Practicle:3 Linear Regression + Error Detection

2021BIT023
Vaibhav Lanjewar

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

Import libraries

In [22]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [23]:

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

In [24]:

```
df.head(2)
```

Out[24]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0

In [25]:

```
df.tail(2)
```

Out[25]:

	YearsExperience	Salary
28	10.3	122391.0
29	10.5	121872.0

In [26]:

```
df.isnull().sum()
```

Out[26]:

```
YearsExperience    0
Salary             3
dtype: int64
```

In [27]:

```
df.notnull().sum()
```

Out[27]:

```
YearsExperience    30
Salary             27
dtype: int64
```

In [28]:

```
df.isnull().sum()
```

Out[28]:

```
YearsExperience    0
Salary             3
dtype: int64
```

In [29]:

```
df.dropna(inplace=True)
```

In [30]:

```
df.corr() # correlation
```

Out[30]:

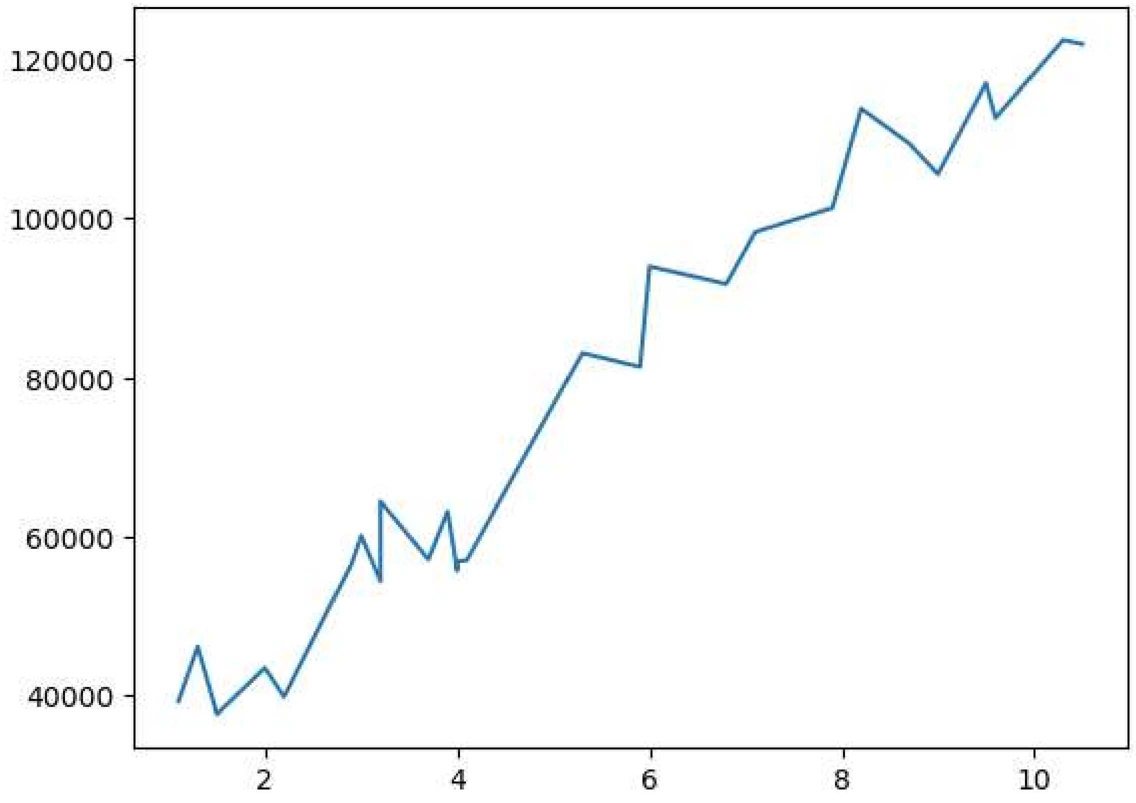
	YearsExperience	Salary
YearsExperience	1.00000	0.98131
Salary	0.98131	1.00000

In [31]:

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

In [32]:

```
# graph
plt.plot(X,y)
plt.show()
```



In [34]:

```
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.
```

```
WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=Non
e)) after connection broken by 'NewConnectionError('<pip._vendor.urllib3.connection.H
TTPSConnection object at 0x0000017E1EB93BD0>: Failed to establish a new connection:
[Errno 11001] getaddrinfo failed')': /simple/scikit-learn/
WARNING: Retrying (Retry(total=3, connect=None, read=None, redirect=None, status=Non
e)) after connection broken by 'NewConnectionError('<pip._vendor.urllib3.connection.H
TTPSConnection object at 0x0000017E220220D0>: Failed to establish a new connection:
[Errno 11001] getaddrinfo failed')': /simple/scikit-learn/
WARNING: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=Non
e)) after connection broken by 'NewConnectionError('<pip._vendor.urllib3.connection.H
TTPSConnection object at 0x0000017E2200E710>: Failed to establish a new connection:
[Errno 11001] getaddrinfo failed')': /simple/scikit-learn/
WARNING: Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=Non
e)) after connection broken by 'NewConnectionError('<pip._vendor.urllib3.connection.H
TTPSConnection object at 0x0000017E220598D0>: Failed to establish a new connection:
[Errno 11001] getaddrinfo failed')': /simple/scikit-learn/
WARNING: Retrying (Retry(total=0, connect=None, read=None, redirect=None, status=Non
e)) after connection broken by 'NewConnectionError('<pip._vendor.urllib3.connection.H
TTPSConnection object at 0x0000017E22058650>: Failed to establish a new connection:
[Errno 11001] getaddrinfo failed')': /simple/scikit-learn/
```

In [36]:
```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 [37]:
```python
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(X_train, y_train)
```

Out[37]:
```
▾ LinearRegression
LinearRegression()
```

In [39]:
```python
pred = lm.predict(X_test)
```

In [18]:
```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 [40]:
```python
pred
```

Out[40]:
```
array([ 41144.69206511, 117316.34008101,  76879.53928245,  90985.40002613,
        54310.16209255,  64654.45997125])
```

y=mx+c
y=ax+c

In [41]:
```
print(lm.intercept())
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[41], line 1
----> 1 print(lm.intercept())

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

In [43]:
```
a=lm.intercept_
c=lm.coef_
```

In [44]:
```
print(a,c)
```

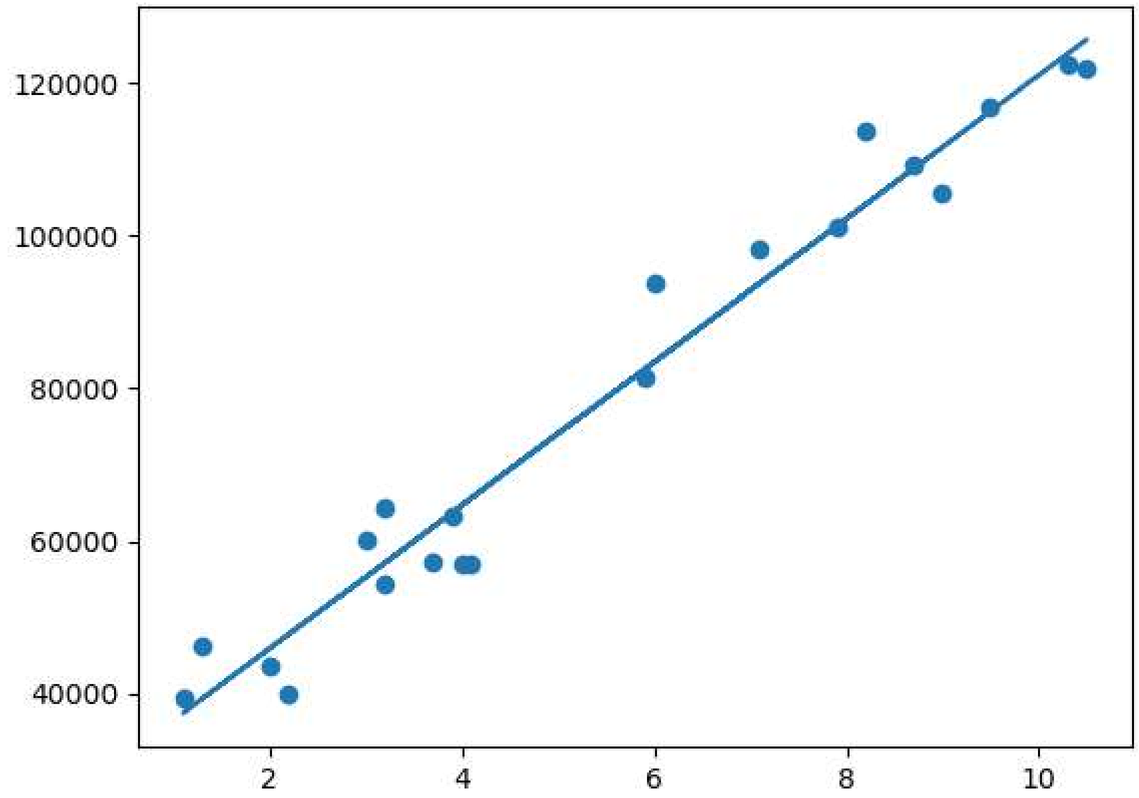
```
27038.831321426056 [9403.90716246]
```

In [45]:
```
my_sal_pred=a*10+c  # prediction of salary for 10 year exp
print(my_sal_pred)
```

```
[279792.22037672]
```

In [46]:
```
# graph od train and test data
plt.scatter(X_train, y_train)
plt.plot(X_train, lm.predict(X_train))
```

Out[46]:
```
[<matplotlib.lines.Line2D at 0x26e1cba60d0>]
```



Model Evaluation Metrix

In [47]:

```
from sklearn import metrics
```

In [49]:

```
print('Mean Absolute Error is : ',metrics.mean_absolute_error(y_test,pred))
```

```
Mean Absolute Error is:  4374.731786040949
```

In [51]:

```
print('Mean Squared Error is :',metrics.mean_squared_error(y_test,pred))
```

```
Mean Squared Error is 26104141.43339284
```

In [53]:

```
print('Root Mean Squared Error is: ',np.sqrt(metrics.mean_squared_error(y_test,pred)))
```

```
Root Mean Squared Error is:  5109.2212159381825
```

In [ ]:

```
#root Mean Squared Error (RMSE)
```

In [56]:

```
train_score_lm = lm.score(X_train, y_train)
test_score_lm = lm.score(X_test, y_test)

print("Train score: ", train_score_lm)
print("Test score : ",test_score_lm)
```

```
Train score:  0.9633907320629322
Test score :  0.9591199103412812
```

In [58]:

```
from sklearn.metrics import r2_score
print(" Root mean Squared error is:",r2_score(y_test,pred) )
r2=r2_score(y_test,pred)
```

```
Root mean Squared error is: 0.9591199103412812
```