In [1]:
```
import numpy as np
import pandas as pd
```

USING LINEAR REGRESSION

In [2]:
```
from matplotlib import pyplot as plt
```

In [3]:
```
df=pd.read_csv("C:/Users/user/Downloads/data- linear regression.csv")
df
```

Out[3]:

	qualification	experience	previous exp	Salary
0	0	1	5	23500
1	0	3	2	24500
2	0	5	6	32500
3	0	8	4	36500
4	0	4	2	26500
5	0	6	2	30500
6	0	6	6	34500
7	0	7	4	34500
8	0	8	10	42500
9	0	5	2	28500
10	0	3	5	27500
11	0	2	3	23500
12	0	4	4	28500
13	0	6	2	30500
14	1	1	5	73000
15	1	3	2	73000
16	1	5	6	87000
17	1	8	4	92000
18	1	4	5	82000
19	1	6	3	84000
20	1	6	4	86000
21	1	7	4	89000
22	1	8	10	104000
23	1	5	2	79000
24	1	3	5	79000
25	1	2	3	72000
26	1	4	4	80000
27	1	6	2	82000
28	1	6	2	82000

In [4]:
```
df.describe
```

Out[4]:
```
<bound method NDFrame.describe of     qualification  experience  previous exp  Salary
0                0           1             5   23500
1                0           3             2   24500
2                0           5             6   32500
3                0           8             4   36500
4                0           4             2   26500
5                0           6             2   30500
6                0           6             6   34500
7                0           7             4   34500
8                0           8            10   42500
9                0           5             2   28500
10               0           3             5   27500
11               0           2             3   23500
12               0           4             4   28500
13               0           6             2   30500
14               1           1             5   73000
15               1           3             2   73000
16               1           5             6   87000
17               1           8             4   92000
18               1           4             5   82000
19               1           6             3   84000
20               1           6             4   86000
21               1           7             4   89000
22               1           8            10  104000
23               1           5             2   79000
24               1           3             5   79000
25               1           2             3   72000
26               1           4             4   80000
27               1           6             2   82000
28               1           6             2   82000>
```

In [5]:
```
from sklearn.linear_model import LinearRegression
```

In [6]:
```
lr=LinearRegression()
```

In [7]:
```
x1=df.drop(['Salary'],axis=1)
y1=df['Salary']
```

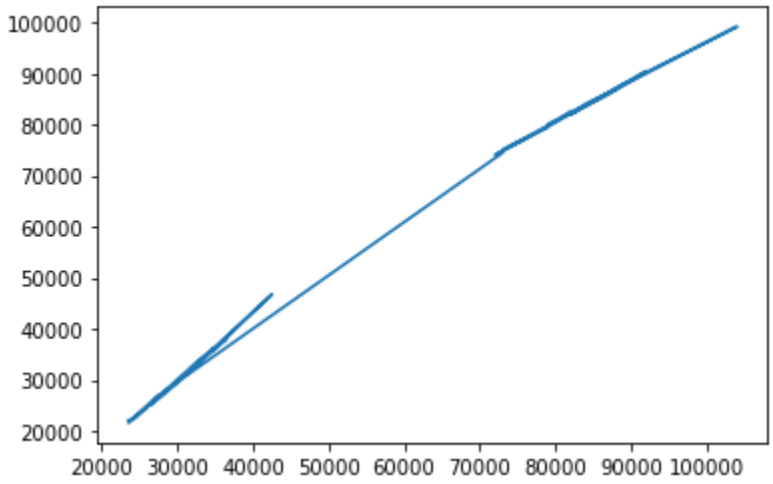
In [8]:
```
lr.fit(x1,y1)
```

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

In [9]:
```
y_predict=lr.predict(x1)
y_predict
```

Out[9]:
```
array([22024.89505713, 22644.27179377, 33447.44384183, 38013.57870087,
       25136.20201453, 30120.06245607, 35939.3740626 , 35521.6484801 ,
       46742.54611066, 27628.1322353 , 27008.75549866, 21607.16947463,
       28045.8578178 , 30120.06245607, 74489.58050651, 75108.95724314,
       85912.12929121, 90478.26415024, 81965.37116881, 84039.57580708,
       85494.40370871, 87986.33392948, 99207.23156004, 80092.81768468,
       79473.44094804, 74071.85492401, 80510.54326718, 82584.74790544,
       82584.74790544])
```

In [10]:
```
plt.plot(y1,y_predict)
plt.show()
```



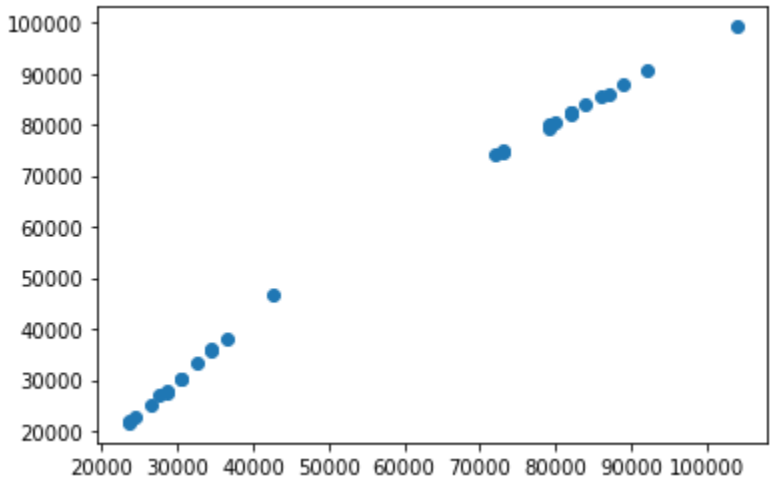
In [11]:
```
from sklearn.metrics import r2_score
```

In [12]:
```
r2_score(y1,y_predict)
```

Out[12]:
```
0.9963584151412598
```

In [13]:
```
plt.scatter(y1,y_predict)
plt.plot()
```

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



USING RANDOM FOREST

In [14]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [15]:
```
rfc1=RandomForestClassifier(n_estimators=200,max_depth=2)
```

In [16]:
```
rfc1
```

Out[16]:
```
RandomForestClassifier(max_depth=2, n_estimators=200)
```

In [17]:
```
from sklearn.model_selection import train_test_split
```

In [18]:
```
x_train,x_test,y_train,y_test=train_test_split(x1,y1,test_size=0.3,random_state=0)
```

In [19]:
```
x_train
```

Out[19]:

	qualification	experience	previous exp
14	1	1	5
8	0	8	10
16	1	5	6
22	1	8	10
17	1	8	4
1	0	3	2
10	0	3	5
28	1	6	2
6	0	6	6
4	0	4	2
18	1	4	5
19	1	6	3
9	0	5	2
7	0	7	4
24	1	3	5
3	0	8	4
0	0	1	5
21	1	7	4
15	1	3	2
12	0	4	4

In [26]:
```
rfc1.fit(x,y1)
```

Out[26]:
```
RandomForestClassifier(max_depth=2, n_estimators=200)
```

In [27]:
```
y_predict=rfc1.predict(x_test)
y_predict
```

Out[27]:
```
array([34500, 82000, 82000, 30500, 23500, 73000, 82000, 82000, 30500],
      dtype=int64)
```

In [28]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
```

In [29]:
```
confusion_matrix(y_test,y_predict)
```

Out[29]:
```
array([[1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 2, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 1, 0]], dtype=int64)
```

In [30]:
```
accuracy_score(y_test,y_predict)
```

Out[30]:
```
0.4444444444444444
```

USING SVM-SUPPORT VECTOR MACHINE

In [34]:
```
from sklearn.svm import SVC
sv1=SVC(kernel='linear')
sv1.fit(x_train,y_train)
y_predict=sv1.predict(x_test)
y_predict
```

Out[34]:
```
array([87000, 89000, 82000, 28500, 73000, 73000, 82000, 82000, 28500],
      dtype=int64)
```

In [35]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
confusion_matrix(y_test,y_predict)
```

Out[35]:
```
array([[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
       [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]], dtype=int64)
```

In [36]:
```
accuracy_score(y_test,y_predict)
```

Out[36]:
```
0.1111111111111111
```

In [ ]:
```

```