In [1]:

```
!pip install panda
```

```
Requirement already satisfied: panda in ./anaconda3/lib/python3.11/s
ite-packages (0.3.1)
Requirement already satisfied: setuptools in ./anaconda3/lib/python
3.11/site-packages (from panda) (68.0.0)
Requirement already satisfied: requests in ./anaconda3/lib/python3.1
1/site-packages (from panda) (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in ./anacond
a3/lib/python3.11/site-packages (from requests->panda) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in ./anaconda3/lib/pytho
n3.11/site-packages (from requests->panda) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in ./anaconda3/li
b/python3.11/site-packages (from requests->panda) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in ./anaconda3/li
b/python3.11/site-packages (from requests->panda) (2023.7.22)
```

In [2]:

```
!pip install numpy
```

```
Requirement already satisfied: numpy in ./anaconda3/lib/python3.11/s
ite-packages (1.24.3)
```

In [3]:

```
import matplotlib.pyplot as plt
```

In [4]:

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
```

In [5]:

```
data=pd.read_csv("Salary_Data.csv")
```

In [6]:

```
data
```

Out[6]:

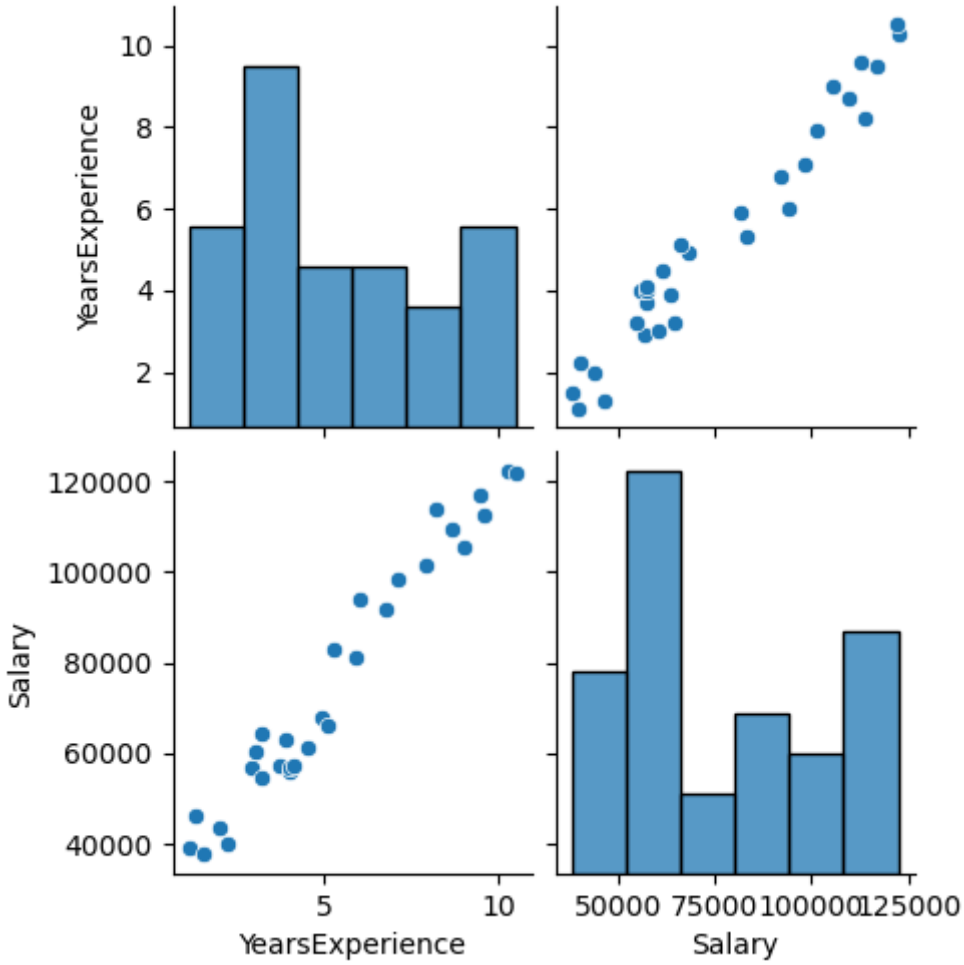
	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0
11	4.0	55794.0
12	4.0	56957.0
13	4.1	57081.0
14	4.5	61111.0
15	4.9	67938.0
16	5.1	66029.0
17	5.3	83088.0
18	5.9	81363.0
19	6.0	93940.0
20	6.8	91738.0
21	7.1	98273.0
22	7.9	101302.0
23	8.2	113812.0
24	8.7	109431.0
25	9.0	105582.0
26	9.5	116969.0
27	9.6	112635.0
28	10.3	122391.0
29	10.5	121872.0

In [60]:

```
import seaborn as sns
sns.pairplot(data)
```

Out[60]:

```
<seaborn.axisgrid.PairGrid at 0x7f500e381750>
```

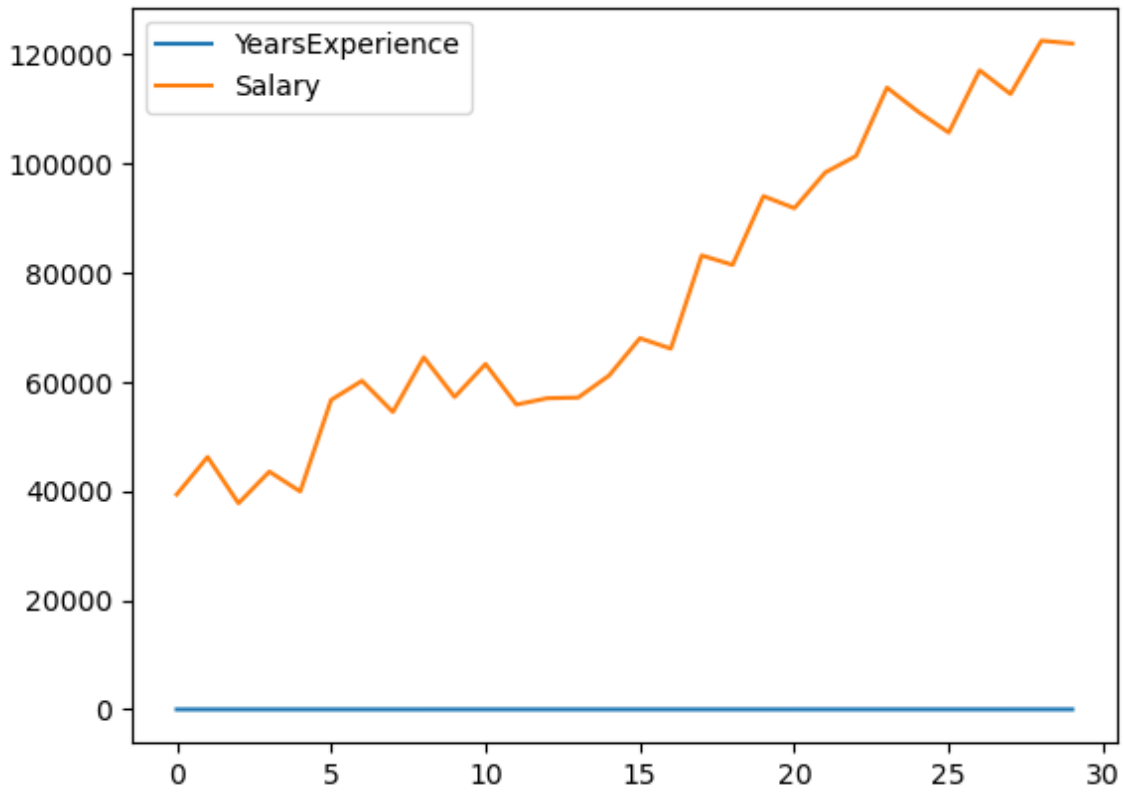


In [7]:

```
data.plot()
```

Out[7]:

```
<Axes: >
```



In [8]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   YearsExperience  30 non-null     float64
 1   Salary           30 non-null     float64
dtypes: float64(2)
memory usage: 612.0 bytes
```

In [9]:

```
x=data.iloc[:,0:1]
```

In [10]:

```
y=data.iloc[:,-1:]
```

In [11]:

```
from sklearn.model_selection import train_test_split
```

In [12]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=1)
```

In [13]:

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(21, 1)
(9, 1)
(21, 1)
(9, 1)
```

In [14]:

```
lr=LinearRegression()
```

In [15]:

```
model=lr.fit(x_train,y_train)
```

In [16]:

```
prediction=lr.predict(x_test)
```

In [17]:

```
prediction
```

Out[17]:

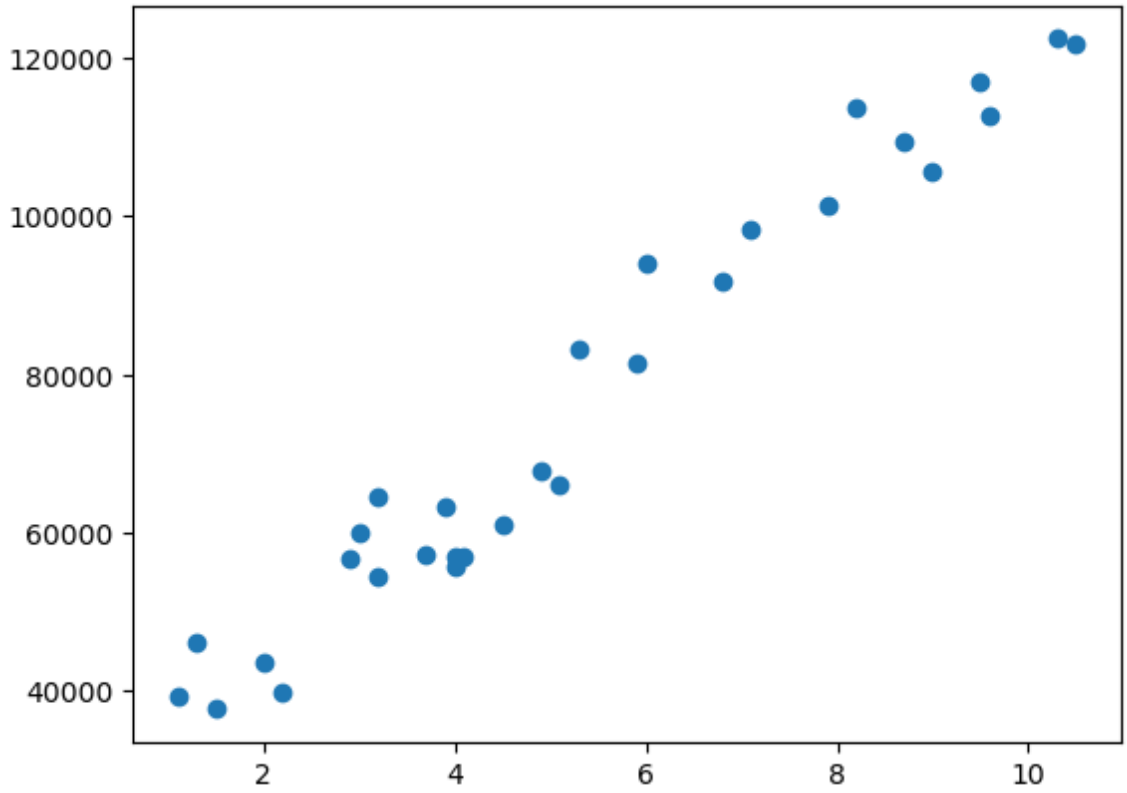
```
array([[ 74821.41578619],
       [ 91385.43626305],
       [ 61938.28874864],
       [ 81262.97930497],
       [ 67459.62890759],
       [ 88624.76618357],
       [113470.79689886],
       [ 44454.04491195],
       [106109.01002026]])
```

In [18]:

```
plt.scatter(x,y)
```

Out[18]:

```
<matplotlib.collections.PathCollection at 0x7f5010148150>
```



In [19]:

```
lr.coef_
```

Out[19]:

```
array([[9202.23359825]])
```

In [20]:

```
lr.intercept_
```

Out[20]:

```
array([26049.57771544])
```

In [21]:

```
#y=mx+c  where m is slope and c= coef - constant
```

In [22]:

```
from sklearn import metrics
```

In [23]:

```
from sklearn.metrics import accuracy_score
```

In [24]:

```
metrics.r2_score(y_test,prediction)
```

Out[24]:

```
0.9248580247217075
```

In [25]:

```
predicted_values=pd.DataFrame(prediction,columns=["Predicted"])
```

In [26]:

```
predicted_values
```

Out[26]:

	Predicted
0	74821.415786
1	91385.436263
2	61938.288749
3	81262.979305
4	67459.628908
5	88624.766184
6	113470.796899
7	44454.044912
8	106109.010020

In [29]:

```
y_test
```

Out[29]:

	Salary
17	83088.0
21	98273.0
10	63218.0
19	93940.0
14	61111.0
20	91738.0
26	116969.0
3	43525.0
24	109431.0

In [30]:

```
y=9202.2*1.1+26050
```

In [31]:

```
y
```

Out[31]:

```
36172.42
```

In [44]:

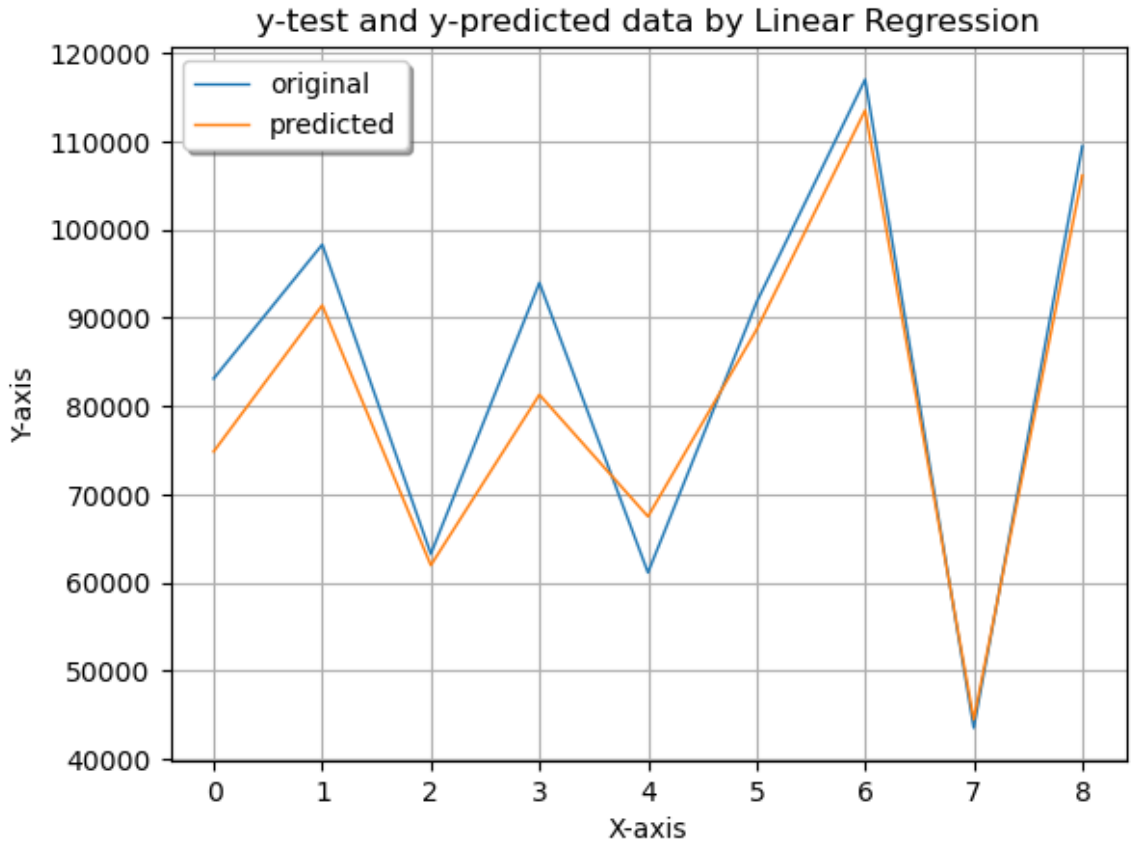
```
y_pred
```

Out[44]:

	Predicted
0	66029.0
1	101302.0
2	56375.5
3	81363.0
4	57081.0
5	81363.0
6	112635.0
7	39891.0
8	105582.0

In [54]:

```
plt.plot(x_ax,y_test, linewidth=1, label="original")
plt.plot(x_ax,prediction,linewidth=1, label="predicted")
plt.title("y-test and y-predicted data by Linear Regression")
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.legend(loc='best',fancybox=True, shadow=True)
plt.grid(True)
plt.show()
```



In [47]:

```
x_ax = range(len(y_test))
```

In [35]:

```
from sklearn.tree import DecisionTreeRegressor
```

In [39]:

```
dt=DecisionTreeRegressor()
```

In [40]:

```
dt.fit(x_train,y_train)
```

Out[40]:

```
▾ DecisionTreeRegressor
DecisionTreeRegressor()
```

In [41]:

```
y_pred=dt.predict(x_test)
```

In [43]:

```
y_pred=pd.DataFrame(y_pred,columns=["Predicted"])
```

In [51]:

```
y_pred
```

Out[51]:

	Predicted
0	66029.0
1	101302.0
2	56375.5
3	81363.0
4	57081.0
5	81363.0
6	112635.0
7	39891.0
8	105582.0

In [53]:

```
plt.plot(x_ax,y_test, linewidth=1, label="original")
plt.plot(x_ax,y_pred,linewidth=1, label="predicted")
plt.title("y-test and y-predicted data by decision tree")
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.legend(loc='best',fancybox=True, shadow=True)
plt.grid(True)
plt.show()
```

In [58]:

```
metrics.r2_score(y_test,y_pred)
```

Out[58]:

```
0.855824415472517
```

In [ ]: