```
import pandas as pd

dataset=pd.read_csv("Billing_Profit_withYear.csv")

dataset
```

	Year	R&D Spend	Admin Spending	Billing	Technology	Profit
0	1990	165349.20	136897.80	471784.10	Quality	192261.83
1	1990	162597.70	151377.59	443898.53	Devlopment	191792.06
2	1991	153441.51	101145.55	407934.54	Quality	191050.39
3	1991	144372.41	118671.85	383199.62	Devlopment	182901.99
4	1992	142107.34	91391.77	366168.42	Quality	166187.94
5	1992	131876.90	99814.71	362861.36	Devlopment	156991.12
6	1993	134615.46	147198.87	127716.82	Quality	156122.51
7	1993	130298.13	145530.06	323876.68	Devlopment	155752.60
8	1994	120542.52	148718.95	311613.29	Quality	152211.77
9	1994	123334.88	108679.17	304981.62	Devlopment	149759.96
10	1995	101913.08	110594.11	229160.95	Quality	146121.95
11	1995	100671.96	91790.61	249744.55	Devlopment	144259.40
12	1996	93863.75	127320.38	249839.44	Quality	141585.52
13	1996	91992.39	135495.07	252664.93	Devlopment	134307.35
14	1997	119943.24	156547.42	256512.92	Quality	132602.65
15	1997	114523.61	122616.84	261776.23	Devlopment	129917.04
16	1998	78013.11	121597.55	264346.06	Quality	126992.93
17	1998	94657.16	145077.58	282574.31	Devlopment	125370.37
18	1999	91749.16	114175.79	294919.57	Quality	124266.90
19	1999	86419.70	153514.11	296019.57	Devlopment	122776.86
20	2000	76253.86	113867.30	298664.47	Quality	118474.03
21	2000	78389.47	153773.43	299737.29	Devlopment	111313.02
22	2001	73994.56	122782.75	303319.26	Quality	110352.25
23	2001	67532.53	105751.03	304768.73	Devlopment	108733.99
24	2002	77044.01	99281.34	140574.81	Quality	108552.04
25	2002	64664.71	139553.16	137962.62	Devlopment	107404.34
26	2003	75328.87	144135.98	134050.07	Quality	105733.54
27	2003	72107.60	127864.55	353183.81	Devlopment	105008.31
28	2004	66051.52	182645.56	118148.20	Quality	103282.38
29	2004	65605.48	153032.06	107138.38	Devlopment	101004.64
30	2005	61994.48	115641.28	91131.24	Quality	99937.59
31	2005	61136.38	152701.92	88218.23	Devlopment	97483.56
32	2006	63408.86	129219.61	46085.25	Quality	97427.84
33	2006	55493.95	103057.49	214634.81	Devlopment	96778.92
34	2007	46426.07	157693.92	210797.67	Quality	96712.80
35	2007	46014.02	85047.44	205517.64	Devlopment	96479.51
36	2008	28663.76	127056.21	201126.82	Quality	90708.19
37	2008	44069.95	51283.14	197029.42	Devlopment	89949.14
38	2009	20229.59	65947.93	185265.10	Quality	81229.06
39	2009	38558.51	82982.09	174999.30	Devlopment	81005.76
40	2010	28754.33	118546.05	172795.67	Quality	78239.91
41	2010	27892.92	84710.77	164470.71	Devlopment	77798.83
42	2011	23640.93	96189.63	148001.11	Quality	71498.49

```
43  2011   15505.73       127382.30   35534.17  Devlopment   69758.98
44  2012   22177.74       154806.14   28334.72     Quality   65200.33
45  2012    1000.23       124153.04    1903.93  Devlopment   64926.08
46  2013    1315.46       115816.21  297114.46     Quality   49490.75
47  2013     542.05       135426.92     542.05  Devlopment   42559.73
48  2014     542.05        51743.15     542.05     Quality   35673.41
49  2014     542.05       116983.80   45173.06  Devlopment  146851.40
```

```
dataset=pd.get_dummies(dataset,drop_first=True)
```

```
dataset=dataset.replace({True: 1, False: 0})
```

```
dataset
```

```
    Year  R&D Spend  Admin Spending    Billing     Profit
Technology_Quality
0   1990  165349.20       136897.80  471784.10  192261.83
1
1   1990  162597.70       151377.59  443898.53  191792.06
0
2   1991  153441.51       101145.55  407934.54  191050.39
1
3   1991  144372.41       118671.85  383199.62  182901.99
0
4   1992  142107.34        91391.77  366168.42  166187.94
1
5   1992  131876.90        99814.71  362861.36  156991.12
0
6   1993  134615.46       147198.87  127716.82  156122.51
1
7   1993  130298.13       145530.06  323876.68  155752.60
0
8   1994  120542.52       148718.95  311613.29  152211.77
1
9   1994  123334.88       108679.17  304981.62  149759.96
0
10  1995  101913.08       110594.11  229160.95  146121.95
1
11  1995  100671.96        91790.61  249744.55  144259.40
0
12  1996   93863.75       127320.38  249839.44  141585.52
1
13  1996   91992.39       135495.07  252664.93  134307.35
0
14  1997  119943.24       156547.42  256512.92  132602.65
1
15  1997  114523.61       122616.84  261776.23  129917.04
0
16  1998   78013.11       121597.55  264346.06  126992.93
1
```

```
17  1998   94657.16       145077.58  282574.31  125370.37
0
18  1999   91749.16       114175.79  294919.57  124266.90
1
19  1999   86419.70       153514.11  296019.57  122776.86
0
20  2000   76253.86       113867.30  298664.47  118474.03
1
21  2000   78389.47       153773.43  299737.29  111313.02
0
22  2001   73994.56       122782.75  303319.26  110352.25
1
23  2001   67532.53       105751.03  304768.73  108733.99
0
24  2002   77044.01        99281.34  140574.81  108552.04
1
25  2002   64664.71       139553.16  137962.62  107404.34
0
26  2003   75328.87       144135.98  134050.07  105733.54
1
27  2003   72107.60       127864.55  353183.81  105008.31
0
28  2004   66051.52       182645.56  118148.20  103282.38
1
29  2004   65605.48       153032.06  107138.38  101004.64
0
30  2005   61994.48       115641.28   91131.24   99937.59
1
31  2005   61136.38       152701.92   88218.23   97483.56
0
32  2006   63408.86       129219.61   46085.25   97427.84
1
33  2006   55493.95       103057.49  214634.81   96778.92
0
34  2007   46426.07       157693.92  210797.67   96712.80
1
35  2007   46014.02        85047.44  205517.64   96479.51
0
36  2008   28663.76       127056.21  201126.82   90708.19
1
37  2008   44069.95        51283.14  197029.42   89949.14
0
38  2009   20229.59        65947.93  185265.10   81229.06
1
39  2009   38558.51        82982.09  174999.30   81005.76
0
40  2010   28754.33       118546.05  172795.67   78239.91
1
41  2010   27892.92        84710.77  164470.71   77798.83
```

```
0
42  2011   23640.93        96189.63  148001.11   71498.49
1
43  2011   15505.73       127382.30   35534.17   69758.98
0
44  2012   22177.74       154806.14   28334.72   65200.33
1
45  2012    1000.23       124153.04    1903.93   64926.08
0
46  2013    1315.46       115816.21  297114.46   49490.75
1
47  2013     542.05       135426.92     542.05   42559.73
0
48  2014     542.05        51743.15     542.05   35673.41
1
49  2014     542.05       116983.80   45173.06  146851.40
0
```

```
dataset.columns
```

```
Index(['Year', 'R&D Spend', 'Admin Spending', 'Billing', 'Profit',
       'Technology_Quality'],
      dtype='object')
```

```
independent=dataset[['Year', 'R&D Spend', 'Admin Spending', 'Billing',
'Technology_Quality']]
```

```
dependent=dataset[['Profit']]
```

```
independent
```

	Year	R&D Spend	Admin Spending	Billing	Technology_Quality
0	1990	165349.20	136897.80	471784.10	1
1	1990	162597.70	151377.59	443898.53	0
2	1991	153441.51	101145.55	407934.54	1
3	1991	144372.41	118671.85	383199.62	0
4	1992	142107.34	91391.77	366168.42	1
5	1992	131876.90	99814.71	362861.36	0
6	1993	134615.46	147198.87	127716.82	1
7	1993	130298.13	145530.06	323876.68	0
8	1994	120542.52	148718.95	311613.29	1
9	1994	123334.88	108679.17	304981.62	0
10	1995	101913.08	110594.11	229160.95	1
11	1995	100671.96	91790.61	249744.55	0
12	1996	93863.75	127320.38	249839.44	1
13	1996	91992.39	135495.07	252664.93	0
14	1997	119943.24	156547.42	256512.92	1
15	1997	114523.61	122616.84	261776.23	0
16	1998	78013.11	121597.55	264346.06	1
17	1998	94657.16	145077.58	282574.31	0
18	1999	91749.16	114175.79	294919.57	1

19	1999	86419.70	153514.11	296019.57		0
20	2000	76253.86	113867.30	298664.47		1
21	2000	78389.47	153773.43	299737.29		0
22	2001	73994.56	122782.75	303319.26		1
23	2001	67532.53	105751.03	304768.73		0
24	2002	77044.01	99281.34	140574.81		1
25	2002	64664.71	139553.16	137962.62		0
26	2003	75328.87	144135.98	134050.07		1
27	2003	72107.60	127864.55	353183.81		0
28	2004	66051.52	182645.56	118148.20		1
29	2004	65605.48	153032.06	107138.38		0
30	2005	61994.48	115641.28	91131.24		1
31	2005	61136.38	152701.92	88218.23		0
32	2006	63408.86	129219.61	46085.25		1
33	2006	55493.95	103057.49	214634.81		0
34	2007	46426.07	157693.92	210797.67		1
35	2007	46014.02	85047.44	205517.64		0
36	2008	28663.76	127056.21	201126.82		1
37	2008	44069.95	51283.14	197029.42		0
38	2009	20229.59	65947.93	185265.10		1
39	2009	38558.51	82982.09	174999.30		0
40	2010	28754.33	118546.05	172795.67		1
41	2010	27892.92	84710.77	164470.71		0
42	2011	23640.93	96189.63	148001.11		1
43	2011	15505.73	127382.30	35534.17		0
44	2012	22177.74	154806.14	28334.72		1
45	2012	1000.23	124153.04	1903.93		0
46	2013	1315.46	115816.21	297114.46		1
47	2013	542.05	135426.92	542.05		0
48	2014	542.05	51743.15	542.05		1
49	2014	542.05	116983.80	45173.06		0

dependent

	Profit
0	192261.83
1	191792.06
2	191050.39
3	182901.99
4	166187.94
5	156991.12
6	156122.51
7	155752.60
8	152211.77
9	149759.96
10	146121.95
11	144259.40
12	141585.52
13	134307.35
14	132602.65

```
15  129917.04
16  126992.93
17  125370.37
18  124266.90
19  122776.86
20  118474.03
21  111313.02
22  110352.25
23  108733.99
24  108552.04
25  107404.34
26  105733.54
27  105008.31
28  103282.38
29  101004.64
30   99937.59
31   97483.56
32   97427.84
33   96778.92
34   96712.80
35   96479.51
36   90708.19
37   89949.14
38   81229.06
39   81005.76
40   78239.91
41   77798.83
42   71498.49
43   69758.98
44   65200.33
45   64926.08
46   49490.75
47   42559.73
48   35673.41
49  146851.40
```

```
#Training code
#Now we need to split the Training and Test data set from parant
dataset. To split, we are using sklearn as class and call function as
model_selection-> train_test_split
from sklearn.model_selection import train_test_split
#input parameters are passed in function train_test_split with
x,y,size for test data(30 percent from x & y) and random state then
assign to variable x_train,x_test,y_train and y_test
x_train,x_test,y_train,y_test=train_test_split(independent, dependent,
test_size=0.30, random_state=600)

#Model Creation by using linear regression algorithm, here importing
LinearRegression algorithm from sklearn.liner_model
from sklearn.linear_model import LinearRegression
```

```
#assign the LinearRegression fucntion without parameters in the
variable regressor
regressor=LinearRegression()
#calling the fit function from LinearRegression by access operator
from variable regressor and pass input train and output train as
parameter
#train model
regressor.fit(x_train,y_train)
```

```
LinearRegression()
```

```
#from above code, weight and bais will be calculated
#slope or weight will be calculated and saved in coef_ of regressor,
we assign this value in weight variable
weight = regressor.coef_
weight
```

```
array([[-1.57596145e+03,  4.95332295e-01,  6.65971832e-04,
         1.22321754e-02, -2.78361261e+03]])
```

```
#bais will be calculated and saved in intercept of regressor, we
assign this value in bais variable
bais = regressor.intercept_
bais
```

```
array([3232322.72833456])
```

```
#now predict using x_test data and assign to y_pred varaible using
above model created which is present in regressor
y_pred=regressor.predict(x_test)
```

```
#import r2_score from sklean.metrics functions
from sklearn.metrics import r2_score
#compare actual test (y_test) vs predicted value on x_test which is
assign in y_pred
r_score=r2_score(y_test,y_pred)
```

```
# print the r_score verify it is near to 1 or not. If near to 1, we
can confirm that created model is good else created model is bad
r_score
```

```
0.9696711040027688
```

```
#Save the model created using pickle function, created the sav file to
save the model
import pickle
filename="finalized_model_Mul_linear_billing_profit.sav"
```

```
#save the regressor where model is present in filename using
pickle.dump function
pickle.dump(regressor,open(filename,'wb'))
```

```
#read the filename where model is saved in the variable by load
function in pickle
loaded_model=pickle.load(open("finalized_model_Mul_linear_billing_prof
it.sav",'rb'))
#assign variable to pass the value during runtime
#Year = int(input("Enter the amount for Year:"))
#R&D Spend = int(input("Enter the amount for R&D Spend:"))
#Admin Spending = int(input("Enter the amount for Admin Spending:"))
#Billing = int(input("Enter the amount for Billing:"))
#Technology_Quality = int(input("Enter the amount for
Technology_Quality:"))
result = loaded_model.predict([[2004,65605.48,153032.06,107138.38,0]])
```

```
C:\Users\Maheshwaran\anaconda3\Lib\site-packages\sklearn\base.py:464:
UserWarning: X does not have valid feature names, but LinearRegression
was fitted with feature names
  warnings.warn(
```

```
result
```

```
array([[108004.94861804]])
```