PROJECT 1

Data Preprocessing of the dataset 'investment_data'

NAME- ANURAG MISHRA SIC- 20BCED17

Step1 : Importing the libraries

In [42]:

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

Step 2: Import data set

In [43]:

```
dataset=pd.read_csv('investment_data.csv')
```

In [44]:

```
dataset
```

Out[44]:

	Capital investment	Employee salary	Advertisement expenditure	City	Turn over
0	165361.76	136897.80	471784.10	Kolkata	192274.39
1	162610.26	151377.59	443898.53	Bengaluru	191804.62
2	153454.07	101145.55	407934.54	Chennai	191062.95
3	144384.97	118671.85	383199.62	Kolkata	182914.55
4	142119.90	91391.77	366168.42	Chennai	166200.50
5	131889.46	99814.71	362861.36	Kolkata	157003.68
6	134628.02	147198.87	127716.82	Bengaluru	156135.07
7	130310.69	145530.06	323876.68	Chennai	155765.16
8	120555.08	148718.95	311613.29	Kolkata	152224.33
9	123347.44	108679.17	304981.62	Bengaluru	149772.52
10	101925.64	110594.11	229160.95	Chennai	146134.51
11	100684.52	91790.61	249744.55	Bengaluru	144271.96
12	93876.31	127320.38	249839.44	Chennai	141598.08
13	92004.95	135495.07	252664.93	Bengaluru	134319.91
14	119955.80	156547.42	256512.92	Chennai	132615.21
15	114536.17	122616.84	261776.23	Kolkata	129929.60
16	78025.67	121597.55	264346.06	Bengaluru	127005.49
17	94669.72	145077.58	282574.31	Kolkata	125382.93
18	91761.72	114175.79	294919.57	Chennai	124279.46
19	86432.26	153514.11	0.00	Kolkata	122789.42
20	76266.42	113867.30	298664.47	Bengaluru	118486.59

	Capital investment	Employee salary	Advertisement expenditure	City	Turn over
21	78402.03	153773.43	299737.29	Kolkata	111325.58
22	74007.12	122782.75	303319.26	Chennai	110364.81
23	67545.09	105751.03	304768.73	Chennai	108746.55
24	77056.57	99281.34	140574.81	Kolkata	108564.60
25	64677.27	139553.16	137962.62	Bengaluru	107416.90
26	75341.43	144135.98	134050.07	Chennai	105746.10
27	72120.16	127864.55	353183.81	Kolkata	105020.87
28	66064.08	182645.56	118148.20	Chennai	103294.94
29	65618.04	153032.06	107138.38	Kolkata	101017.20
30	62007.04	115641.28	91131.24	Chennai	99950.15
31	61148.94	152701.92	88218.23	Kolkata	97496.12
32	63421.42	129219.61	46085.25	Bengaluru	97440.40
33	55506.51	103057.49	214634.81	Chennai	96791.48
34	46438.63	157693.92	210797.67	Bengaluru	96725.36
35	46026.58	85047.44	205517.64	Kolkata	96492.07
36	28676.32	127056.21	201126.82	Chennai	90720.75
37	44082.51	51283.14	197029.42	Bengaluru	89961.70
38	20242.15	65947.93	185265.10	Kolkata	81241.62
39	38571.07	82982.09	174999.30	Bengaluru	81018.32
40	28766.89	118546.05	172795.67	Bengaluru	78252.47
41	27905.48	84710.77	164470.71	Chennai	77811.39
42	23653.49	96189.63	148001.11	Bengaluru	71511.05
43	15518.29	127382.30	35534.17	Kolkata	69771.54
44	22190.30	154806.14	28334.72	Bengaluru	65212.89
45	1012.79	124153.04	1903.93	Kolkata	64938.64
46	1328.02	115816.21	297114.46	Chennai	49503.31
47	12.56	135426.92	0.00	Bengaluru	42572.29
48	554.61	51743.15	0.00	Kolkata	35685.97
49	12.56	116983.80	45173.06	Bengaluru	14693.96

Step3: To create feature matrix and dependent variable vector

In [45]:

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

In [46]:

```
x
```

Out[46]:

```
array([[165361.76, 136897.8, 471784.1, 'Kolkata'],
       [162610.26, 151377.59, 443898.53, 'Bengaluru'],
       [153454.07, 101145.55, 407934.54, 'Chennai'],
       [144384.97, 118671.85, 383199.62, 'Kolkata'],
       [142119.9, 91391.77, 366168.42, 'Chennai'],
       [131889.46, 99814.71, 362861.36, 'Kolkata'],
       [134628.02, 147198.87, 127716.82, 'Bengaluru'],
       [130310.69, 145530.06, 323876.68, 'Chennai'],
```

```
       [120555.08, 148718.95, 311613.29, 'Kolkata'],
       [123347.44, 108679.17, 304981.62, 'Bengaluru'],
       [101925.64, 110594.11, 229160.95, 'Chennai'],
       [100684.52, 91790.61, 249744.55, 'Bengaluru'],
       [93876.31, 127320.38, 249839.44, 'Chennai'],
       [92004.95, 135495.07, 252664.93, 'Bengaluru'],
       [119955.8, 156547.42, 256512.92, 'Chennai'],
       [114536.17, 122616.84, 261776.23, 'Kolkata'],
       [78025.67, 121597.55, 264346.06, 'Bengaluru'],
       [94669.72, 145077.58, 282574.31, 'Kolkata'],
       [91761.72, 114175.79, 294919.57, 'Chennai'],
       [86432.26, 153514.11, 0.0, 'Kolkata'],
       [76266.42, 113867.3, 298664.47, 'Bengaluru'],
       [78402.03, 153773.43, 299737.29, 'Kolkata'],
       [74007.12, 122782.75, 303319.26, 'Chennai'],
       [67545.09, 105751.03, 304768.73, 'Chennai'],
       [77056.57, 99281.34, 140574.81, 'Kolkata'],
       [64677.27, 139553.16, 137962.62, 'Bengaluru'],
       [75341.43, 144135.98, 134050.07, 'Chennai'],
       [72120.16, 127864.55, 353183.81, 'Kolkata'],
       [66064.08, 182645.56, 118148.2, 'Chennai'],
       [65618.04, 153032.06, 107138.38, 'Kolkata'],
       [62007.04, 115641.28, 91131.24, 'Chennai'],
       [61148.94, 152701.92, 88218.23, 'Kolkata'],
       [63421.42, 129219.61, 46085.25, 'Bengaluru'],
       [55506.51, 103057.49, 214634.81, 'Chennai'],
       [46438.63, 157693.92, 210797.67, 'Bengaluru'],
       [46026.58, 85047.44, 205517.64, 'Kolkata'],
       [28676.32, 127056.21, 201126.82, 'Chennai'],
       [44082.51, 51283.14, 197029.42, 'Bengaluru'],
       [20242.15, 65947.93, 185265.1, 'Kolkata'],
       [38571.07, 82982.09, 174999.3, 'Bengaluru'],
       [28766.89, 118546.05, 172795.67, 'Bengaluru'],
       [27905.48, 84710.77, 164470.71, 'Chennai'],
       [23653.49, 96189.63, 148001.11, 'Bengaluru'],
       [15518.29, 127382.3, 35534.17, 'Kolkata'],
       [22190.3, 154806.14, 28334.72, 'Bengaluru'],
       [1012.79, 124153.04, 1903.93, 'Kolkata'],
       [1328.02, 115816.21, 297114.46, 'Chennai'],
       [12.56, 135426.92, 0.0, 'Bengaluru'],
       [554.61, 51743.15, 0.0, 'Kolkata'],
       [12.56, 116983.8, 45173.06, 'Bengaluru']], dtype=object)
```

In [47]:

```
y
```

Out[47]:

```
array([192274.39, 191804.62, 191062.95, 182914.55, 166200.5 , 157003.68,
       156135.07, 155765.16, 152224.33, 149772.52, 146134.51, 144271.96,
       141598.08, 134319.91, 132615.21, 129929.6 , 127005.49, 125382.93,
       124279.46, 122789.42, 118486.59, 111325.58, 110364.81, 108746.55,
       108564.6 , 107416.9 , 105746.1 , 105020.87, 103294.94, 101017.2 ,
        99950.15,  97496.12,  97440.4 ,  96791.48,  96725.36,  96492.07,
        90720.75,  89961.7 ,  81241.62,  81018.32,  78252.47,  77811.39,
        71511.05,  69771.54,  65212.89,  64938.64,  49503.31,  42572.29,
        35685.97,  14693.96])
```

Step4: Replace missing data

In [48]:

```
from sklearn.impute import SimpleImputer
imputer=SimpleImputer(missing_values=np.nan,strategy='mean')
imputer.fit(x[:,:3])
x[:,:3]=imputer.transform(x[:,:3])
```

In [49]:

```
x
```

Out[49]:

```
array([[165361.76, 136897.8, 471784.1, 'Kolkata'],
       [162610.26, 151377.59, 443898.53, 'Bengaluru'],
       [153454.07, 101145.55, 407934.54, 'Chennai'],
       [144384.97, 118671.85, 383199.62, 'Kolkata'],
       [142119.9, 91391.77, 366168.42, 'Chennai'],
       [131889.46, 99814.71, 362861.36, 'Kolkata'],
       [134628.02, 147198.87, 127716.82, 'Bengaluru'],
       [130310.69, 145530.06, 323876.68, 'Chennai'],
       [120555.08, 148718.95, 311613.29, 'Kolkata'],
       [123347.44, 108679.17, 304981.62, 'Bengaluru'],
       [101925.64, 110594.11, 229160.95, 'Chennai'],
       [100684.52, 91790.61, 249744.55, 'Bengaluru'],
       [93876.31, 127320.38, 249839.44, 'Chennai'],
       [92004.95, 135495.07, 252664.93, 'Bengaluru'],
       [119955.8, 156547.42, 256512.92, 'Chennai'],
       [114536.17, 122616.84, 261776.23, 'Kolkata'],
       [78025.67, 121597.55, 264346.06, 'Bengaluru'],
       [94669.72, 145077.58, 282574.31, 'Kolkata'],
       [91761.72, 114175.79, 294919.57, 'Chennai'],
       [86432.26, 153514.11, 0.0, 'Kolkata'],
       [76266.42, 113867.3, 298664.47, 'Bengaluru'],
       [78402.03, 153773.43, 299737.29, 'Kolkata'],
       [74007.12, 122782.75, 303319.26, 'Chennai'],
       [67545.09, 105751.03, 304768.73, 'Chennai'],
       [77056.57, 99281.34, 140574.81, 'Kolkata'],
       [64677.27, 139553.16, 137962.62, 'Bengaluru'],
       [75341.43, 144135.98, 134050.07, 'Chennai'],
       [72120.16, 127864.55, 353183.81, 'Kolkata'],
       [66064.08, 182645.56, 118148.2, 'Chennai'],
       [65618.04, 153032.06, 107138.38, 'Kolkata'],
       [62007.04, 115641.28, 91131.24, 'Chennai'],
       [61148.94, 152701.92, 88218.23, 'Kolkata'],
       [63421.42, 129219.61, 46085.25, 'Bengaluru'],
       [55506.51, 103057.49, 214634.81, 'Chennai'],
       [46438.63, 157693.92, 210797.67, 'Bengaluru'],
       [46026.58, 85047.44, 205517.64, 'Kolkata'],
       [28676.32, 127056.21, 201126.82, 'Chennai'],
       [44082.51, 51283.14, 197029.42, 'Bengaluru'],
       [20242.15, 65947.93, 185265.1, 'Kolkata'],
       [38571.07, 82982.09, 174999.3, 'Bengaluru'],
       [28766.89, 118546.05, 172795.67, 'Bengaluru'],
       [27905.48, 84710.77, 164470.71, 'Chennai'],
       [23653.49, 96189.63, 148001.11, 'Bengaluru'],
       [15518.29, 127382.3, 35534.17, 'Kolkata'],
       [22190.3, 154806.14, 28334.72, 'Bengaluru'],
       [1012.79, 124153.04, 1903.93, 'Kolkata'],
       [1328.02, 115816.21, 297114.46, 'Chennai'],
       [12.56, 135426.92, 0.0, 'Bengaluru'],
       [554.61, 51743.15, 0.0, 'Kolkata'],
       [12.56, 116983.8, 45173.06, 'Bengaluru']], dtype=object)
```

Step5: Encoding

Feature matrix using OneHotEncoding

In [50]:

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct=ColumnTransformer(transformers=[('encoder',OneHotEncoder(),[3])],remainder='passthroug
h')
x=np.array(ct.fit_transform(x))
```

In [51]:

```
x
```

Out[51]:

```
array([[0.0, 0.0, 1.0, 165361.76, 136897.8, 471784.1],
       [1.0, 0.0, 0.0, 162610.26, 151377.59, 443898.53],
       [0.0, 1.0, 0.0, 153454.07, 101145.55, 407934.54],
       [0.0, 0.0, 1.0, 144384.97, 118671.85, 383199.62],
       [0.0, 1.0, 0.0, 142119.9, 91391.77, 366168.42],
       [0.0, 0.0, 1.0, 131889.46, 99814.71, 362861.36],
       [1.0, 0.0, 0.0, 134628.02, 147198.87, 127716.82],
       [0.0, 1.0, 0.0, 130310.69, 145530.06, 323876.68],
       [0.0, 0.0, 1.0, 120555.08, 148718.95, 311613.29],
       [1.0, 0.0, 0.0, 123347.44, 108679.17, 304981.62],
       [0.0, 1.0, 0.0, 101925.64, 110594.11, 229160.95],
       [1.0, 0.0, 0.0, 100684.52, 91790.61, 249744.55],
       [0.0, 1.0, 0.0, 93876.31, 127320.38, 249839.44],
       [1.0, 0.0, 0.0, 92004.95, 135495.07, 252664.93],
       [0.0, 1.0, 0.0, 119955.8, 156547.42, 256512.92],
       [0.0, 0.0, 1.0, 114536.17, 122616.84, 261776.23],
       [1.0, 0.0, 0.0, 78025.67, 121597.55, 264346.06],
       [0.0, 0.0, 1.0, 94669.72, 145077.58, 282574.31],
       [0.0, 1.0, 0.0, 91761.72, 114175.79, 294919.57],
       [0.0, 0.0, 1.0, 86432.26, 153514.11, 0.0],
       [1.0, 0.0, 0.0, 76266.42, 113867.3, 298664.47],
       [0.0, 0.0, 1.0, 78402.03, 153773.43, 299737.29],
       [0.0, 1.0, 0.0, 74007.12, 122782.75, 303319.26],
       [0.0, 1.0, 0.0, 67545.09, 105751.03, 304768.73],
       [0.0, 0.0, 1.0, 77056.57, 99281.34, 140574.81],
       [1.0, 0.0, 0.0, 64677.27, 139553.16, 137962.62],
       [0.0, 1.0, 0.0, 75341.43, 144135.98, 134050.07],
       [0.0, 0.0, 1.0, 72120.16, 127864.55, 353183.81],
       [0.0, 1.0, 0.0, 66064.08, 182645.56, 118148.2],
       [0.0, 0.0, 1.0, 65618.04, 153032.06, 107138.38],
       [0.0, 1.0, 0.0, 62007.04, 115641.28, 91131.24],
       [0.0, 0.0, 1.0, 61148.94, 152701.92, 88218.23],
       [1.0, 0.0, 0.0, 63421.42, 129219.61, 46085.25],
       [0.0, 1.0, 0.0, 55506.51, 103057.49, 214634.81],
       [1.0, 0.0, 0.0, 46438.63, 157693.92, 210797.67],
       [0.0, 0.0, 1.0, 46026.58, 85047.44, 205517.64],
       [0.0, 1.0, 0.0, 28676.32, 127056.21, 201126.82],
       [1.0, 0.0, 0.0, 44082.51, 51283.14, 197029.42],
       [0.0, 0.0, 1.0, 20242.15, 65947.93, 185265.1],
       [1.0, 0.0, 0.0, 38571.07, 82982.09, 174999.3],
       [1.0, 0.0, 0.0, 28766.89, 118546.05, 172795.67],
       [0.0, 1.0, 0.0, 27905.48, 84710.77, 164470.71],
       [1.0, 0.0, 0.0, 23653.49, 96189.63, 148001.11],
       [0.0, 0.0, 1.0, 15518.29, 127382.3, 35534.17],
       [1.0, 0.0, 0.0, 22190.3, 154806.14, 28334.72],
       [0.0, 0.0, 1.0, 1012.79, 124153.04, 1903.93],
       [0.0, 1.0, 0.0, 1328.02, 115816.21, 297114.46],
       [1.0, 0.0, 0.0, 12.56, 135426.92, 0.0],
       [0.0, 0.0, 1.0, 554.61, 51743.15, 0.0],
       [1.0, 0.0, 0.0, 12.56, 116983.8, 45173.06]], dtype=object)
```

Dependent variable vector using label encoder

In [52]:

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
y=np.array(le.fit_transform(y))
```

In [53]:

```
y
```

Out[53]:

```
array([49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33,
       32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16,
```

```
 15, 14, 13, 12, 11, 10,  9,  8,  7,  6,  5,  4,  3,  2,  1,  0],
dtype=int64)
```

Step6: Splitting of data into training data set and testing data set

In [54]:

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2,random_state=1)
```

In [55]:

```
xtest
```

Out[55]:

```
array([[0.0, 0.0, 1.0, 72120.16, 127864.55, 353183.81],
       [0.0, 0.0, 1.0, 46026.58, 85047.44, 205517.64],
       [1.0, 0.0, 0.0, 28766.89, 118546.05, 172795.67],
       [0.0, 0.0, 1.0, 20242.15, 65947.93, 185265.1],
       [0.0, 1.0, 0.0, 153454.07, 101145.55, 407934.54],
       [0.0, 0.0, 1.0, 144384.97, 118671.85, 383199.62],
       [0.0, 0.0, 1.0, 554.61, 51743.15, 0.0],
       [0.0, 0.0, 1.0, 65618.04, 153032.06, 107138.38],
       [0.0, 1.0, 0.0, 1328.02, 115816.21, 297114.46],
       [0.0, 0.0, 1.0, 61148.94, 152701.92, 88218.23]], dtype=object)
```

In [56]:

```
ytest
```

Out[56]:

```
array([22, 14,  9, 11, 47, 46,  1, 20,  3, 18], dtype=int64)
```

Step7: Feature scaling

In [57]:

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
xtrain[:,3:]=sc.fit_transform(xtrain[:,3:])
xtest[:,3:]=sc.fit_transform(xtest[:,3:])
```

In [58]:

```
xtrain
```

Out[58]:

```
array([[1.0, 0.0, 0.0, -0.3213303849622652, 0.18700266744239274,
        -1.3700842498314485],
       [1.0, 0.0, 0.0, -0.8955888632549122, -1.6137113933587277,
        -0.28442172627772067],
       [0.0, 0.0, 1.0, 0.024851552213493347, 1.1432479184934816,
        0.7660716916733844],
       [0.0, 1.0, 0.0, -1.1242433521488782, 0.102749342455734,
        -0.06438621193773587],
       [0.0, 0.0, 1.0, 0.21041946816397133, 1.133148735616741,
        -1.758195776958278],
       [1.0, 0.0, 0.0, -1.2403142611108307, -1.0993455063779112,
        -0.5117896748200481],
       [1.0, 0.0, 0.0, -1.786624654615268, -0.2895193240547929,
        -1.377766348852932],
       [0.0, 1.0, 0.0, -0.04587483589220653, 0.7679187112767512,
        -0.6292797489632387],
       [0.0, 1.0, 0.0, -0.07670896217898561, -0.06367998831873588,
        0.7962376091160044]
```

```
 0.7902376091100044],
[1.0, 0.0, 0.0, 0.3391969091012882, 0.43139962169069607,
 0.36964711927627536],
[0.0, 1.0, 0.0, -1.1420564374940387, -1.546388180897815,
 -0.3730892965608864],
[0.0, 0.0, 1.0, 0.40077619330452474, 0.8045891979215861,
 0.6215319446209713],
[0.0, 0.0, 1.0, -1.7635106720056608, -0.010314226695379883,
 -1.7421616406599485],
[0.0, 0.0, 1.0, -0.006240235717204621, -0.9789392720271195,
 -0.5743310006000313],
[0.0, 1.0, 0.0, -0.2260378655739653, -0.7269780562866471,
 0.8084444652190854],
[0.0, 1.0, 0.0, 1.4972866312465332, -1.2861975253390332,
 1.3255280755068437],
[0.0, 1.0, 0.0, -0.5042334093632597, -0.8318776129957924,
 0.049372672462672844],
[0.0, 1.0, 0.0, 0.9851038113375158, 1.2512805897423231,
 0.4020533506685055],
[0.0, 1.0, 0.0, -0.35401482226506187, -0.3418035867251681,
 -0.9907249543987474],
[0.0, 1.0, 0.0, 0.5684508368332148, -0.5383649504456615,
 0.17170600858774815],
[0.0, 1.0, 0.0, -0.26026203340001963, 2.2676691775981626,
 -0.7631989324302334],
[1.0, 0.0, 0.0, -1.2741266324799672, 1.1834666719588485,
 -1.5195721077818902],
[1.0, 0.0, 0.0, -0.7137800342655379, 1.2959308796088318,
 0.01705781542708258],
[0.0, 1.0, 0.0, 0.33357618787703514, -0.398876899674553,
 0.725498782413492],
[1.0, 0.0, 0.0, -0.024499549434031147, -0.41089100165529335,
 0.7570368310936499],
[1.0, 0.0, 0.0, -0.2923093647366418, 0.5894413964873636,
 -0.5963298190950038],
[1.0, 0.0, 0.0, 1.3241592665071586, 0.8872025545643478,
 -0.682615845201753],
[0.0, 1.0, 0.0, 1.2243915225483335, 0.8222109712829453,
 0.9693639710211176],
[1.0, 0.0, 0.0, -1.786624654615268, 0.4287455290135026,
 -1.758195776958278],
[1.0, 0.0, 0.0, 1.9707915498198545, 1.0499422486515493,
 1.9801399458088698],
[1.0, 0.0, 0.0, 0.01615437406951465, -0.10983744453311507,
 0.46802092006648527],
[0.0, 0.0, 1.0, 2.0343750487654315, 0.4860287547953159,
 2.2149810534843644],
[0.0, 0.0, 1.0, 0.8598633830655122, -0.07014133118114546,
 0.4463788405343934],
[0.0, 0.0, 1.0, 1.2608747937205333, -0.9581672483846423,
 1.297677340035081],
[1.0, 0.0, 0.0, 0.539770207281576, -1.270664754636584,
 0.34505284797275304],
[1.0, 0.0, 0.0, 1.0634800927708346, -0.6129420338097887,
 0.8102373395771508],
[0.0, 0.0, 1.0, 0.9989523736664655, 0.946401867796169,
 0.8660866095952406],
[0.0, 1.0, 0.0, 0.38244154533127, 0.11303740793679416,
 0.3458519735673645],
[0.0, 0.0, 1.0, -1.4283078939009661, 0.11544887409338739,
 -1.4589412218916487],
[1.0, 0.0, 0.0, -0.7682268282090787, -2.8482227810289933,
 -0.09889287450872049]], dtype=object)
```

In [59]:

```
ytrain
```

Out[59]:

```
array([17, 10, 28, 13, 30,  7,  0, 23, 27, 36,  8, 32,  4, 25, 26, 45, 16,
       35, 19, 39, 21,  5, 15, 31, 29, 24, 43, 42,  2, 48, 33, 49, 34, 44,
       38, 40, 41, 37,  6, 12], dtype=int64)
```

Build a multiple linear model

In [60]:

```
from sklearn.linear_model import LinearRegression
regn=LinearRegression()
regn.fit(xtrain,ytrain)
```

Out[60]:

```
LinearRegression()
```

In [61]:

```
yestimated=regn.predict(xtest)
```

In [62]:

```
print(yestimated)
```

```
[30.01158205 22.26888696 17.69685572 15.49475121 51.05597432 48.55924607
  9.27563715 26.94608678 11.696468   25.68977058]
```

In [63]:

```
print(ytest)
```

```
[22 14  9 11 47 46  1 20  3 18]
```

In [64]:

```
np.concatenate((yestimated.reshape(len(yestimated),1),yestimated.reshape(len(yestimated)
,1)),1)
```

Out[64]:

```
array([[30.01158205, 30.01158205],
       [22.26888696, 22.26888696],
       [17.69685572, 17.69685572],
       [15.49475121, 15.49475121],
       [51.05597432, 51.05597432],
       [48.55924607, 48.55924607],
       [ 9.27563715,  9.27563715],
       [26.94608678, 26.94608678],
       [11.696468  , 11.696468  ],
       [25.68977058, 25.68977058]])
```

Coefficient of the regressor

In [65]:

```
regn.coef_
```

Out[65]:

```
array([-1.17234733e-01,  1.14000278e-01,  3.23445538e-03,  1.29240964e+01,
        1.38903961e-01,  7.99777571e-01])
```

In [40]:

```
regn.intercept_
```

Out[40]:

```
25.85618518220345
```

In [41]:

```
print(regn.predict([[1,0,0,50661,115641,92496]]))
```

```
[744812.60822068]
```

In [ ]:

```
```