Superwise and unsuperwise learning lab(4)

NAME:- AYUB ALAM

Assignment date:-

Assignment date:-

multiple linear Regression :-

In [ ]:
```
here upload the linear_regression data set
import pandas as pd
import numpy as np
```

In [2]:
```
data=pd.read_csv('multiple_regression.csv')
data
```

Out[2]:

	R&D Spend	Administration	Marketing Spend	Profit
0	165349.20	136897.80	471784.10	192261.83
1	162597.70	151377.59	443898.53	191792.06
2	153441.51	101145.55	407934.54	191050.39
3	144372.41	118671.85	383199.62	182901.99
4	142107.34	91391.77	366168.42	166187.94
5	131876.90	99814.71	362861.36	156991.12
6	134615.46	147198.87	127716.82	156122.51
7	130298.13	145530.06	323876.68	155752.60
8	120542.52	148718.95	311613.29	152211.77
9	123334.88	108679.17	304981.62	149759.96
10	101913.08	110594.11	229160.95	146121.95
11	100671.96	91790.61	249744.55	144259.40
12	93863.75	127320.38	249839.44	141585.52
13	91992.39	135495.07	252664.93	134307.35
14	119943.24	156547.42	256512.92	132602.65
15	114523.61	122616.84	261776.23	129917.04
16	78013.11	121597.55	264346.06	126992.93
17	94657.16	145077.58	282574.31	125370.37
18	91749.16	114175.79	294919.57	124266.90
19	86419.70	153514.11	0.00	122776.86
20	76253.86	113867.30	298664.47	118474.03
21	78389.47	153773.43	299737.29	111313.02
22	73994.56	122782.75	303319.26	110352.25
23	67532.53	105751.03	304768.73	108733.99
24	77044.01	99281.34	140574.81	108552.04
25	64664.71	139553.16	137962.62	107404.34
26	75328.87	144135.98	134050.07	105733.54
27	72107.60	127864.55	353183.81	105008.31
28	66051.52	182645.56	118148.20	103282.38
29	65605.48	153032.06	107138.38	101004.64
30	61994.48	115641.28	91131.24	99937.59
31	61136.38	152701.92	88218.23	97483.56
32	63408.86	129219.61	46085.25	97427.84
33	55493.95	103057.49	214634.81	96778.92
34	46426.07	157693.92	210797.67	96712.80
35	46014.02	85047.44	205517.64	96479.51
36	28663.76	127056.21	201126.82	90708.19
37	44069.95	51283.14	197029.42	89949.14
38	20229.59	65947.93	185265.10	81229.06
39	38558.51	82982.09	174999.30	81005.76
40	28754.33	118546.05	172795.67	78239.91
41	27892.92	84710.77	164470.71	77798.83
42	23640.93	96189.63	148001.11	71498.49

	R&D Spend	Administration	Marketing Spend	Profit
43	15505.73	127382.30	35534.17	69758.98
44	22177.74	154806.14	28334.72	65200.33
45	1000.23	124153.04	1903.93	64926.08
46	1315.46	115816.21	297114.46	49490.75
47	0.00	135426.92	0.00	42559.73
48	542.05	51743.15	0.00	35673.41
49	0.00	116983.80	45173.06	14681.40

In [3]:
```
# here we define dependent and dependent variable
x=data.iloc[:,:-1].values
x
```

Out[3]:
```
array([[165349.2 , 136897.8 , 471784.1 ],
       [162597.7 , 151377.59, 443898.53],
       [153441.51, 101145.55, 407934.54],
       [144372.41, 118671.85, 383199.62],
       [142107.34,  91391.77, 366168.42],
       [131876.9 ,  99814.71, 362861.36],
       [134615.46, 147198.87, 127716.82],
       [130298.13, 145530.06, 323876.68],
       [120542.52, 148718.95, 311613.29],
       [123334.88, 108679.17, 304981.62],
       [101913.08, 110594.11, 229160.95],
       [100671.96,  91790.61, 249744.55],
       [ 93863.75, 127320.38, 249839.44],
       [ 91992.39, 135495.07, 252664.93],
       [119943.24, 156547.42, 256512.92],
       [114523.61, 122616.84, 261776.23],
       [ 78013.11, 121597.55, 264346.06],
       [ 94657.16, 145077.58, 282574.31],
       [ 91749.16, 114175.79, 294919.57],
       [ 86419.7 , 153514.11,      0.  ],
       [ 76253.86, 113867.3 , 298664.47],
       [ 78389.47, 153773.43, 299737.29],
       [ 73994.56, 122782.75, 303319.26],
       [ 67532.53, 105751.03, 304768.73],
       [ 77044.01,  99281.34, 140574.81],
       [ 64664.71, 139553.16, 137962.62],
       [ 75328.87, 144135.98, 134050.07],
       [ 72107.6 , 127864.55, 353183.81],
       [ 66051.52, 182645.56, 118148.2 ],
       [ 65605.48, 153032.06, 107138.38],
       [ 61994.48, 115641.28,  91131.24],
       [ 61136.38, 152701.92,  88218.23],
       [ 63408.86, 129219.61,  46085.25],
       [ 55493.95, 103057.49, 214634.81],
       [ 46426.07, 157693.92, 210797.67],
       [ 46014.02,  85047.44, 205517.64],
       [ 28663.76, 127056.21, 201126.82],
       [ 44069.95,  51283.14, 197029.42],
       [ 20229.59,  65947.93, 185265.1 ],
       [ 38558.51,  82982.09, 174999.3 ],
       [ 28754.33, 118546.05, 172795.67],
       [ 27892.92,  84710.77, 164470.71],
       [ 23640.93,  96189.63, 148001.11],
       [ 15505.73, 127382.3 ,  35534.17],
       [ 22177.74, 154806.14,  28334.72],
       [  1000.23, 124153.04,   1903.93],
       [  1315.46, 115816.21, 297114.46],
       [     0.  , 135426.92,      0.  ],
       [   542.05,  51743.15,      0.  ],
       [     0.  , 116983.8 ,  45173.06]])
```

In [4]:
```
# here we define dependent and dependent variable
y=data.iloc[:,-1].values
y
```

Out[4]:
```
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 [5]:
```
# use the sklearn model for spliting the testing or training data
from sklearn.model_selection import train_test_split
```

In [6]:
```
#here we spliting the hole data in 1/3(there is X_trian,y_train (75) for training and 25 testing)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=1/3,random_state=0)
x_train
```

Out[6]:
```
array([[  1000.23, 124153.04,   1903.93],
       [   542.05,  51743.15,      0.  ],
       [ 65605.48, 153032.06, 107138.38],
       [114523.61, 122616.84, 261776.23],
       [ 61994.48, 115641.28,  91131.24],
       [ 63408.86, 129219.61,  46085.25],
       [ 78013.11, 121597.55, 264346.06],
       [ 23640.93,  96189.63, 148001.11],
```

```
       [ 76253.86, 113867.3 , 298664.47],
       [ 15505.73, 127382.3 ,  35534.17],
       [120542.52, 148718.95, 311613.29],
       [ 91992.39, 135495.07, 252664.93],
       [ 64664.71, 139553.16, 137962.62],
       [131876.9 ,  99814.71, 362861.36],
       [ 94657.16, 145077.58, 282574.31],
       [ 28754.33, 118546.05, 172795.67],
       [     0.  , 116983.8 ,  45173.06],
       [162597.7 , 151377.59, 443898.53],
       [ 93863.75, 127320.38, 249839.44],
       [ 44069.95,  51283.14, 197029.42],
       [ 77044.01,  99281.34, 140574.81],
       [134615.46, 147198.87, 127716.82],
       [ 67532.53, 105751.03, 304768.73],
       [ 28663.76, 127056.21, 201126.82],
       [ 78389.47, 153773.43, 299737.29],
       [ 86419.7 , 153514.11,      0.  ],
       [123334.88, 108679.17, 304981.62],
       [ 38558.51,  82982.09, 174999.3 ],
       [  1315.46, 115816.21, 297114.46],
       [144372.41, 118671.85, 383199.62],
       [165349.2 , 136897.8 , 471784.1 ],
       [     0.  , 135426.92,      0.  ],
       [ 22177.74, 154806.14,  28334.72]])
```

In [7]:
```
x_train
```

Out[7]:
```
array([[  1000.23, 124153.04,   1903.93],
       [   542.05,  51743.15,      0.  ],
       [ 65605.48, 153032.06, 107138.38],
       [114523.61, 122616.84, 261776.23],
       [ 61994.48, 115641.28,  91131.24],
       [ 63408.86, 129219.61,  46085.25],
       [ 78013.11, 121597.55, 264346.06],
       [ 23640.93,  96189.63, 148001.11],
       [ 76253.86, 113867.3 , 298664.47],
       [ 15505.73, 127382.3 ,  35534.17],
       [120542.52, 148718.95, 311613.29],
       [ 91992.39, 135495.07, 252664.93],
       [ 64664.71, 139553.16, 137962.62],
       [131876.9 ,  99814.71, 362861.36],
       [ 94657.16, 145077.58, 282574.31],
       [ 28754.33, 118546.05, 172795.67],
       [     0.  , 116983.8 ,  45173.06],
       [162597.7 , 151377.59, 443898.53],
       [ 93863.75, 127320.38, 249839.44],
       [ 44069.95,  51283.14, 197029.42],
       [ 77044.01,  99281.34, 140574.81],
       [134615.46, 147198.87, 127716.82],
       [ 67532.53, 105751.03, 304768.73],
       [ 28663.76, 127056.21, 201126.82],
       [ 78389.47, 153773.43, 299737.29],
       [ 86419.7 , 153514.11,      0.  ],
       [123334.88, 108679.17, 304981.62],
       [ 38558.51,  82982.09, 174999.3 ],
       [  1315.46, 115816.21, 297114.46],
       [144372.41, 118671.85, 383199.62],
       [165349.2 , 136897.8 , 471784.1 ],
       [     0.  , 135426.92,      0.  ],
       [ 22177.74, 154806.14,  28334.72]])
```

In [8]:
```
y_train
```

Out[8]:
```
array([ 64926.08,  35673.41, 101004.64, 129917.04,  99937.59,  97427.84,
       126992.93,  71498.49, 118474.03,  69758.98, 152211.77, 134307.35,
       107404.34, 156991.12, 125370.37,  78239.91,  14681.4 , 191792.06,
       141585.52,  89949.14, 108552.04, 156122.51, 108733.99,  90708.19,
       111313.02, 122776.86, 149759.96,  81005.76,  49490.75, 182901.99,
       192261.83,  42559.73,  65200.33])
```

In [9]:
```
y_test
```

Out[9]:
```
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 , 155752.6 , 132602.65])
```

In [10]:
```
#use sklearn. linear_model import LinearRegression
from sklearn.linear_model import LinearRegression
linear=LinearRegression()
linear
```

Out[10]:
```
LinearRegression()
```

In [11]:
```
linear=LinearRegression()
```

In [12]:
```
linear
```

Out[12]:
```
LinearRegression()
```

In [13]:
```
# here fit the model using fit method
model=linear.fit(x_train,y_train)
```

In [14]:
```
model
```

Out[14]:
```
LinearRegression()
```

In [15]:
```
#here we pridect the model using x_test
y_pred=model.predict(x_test)
```

In [16]:
```
y_pred
```

Out[16]:
```
array([106297.59966882, 132926.07997436, 134362.77842228,  71519.46921658,
       180650.71715655, 115248.2571595 ,  64939.68480136,  99726.11825428,
       114940.23331478, 169718.80610512,  96249.07005881,  87336.2065521 ,
       112030.13596157,  91975.69916321, 128433.62518338, 162036.7231283 ,
       152289.76520475])
```

In [17]:
```
# find the coefficient
m=model.coef_
m
```

Out[17]:
```
array([0.80219763, 0.05708141, 0.03071646])
```

In [20]:
```
#find the intercept value
c=model.intercept_
c
```

Out[20]:
```
39256.46733772641
```

In [18]:
```
#find the mean_absolute_error and mean_squared_error using the sklearn.metrics
from sklearn.metrics import mean_absolute_error,mean_squared_error
```

In [19]:
```
print(mean_absolute_error(y_test,y_pred))
```

```
7677.781684426058
```

In [23]:
```
x_test
```

Out[23]:
```
array([[ 66051.52, 182645.56, 118148.2 ],
       [100671.96,  91790.61, 249744.55],
       [101913.08, 110594.11, 229160.95],
       [ 27892.92,  84710.77, 164470.71],
       [153441.51, 101145.55, 407934.54],
       [ 72107.6 , 127864.55, 353183.81],
       [ 20229.59,  65947.93, 185265.1 ],
       [ 61136.38, 152701.92,  88218.23],
       [ 73994.56, 122782.75, 303319.26],
       [142107.34,  91391.77, 366168.42],
       [ 55493.95, 103057.49, 214634.81],
       [ 46014.02,  85047.44, 205517.64],
       [ 75328.87, 144135.98, 134050.07],
       [ 46426.07, 157693.92, 210797.67],
       [ 91749.16, 114175.79, 294919.57],
       [130298.13, 145530.06, 323876.68],
       [119943.24, 156547.42, 256512.92]])
```

In [24]:
```
#here we define x1,x2,x3 for each error and put those
x1,x2,x3=66051.52, 182645.56, 118148.2
```

In [25]:
```
c1,c2,c3=m
```

In [26]:
```
y=c1*x1+c2*x2+c3*x3+c
```

In [27]:
```
y
```

Out[27]:
```
106297.599668825
```

In [28]:
```
y_pred
```

Out[28]:
```
array([106297.59966882, 132926.07997436, 134362.77842228,  71519.46921658,
       180650.71715655, 115248.2571595 ,  64939.68480136,  99726.11825428,
       114940.23331478, 169718.80610512,  96249.07005881,  87336.2065521 ,
       112030.13596157,  91975.69916321, 128433.62518338, 162036.7231283 ,
       152289.76520475])
```