Practicle:2 Linear Regression

2021BIT023
Vaibhav Lanjewar

Use case : We have to predict the salary using experience by using linear regresion

In [ ]:

Import libraries

In [3]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [4]:
```python
df=pd.read_csv("Salary_Data.csv")
```

In [5]:
```python
df.head(2)
```

Out[5]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0

In [6]:
```python
df.tail(2)
```

Out[6]:

	YearsExperience	Salary
28	10.3	122391.0
29	10.5	121872.0

In [7]:
```python
df.isnull().sum()
```

Out[7]:
```
YearsExperience      0
Salary               3
dtype: int64
```

In [8]:
```python
df.notnull().sum()
```

Out[8]:
```
YearsExperience     30
Salary              27
dtype: int64
```

In [9]:
```python
df.isnull().sum()
```

```
Out[9]: YearsExperience     0
        Salary              3
        dtype: int64
```
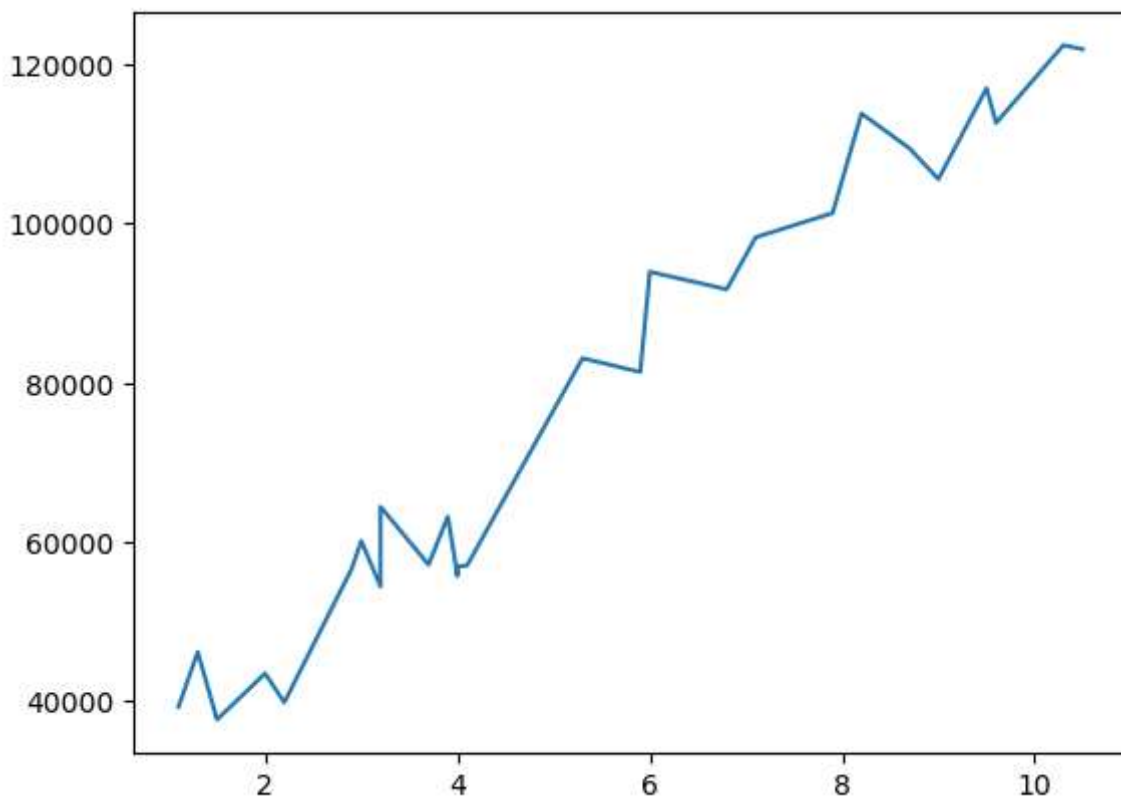
In [10]:
```python
df.dropna(inplace=True)
```

In [11]:
```python
df.corr() # correlation
```

Out[11]:

	YearsExperience	Salary
YearsExperience	1.00000	0.98131
Salary	0.98131	1.00000

In [13]:
```python
X=df.iloc[:,:-1].values # independent variable
y = df.iloc[:, -1].values # dependent variable
```

In [14]:
```python
# graph
plt.plot(X,y)
plt.show()
```


In [ ]:

In [19]:
```python
X
```

```
Out[19]:  array([[ 1.1],
                 [ 1.3],
                 [ 1.5],
                 [ 2. ],
                 [ 2.2],
                 [ 2.9],
                 [ 3. ],
                 [ 3.2],
                 [ 3.2],
                 [ 3.7],
                 [ 3.9],
                 [ 4. ],
                 [ 4. ],
                 [ 4.1],
                 [ 5.3],
                 [ 5.9],
                 [ 6. ],
                 [ 6.8],
                 [ 7.1],
                 [ 7.9],
                 [ 8.2],
                 [ 8.7],
                 [ 9. ],
                 [ 9.5],
                 [ 9.6],
                 [10.3],
                 [10.5]])
```

In [16]:
```
y
```

```
Out[16]:  array([ 39343.,  46205.,  37731.,  43525.,  39891.,  56642.,  60150.,
                  54445.,  64445.,  57189.,  63218.,  55794.,  56957.,  57081.,
                  83088.,  81363.,  93940.,  91738.,  98273., 101302., 113812.,
                 109431., 105582., 116969., 112635., 122391., 121872.])
```

In [20]:
```
pip install -U scikit-learn
```

```
Requirement already satisfied: scikit-learn in c:\users\vlanj\anaconda3\lib\site-pack
ages (1.4.0)
Requirement already satisfied: numpy>=1.19.5 in c:\users\vlanj\anaconda3\lib\site-pac
kages (from scikit-learn) (1.24.3)
Requirement already satisfied: scipy>=1.6.0 in c:\users\vlanj\anaconda3\lib\site-pack
ages (from scikit-learn) (1.11.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\vlanj\anaconda3\lib\site-pac
kages (from scikit-learn) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\vlanj\anaconda3\lib\s
ite-packages (from scikit-learn) (2.2.0)
Note: you may need to restart the kernel to use updated packages.
```

In [26]:
```python
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=
```

In [25]:
```python
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(X_train, y_train)
```

Out[25]:

▼   **LinearRegression** ⓘ ⓘ

LinearRegression()

In [27]:
```python
pred = lm.predict(X_test)
```

In [32]:
```python
print("y_test",y_test)
print("X_test",X_test)
```

```
y_test [ 37731. 112635.  83088.  91738.  56642.  55794.]
X_test [[1.5]
 [9.6]
 [5.3]
 [6.8]
 [2.9]
 [4. ]]
```

In [33]:
```python
pred
```

Out[33]:
```
array([ 41144.69206511, 117316.34008101,  76879.53928245,  90985.40002613,
        54310.16209255,  64654.45997125])
```

In [34]:
```python
print(lm.intercept())
```

```
---------------------------------------------------------------------------
AttributeError                              Traceback (most recent call last)
Cell In[34], line 1
----> 1 print(lm.intercept())

AttributeError: 'LinearRegression' object has no attribute 'intercept'
```

In [35]:
```python
#y=mx+c
#y=ax+c
print(lm.intercept_) # a
print(lm.coef_)    #c
```
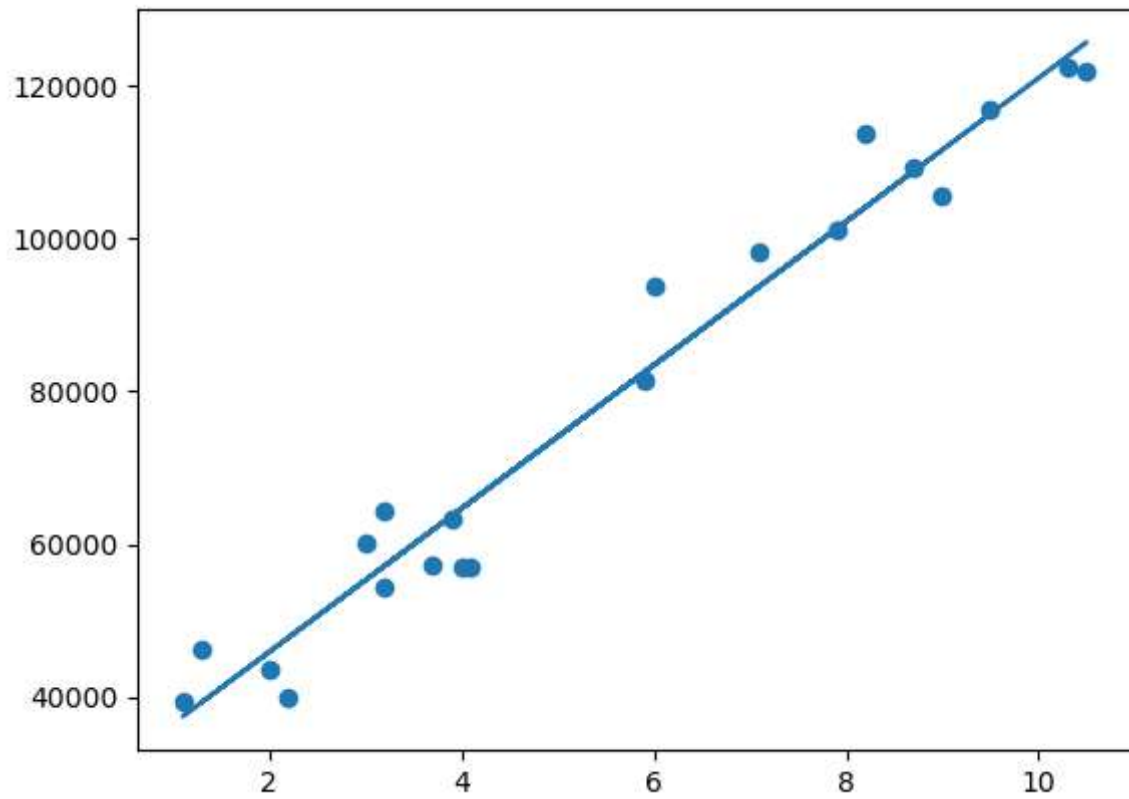
```
27038.831321426056
[9403.90716246]
```

In [36]:
```python
a=lm.intercept_
c=lm.coef_
```

In [37]:
```python
print(a,c)
```

```
27038.831321426056 [9403.90716246]
```

In [38]:
```python
my_sal_pred=a*10+c   # prediction of salary for 10 year exp
print(my_sal_pred)
```

```
[279792.22037672]
```

In [39]:
```python
# graph od train and test data
plt.scatter(X_train, y_train)
plt.plot(X_train, lm.predict(X_train))
```

Out[39]:
```
[<matplotlib.lines.Line2D at 0x29ef145a990>]
```

In [ ]: