinearRegression()
regress.fit(x_train, y_train)
```

Out[11]:

```
LinearRegression()
```

In [12]:

```
#prediction
y_pred = regress.predict(x_test)
```

In [13]:

```
y_pred
```

Out[13]:

```
array([104282.74845358, 132536.53849927, 133911.11921423,  72584.35572911,
       179920.88369286, 114549.21392778,  66444.25735029,  98404.98885339,
       114499.77312619, 169367.57489703,  96522.21039595,  88040.95894935,
       110949.65365898,  90419.61586831, 128020.69204362])
```

In [14]:

```
y_test
```

Out[14]:

```
array([103282.38, 144259.4 , 146121.95,  77798.83, 191050.39, 105008.31,
        81229.06,  97483.56, 110352.25, 166187.94,  96778.92,  96479.51,
       105733.54,  96712.8 , 124266.9 ])
```

In [15]:

```
#model performance - Best fitted line error rate
regress.score(x_train, y_train)
```

Out[15]:

```
0.9515563254725671
```

In [16]:

```
regress.score(x_test, y_test)
```

Out[16]:

```
0.9358682078683178
```

In [17]:

```
# lets build the model using backward elimination
# Now we will use linear regression from statsmodel library
import statsmodels.api as sm
```

In [18]:

```
x = np.append(arr = np.ones(shape = (50,1), dtype = int), values = x, axis =1) # x =c+x
#x = np.append(arr =x , values = np.ones(shape = (50,1), dtype = int), axis =1) # x =x+c
```

In [19]:

```
x
```

Out[19]:

```
array([[     1,      0,      0, 165349, 136897, 471784],
       [     1,      0,      1, 162597, 151377, 443898],
       [     1,      1,      0, 153441, 101145, 407934],
       [     1,      0,      0, 144372, 118671, 383199],
       [     1,      1,      0, 142107,  91391, 366168],
       [     1,      0,      0, 131876,  99814, 362861],
       [     1,      0,      1, 134615, 147198, 127716],
       [     1,      1,      0, 130298, 145530, 323876],
       [     1,      0,      0, 120542, 148718, 311613],
       [     1,      0,      1, 123334, 108679, 304981],
       [     1,      1,      0, 101913, 110594, 229160],
       [     1,      0,      1, 100671,  91790, 249744],
       [     1,      1,      0,  93863, 127320, 249839],
       [     1,      0,      1,  91992, 135495, 252664],
       [     1,      1,      0, 119943, 156547, 256512],
       [     1,      0,      0, 114523, 122616, 261776],
       [     1,      0,      1,  78013, 121597, 264346],
       [     1,      0,      0,  94657, 145077, 282574],
       [     1,      1,      0,  91749, 114175, 294919],
       [     1,      0,      0,  86419, 153514,      0],
       [     1,      0,      1,  76253, 113867, 298664],
       [     1,      0,      0,  78389, 153773, 299737],
       [     1,      1,      0,  73994, 122782, 303319],
       [     1,      1,      0,  67532, 105751, 304768],
       [     1,      0,      0,  77044,  99281, 140574],
       [     1,      0,      1,  64664, 139553, 137962],
       [     1,      1,      0,  75328, 144135, 134050],
       [     1,      0,      0,  72107, 127864, 353183],
       [     1,      1,      0,  66051, 182645, 118148],
       [     1,      0,      0,  65605, 153032, 107138],
       [     1,      1,      0,  61994, 115641,  91131],
       [     1,      0,      0,  61136, 152701,  88218],
       [     1,      0,      1,  63408, 129219,  46085],
       [     1,      1,      0,  55493, 103057, 214634],
       [     1,      0,      1,  46426, 157693, 210797],
       [     1,      0,      0,  46014,  85047, 205517],
       [     1,      1,      0,  28663, 127056, 201126],
       [     1,      0,      1,  44069,  51283, 197029],
       [     1,      0,      0,  20229,  65947, 185265],
       [     1,      0,      1,  38558,  82982, 174999],
       [     1,      0,      1,  28754, 118546, 172795],
       [     1,      1,      0,  27892,  84710, 164470],
       [     1,      0,      1,  23640,  96189, 148001],
       [     1,      0,      0,  15505, 127382,  35534],
       [     1,      0,      1,  22177, 154806,  28334],
       [     1,      0,      0,   1000, 124153,   1903],
       [     1,      1,      0,   1315, 115816, 297114],
       [     1,      0,      1,      0, 135426,      0],
       [     1,      0,      0,    542,  51743,      0],
       [     1,      0,      1,      0, 116983,  45173]])
```

In [20]:

```
#backward elimination
x_ov = x[:,[0,1,2,3,4,5]] # x_ov = x
regress_ols = sm.OLS(endog=y, exog = x_ov).fit()
regress_ols.summary()
```

Out[20]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.951
Model:	OLS	**Adj. R-squared:**	0.945
Method:	Least Squares	**F-statistic:**	169.9
Date:	Sat, 09 Jan 2021	**Prob (F-statistic):**	1.34e-27
Time:	19:26:15	**Log-Likelihood:**	-525.38
No. Observations:	50	**AIC:**	1063.
Df Residuals:	44	**BIC:**	1074.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	5.008e+04	6952.617	7.204	0.000	3.61e+04	6.41e+04
x1	240.7605	3338.877	0.072	0.943	-6488.304	6969.825
x2	42.0063	3256.058	0.013	0.990	-6520.148	6604.161
x3	0.8060	0.046	17.368	0.000	0.712	0.900
x4	-0.0270	0.052	-0.517	0.608	-0.132	0.078
x5	0.0270	0.017	1.574	0.123	-0.008	0.062

Omnibus:	14.783	**Durbin-Watson:**	1.283
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	21.267
Skew:	-0.948	**Prob(JB):**	2.41e-05
Kurtosis:	5.572	**Cond. No.**	1.47e+06

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.47e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

In [21]:

```
x_ov
```

Out[21]:

```
array([[     1,      0,      0, 165349, 136897, 471784],
       [     1,      0,      1, 162597, 151377, 443898],
       [     1,      1,      0, 153441, 101145, 407934],
       [     1,      0,      0, 144372, 118671, 383199],
       [     1,      1,      0, 142107,  91391, 366168],
       [     1,      0,      0, 131876,  99814, 362861],
       [     1,      0,      1, 134615, 147198, 127716],
       [     1,      1,      0, 130298, 145530, 323876],
       [     1,      0,      0, 120542, 148718, 311613],
       [     1,      0,      1, 123334, 108679, 304981],
       [     1,      1,      0, 101913, 110594, 229160],
       [     1,      0,      1, 100671,  91790, 249744],
       [     1,      1,      0,  93863, 127320, 249839],
       [     1,      0,      1,  91992, 135495, 252664],
       [     1,      1,      0, 119943, 156547, 256512],
       [     1,      0,      0, 114523, 122616, 261776],
       [     1,      0,      1,  78013, 121597, 264346],
       [     1,      0,      0,  94657, 145077, 282574],
       [     1,      1,      0,  91749, 114175, 294919],
       [     1,      0,      0,  86419, 153514,      0],
       [     1,      0,      1,  76253, 113867, 298664],
       [     1,      0,      0,  78389, 153773, 299737],
       [     1,      1,      0,  73994, 122782, 303319],
       [     1,      1,      0,  67532, 105751, 304768],
       [     1,      0,      0,  77044,  99281, 140574],
       [     1,      0,      1,  64664, 139553, 137962],
       [     1,      1,      0,  75328, 144135, 134050],
       [     1,      0,      0,  72107, 127864, 353183],
       [     1,      1,      0,  66051, 182645, 118148],
       [     1,      0,      0,  65605, 153032, 107138],
       [     1,      1,      0,  61994, 115641,  91131],
       [     1,      0,      0,  61136, 152701,  88218],
       [     1,      0,      1,  63408, 129219,  46085],
       [     1,      1,      0,  55493, 103057, 214634],
       [     1,      0,      1,  46426, 157693, 210797],
       [     1,      0,      0,  46014,  85047, 205517],
       [     1,      1,      0,  28663, 127056, 201126],
       [     1,      0,      1,  44069,  51283, 197029],
       [     1,      0,      0,  20229,  65947, 185265],
       [     1,      0,      1,  38558,  82982, 174999],
       [     1,      0,      1,  28754, 118546, 172795],
       [     1,      1,      0,  27892,  84710, 164470],
       [     1,      0,      1,  23640,  96189, 148001],
       [     1,      0,      0,  15505, 127382,  35534],
       [     1,      0,      1,  22177, 154806,  28334],
       [     1,      0,      0,   1000, 124153,   1903],
       [     1,      1,      0,   1315, 115816, 297114],
       [     1,      0,      1,      0, 135426,      0],
       [     1,      0,      0,    542,  51743,      0],
       [     1,      0,      1,      0, 116983,  45173]])
```

In [22]:

```
x_ovc = x_ov[:,1:]
```

In [23]:

```
#split and train data

from sklearn.model_selection import train_test_split
x_ov_train, x_ov_test, y_ov_train, y_ov_test = train_test_split(x_ov,y,test_size=.30,random

# MLR model from sklearn
from sklearn.linear_model import LinearRegression
regress_ov1 = LinearRegression()
regress_ov1.fit(x_ov_train, y_ov_train)
```

Out[23]:

```
LinearRegression()
```

In [24]:

```
regress_ov1.score(x_ov_train, y_ov_train)
```

Out[24]:

```
0.9515563254725671
```

In [25]:

```
regress_ov1.score(x_ov_test, y_ov_test)
```

Out[25]:

```
0.9358682078683345
```

In [26]:

```
#iteration 2
x_ov = x[:,[0,1,3,4,5]]
regress_ols = sm.OLS(endog=y, exog = x_ov).fit()
regress_ols.summary()
```

Out[26]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.951
Model:	OLS	**Adj. R-squared:**	0.946
Method:	Least Squares	**F-statistic:**	217.2
Date:	Sat, 09 Jan 2021	**Prob (F-statistic):**	8.49e-29
Time:	19:27:29	**Log-Likelihood:**	-525.38
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]
const	5.011e+04	6647.901	7.537	0.000	3.67e+04	6.35e+04
x1	220.1847	2900.553	0.076	0.940	-5621.828	6062.197
x2	0.8060	0.046	17.606	0.000	0.714	0.898
x3	-0.0270	0.052	-0.523	0.604	-0.131	0.077
x4	0.0270	0.017	1.592	0.118	-0.007	0.061

Omnibus:	14.759	**Durbin-Watson:**	1.282
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	21.173
Skew:	-0.948	**Prob(JB):**	2.53e-05
Kurtosis:	5.563	**Cond. No.**	1.40e+06

Warnings:
[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.

In [27]:

```
x_ovc = x_ov[:,1:]
x_ov_train, x_ov_test, y_ov_train, y_ov_test = train_test_split(x_ov,y,test_size=.30,random

# MLR model from sklearn
from sklearn.linear_model import LinearRegression
regress_ov2 = LinearRegression()
regress_ov2.fit(x_ov_train, y_ov_train)
```

Out[27]:

```
LinearRegression()
```

In [28]:

```
regress_ov2.score(x_ov_train, y_ov_train)
```

Out[28]:

```
0.9515467487890941
```

In [29]:

```
regress_ov2.score(x_ov_test, y_ov_test)
```

Out[29]:

```
0.935885414206326
```

In [30]:

```
#iteration 3
x_ov = x[:,[0,3,4,5]]
regress_ols = sm.OLS(endog=y, exog = x_ov).fit()
regress_ols.summary()
```

Out[30]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.951
Model:	OLS	**Adj. R-squared:**	0.948
Method:	Least Squares	**F-statistic:**	296.0
Date:	Sat, 09 Jan 2021	**Prob (F-statistic):**	4.53e-30
Time:	19:28:19	**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]
const	5.012e+04	6572.384	7.626	0.000	3.69e+04	6.34e+04
x1	0.8057	0.045	17.846	0.000	0.715	0.897
x2	-0.0268	0.051	-0.526	0.602	-0.130	0.076
x3	0.0272	0.016	1.655	0.105	-0.006	0.060

Omnibus:	14.839	**Durbin-Watson:**	1.282
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	21.443
Skew:	-0.949	**Prob(JB):**	2.21e-05
Kurtosis:	5.587	**Cond. No.**	1.40e+06

Warnings:
[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.

In [31]:

```
x_ovc = x_ov[:,1:]
x_ov_train, x_ov_test, y_ov_train, y_ov_test = train_test_split(x_ov,y,test_size=.30,random

# MLR model from sklearn
from sklearn.linear_model import LinearRegression
regress_ov3 = LinearRegression()
regress_ov3.fit(x_ov_train, y_ov_train)
```

Out[31]:

```
LinearRegression()
```

In [32]:

```
regress_ov3.score(x_ov_train, y_ov_train)
```

Out[32]:

```
0.9515383544673244
```

In [33]:

```
regress_ov3.score(x_ov_test, y_ov_test)
```

Out[33]:

```
0.9355189542547092
```

In [35]:

```
x_ov
```

Out[35]:

```
array([[     1, 165349, 136897, 471784],
       [     1, 162597, 151377, 443898],
       [     1, 153441, 101145, 407934],
       [     1, 144372, 118671, 383199],
       [     1, 142107,  91391, 366168],
       [     1, 131876,  99814, 362861],
       [     1, 134615, 147198, 127716],
       [     1, 130298, 145530, 323876],
       [     1, 120542, 148718, 311613],
       [     1, 123334, 108679, 304981],
       [     1, 101913, 110594, 229160],
       [     1, 100671,  91790, 249744],
       [     1,  93863, 127320, 249839],
       [     1,  91992, 135495, 252664],
       [     1, 119943, 156547, 256512],
       [     1, 114523, 122616, 261776],
       [     1,  78013, 121597, 264346],
       [     1,  94657, 145077, 282574],
       [     1,  91749, 114175, 294919],
       [     1,  86419, 153514,      0],
       [     1,  76253, 113867, 298664],
       [     1,  78389, 153773, 299737],
       [     1,  73994, 122782, 303319],
       [     1,  67532, 105751, 304768],
       [     1,  77044,  99281, 140574],
       [     1,  64664, 139553, 137962],
       [     1,  75328, 144135, 134050],
       [     1,  72107, 127864, 353183],
       [     1,  66051, 182645, 118148],
       [     1,  65605, 153032, 107138],
       [     1,  61994, 115641,  91131],
       [     1,  61136, 152701,  88218],
       [     1,  63408, 129219,  46085],
       [     1,  55493, 103057, 214634],
       [     1,  46426, 157693, 210797],
       [     1,  46014,  85047, 205517],
       [     1,  28663, 127056, 201126],
       [     1,  44069,  51283, 197029],
       [     1,  20229,  65947, 185265],
       [     1,  38558,  82982, 174999],
       [     1,  28754, 118546, 172795],
       [     1,  27892,  84710, 164470],
       [     1,  23640,  96189, 148001],
       [     1,  15505, 127382,  35534],
       [     1,  22177, 154806,  28334],
       [     1,   1000, 124153,   1903],
       [     1,   1315, 115816, 297114],
       [     1,      0, 135426,      0],
       [     1,    542,  51743,      0],
       [     1,      0, 116983,  45173]])
```

In [36]:

```
#iteration 4
x_ov = x[:,[0,3,5]]
regress_ols = sm.OLS(endog=y, exog = x_ov).fit()
regress_ols.summary()
```

Out[36]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.950
Model:	OLS	**Adj. R-squared:**	0.948
Method:	Least Squares	**F-statistic:**	450.8
Date:	Sat, 09 Jan 2021	**Prob (F-statistic):**	2.16e-31
Time:	19:29:47	**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]**
const	4.698e+04	2689.941	17.464	0.000	4.16e+04	5.24e+04
x1	0.7966	0.041	19.265	0.000	0.713	0.880
x2	0.0299	0.016	1.927	0.060	-0.001	0.061

Omnibus:	14.678	**Durbin-Watson:**	1.257
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	21.162
Skew:	-0.939	**Prob(JB):**	2.54e-05
Kurtosis:	5.575	**Cond. No.**	5.32e+05

Warnings:
[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.

In [37]:

```
x_ovc = x_ov[:,1:]
x_ov_train, x_ov_test, y_ov_train, y_ov_test = train_test_split(x_ov,y,test_size=.30,random

# MLR model from sklearn
from sklearn.linear_model import LinearRegression
regress_ov4 = LinearRegression()
regress_ov4.fit(x_ov_train, y_ov_train)
```

Out[37]:

```
LinearRegression()
```

In [38]:

```
regress_ov4.score(x_ov_train, y_ov_train)
```

Out[38]:

```
0.9512556961080693
```

In [39]:

```
regress_ov4.score(x_ov_test, y_ov_test)
```

Out[39]:

```
0.9431309583591085
```

In [40]:

```
#iteration 5
x_ov = x[:,[0,3]]
regress_ols = sm.OLS(endog=y, exog = x_ov).fit()
regress_ols.summary()
```

Out[40]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.947
Model:	OLS	**Adj. R-squared:**	0.945
Method:	Least Squares	**F-statistic:**	849.8
Date:	Sat, 09 Jan 2021	**Prob (F-statistic):**	3.50e-32
Time:	19:30:32	**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]**
const	4.903e+04	2537.900	19.320	0.000	4.39e+04	5.41e+04
x1	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):**	18.538
Skew:	-0.911	**Prob(JB):**	9.43e-05
Kurtosis:	5.361	**Cond. No.**	1.65e+05

Warnings:
[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.

In [41]:

```
x_ovc = x_ov[:,1:]
x_ov_train, x_ov_test, y_ov_train, y_ov_test = train_test_split(x_ov,y,test_size=.30,random

# MLR model from sklearn
from sklearn.linear_model import LinearRegression
regress_ov5 = LinearRegression()
regress_ov5.fit(x_ov_train, y_ov_train)
```

Out[41]:

```
LinearRegression()
```

In [42]:

```
regress_ov5.score(x_ov_train, y_ov_train)
```

Out[42]:

```
0.9476436934242192
```

In [43]:

```
regress_ov5.score(x_ov_test, y_ov_test)
```

Out[43]:

```
0.9360403445633265
```

In [ ]: