In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
df=pd.read_csv("/content/sample_data/Salary_Data.csv")
```

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

Out[3]:

	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

In [ ]:
```
df.ndim
```

Out[4]:
```
2
```

In [ ]:
```
df.size
```

Out[5]:
```
60
```

In [ ]:
```
df.dtypes
```

Out[6]:
```
YearsExperience    float64
Salary             float64
dtype: object
```

In [ ]:
```
df.shape
```

Out[7]:
```
(30, 2)
```

In [ ]:
```
df.describe()
```

Out[9]:

	YearsExperience	Salary
count	30.000000	30.000000
mean	5.313333	76003.000000
std	2.837888	27414.429785
min	1.100000	37731.000000
25%	3.200000	56720.750000
50%	4.700000	65237.000000
75%	7.700000	100544.750000
max	10.500000	122391.000000

In [ ]:
```
df.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: 608.0 bytes
```

In [ ]:
```
df.columns
```

Out[11]:
```
Index(['YearsExperience', 'Salary'], dtype='object')
```

In [ ]:
```
df.nunique()
```

Out[12]:
```
YearsExperience    28
Salary             30
dtype: int64
```

In [ ]:
```
df["YearsExperience"].unique()
```

Out[13]:
```
array([ 1.1,  1.3,  1.5,  2. ,  2.2,  2.9,  3. ,  3.2,  3.7,  3.9,  4. ,
        4.1,  4.5,  4.9,  5.1,  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 [ ]:
```
df["Salary"].unique()
```

Out[14]:
```
array([ 39343.,  46205.,  37731.,  43525.,  39891.,  56642.,  60150.,
        54445.,  64445.,  57189.,  63218.,  55794.,  56957.,  57081.,
        61111.,  67938.,  66029.,  83088.,  81363.,  93940.,  91738.,
        98273., 101302., 113812., 109431., 105582., 116969., 112635.,
       122391., 121872.])
```

In [ ]:
```
df.duplicated()
```

Out[15]:
```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12    False
13    False
14    False
15    False
16    False
17    False
18    False
19    False
20    False
21    False
22    False
23    False
24    False
25    False
26    False
27    False
28    False
29    False
dtype: bool
```

In [ ]:
```
df.duplicated().sum()
```

Out[16]:
```
0
```

In [ ]:
```
df.drop_duplicates(inplace=True)
```

In [ ]:
```
df
```

Out[18]:

	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 [ ]:
```
df.isnull().sum()
```

Out[19]:
```
YearsExperience    0
Salary             0
dtype: int64
```

In [ ]:
```
df.isnull().any(axis=0)
```

Out[20]:
```
YearsExperience    False
Salary             False
dtype: bool
```

In [ ]:
```
df["YearsExperience"]
```

Out[23]:
```
0      1.1
1      1.3
2      1.5
3      2.0
4      2.2
5      2.9
6      3.0
7      3.2
8      3.2
9      3.7
10     3.9
11     4.0
12     4.0
13     4.1
14     4.5
15     4.9
16     5.1
17     5.3
18     5.9
19     6.0
20     6.8
21     7.1
22     7.9
23     8.2
24     8.7
25     9.0
26     9.5
27     9.6
28    10.3
29    10.5
Name: YearsExperience, dtype: float64
```

In [ ]:
```
df["Salary"]
```

Out[24]:
```
0      39343.0
1      46205.0
2      37731.0
3      43525.0
4      39891.0
5      56642.0
6      60150.0
7      54445.0
8      64445.0
9      57189.0
10     63218.0
11     55794.0
12     56957.0
13     57081.0
14     61111.0
15     67938.0
16     66029.0
17     83088.0
18     81363.0
19     93940.0
20     91738.0
21     98273.0
22    101302.0
23    113812.0
24    109431.0
25    105582.0
26    116969.0
27    112635.0
28    122391.0
29    121872.0
Name: Salary, dtype: float64
```

In [ ]:
```
features=df["YearsExperience"].values #feature (x)(dependent)
print(features)
```

```
[ 1.1  1.3  1.5  2.   2.2  2.9  3.   3.2  3.2  3.7  3.9  4.   4.   4.1
  4.5  4.9  5.1  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 [ ]:
```
labels=df["Salary"].values #labels (y) (independent)
print(labels)
```

```
[ 39343.  46205.  37731.  43525.  39891.  56642.  60150.  54445.  64445.
  57189.  63218.  55794.  56957.  57081.  61111.  67938.  66029.  83088.
  81363.  93940.  91738.  98273. 101302. 113812. 109431. 105582. 116969.
 112635. 122391. 121872.]
```

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

Out[27]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0

In [ ]:
```
features.shape
```

Out[28]:
```
(30,)
```

In [ ]:
```
features.ndim
```

Out[29]:
```
1
```

In [ ]:
```
features.size
```

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

In [ ]:
```
labels.shape
```

Out[33]:
```
(30,)
```

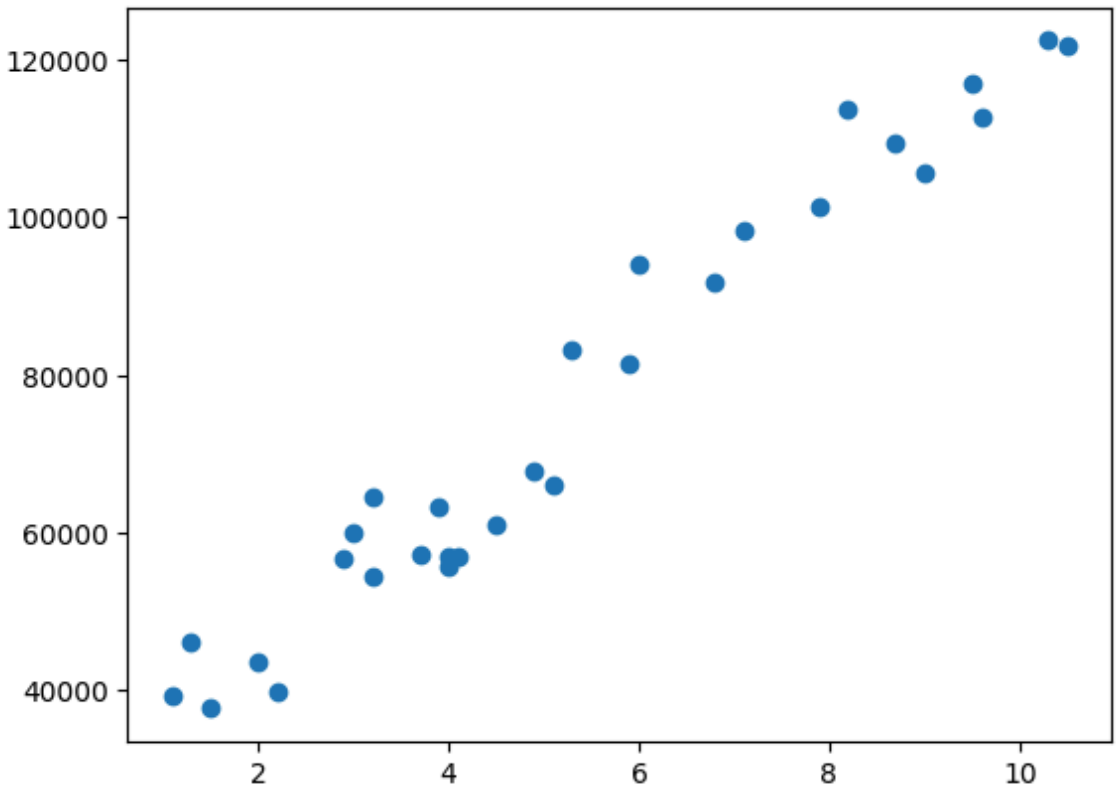
In [ ]:
```
labels.size
```

Out[34]:
```
30
```

In [ ]:
```
#representing the labels and features values within the scatter plot
```

In [ ]:
```
plt.scatter(features,labels)
```

Out[36]:
```
<matplotlib.collections.PathCollection at 0x78d5329a56f0>
```



In [ ]:
```
#now implement the Linear Regression to find the best fit line
```

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

In [ ]:
```
reg=LinearRegression()
```

In [ ]:
```
features=features.reshape(30,1)
```

In [ ]:
```
features.ndim
```

Out[41]:
```
2
```

In [ ]:
```
#now fit the Linear Regression Model
```

In [ ]:
```
reg.fit(features,labels)
```

Out[43]:
```
LinearRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [ ]:
```
m=reg.coef_  #coefficient
m
```

Out[45]:
```
array([9449.96232146])
```

In [ ]:
```
c=reg.intercept_ #intercept
c
```

Out[47]:
```
25792.200198668696
```

In [ ]:
```
x=3
y=m*x+c
```

In [ ]:
```
y
```

Out[49]:
```
array([54142.08716303])
```

In [ ]:
```
reg.predict([[3]])
```

Out[50]:
```
array([54142.08716303])
```

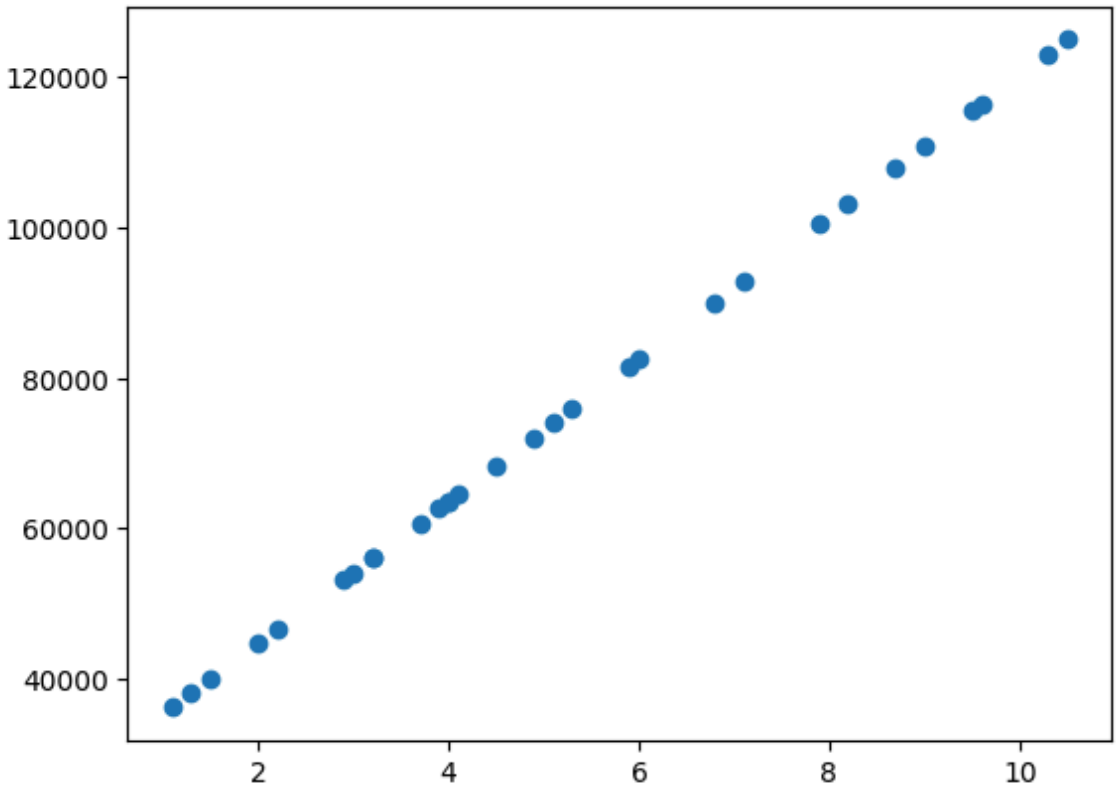
In [ ]:
```
reg.predict([[3.2]])
```

Out[51]:
```
array([56032.07962732])
```

In [ ]:
```
#now print the best fit line
```

In [ ]:
```
plt.scatter(features,reg.predict(features))
```

Out[53]:
```
<matplotlib.collections.PathCollection at 0x78d531fdf040>
```



In [ ]:
```
reg.predict(([[5.2]]))
```

Out[55]:
```
array([74932.00427024])
```

In [ ]:
```
#now train the model and find the score of the model (accuracy)
```

In [ ]:
```
from sklearn.model_selection import train_test_split
features_train,features_test,labels_train,labels_test=train_test_split(feat
```

In [ ]:
```
features_train.ndim
```

Out[58]:
```
2
```

In [ ]:
```
features_test.ndim
```

Out[59]:
```
2
```

In [ ]:
```
labels_train.ndim
```

Out[60]:
```
1
```

In [ ]:
```
labels_test.ndim
```

Out[61]:
```
1
```

In [ ]:
```
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
```

In [ ]:
```
reg.fit(features_train,labels_train)
```

Out[63]:
```
LinearRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [ ]:
```
reg.predict(features_test)
```

Out[64]:
```
array([116361.74135117, 117295.47150157,  75277.61473337,  54735.55142448,
        82747.45593661,  65940.31322933,  65006.58307892,  48199.4403716
4])
```

In [ ]:
```
labels_test
```

Out[65]:
```
array([116969., 112635.,  66029.,  56642.,  81363.,  57081.,  56957.,
        39891.])
```

In [ ]:
```
labels_train
```

Out[66]:
```
array([ 60150.,  55794.,  91738., 101302.,  67938., 122391.,  61111.,
        54445., 113812.,  43525.,  63218., 109431., 121872.,  83088.,
        46205.,  64445.,  39343.,  98273.,  57189.,  37731.,  93940.,
       105582.])
```

In [ ]:
```
pred=reg.predict(features_test)
```

In [ ]:
```
pred
```

Out[68]:
```
array([116361.74135117, 117295.47150157,  75277.61473337,  54735.55142448,
        82747.45593661,  65940.31322933,  65006.58307892,  48199.4403716
4])
```

In [ ]:
```
pd.DataFrame(zip(pred,labels_test))
```

Out[71]:

	0	1
0	116361.741351	116969.0
1	117295.471502	112635.0
2	75277.614733	66029.0
3	54735.551424	56642.0
4	82747.455937	81363.0
5	65940.313229	57081.0
6	65006.583079	56957.0
7	48199.440372	39891.0

In [ ]:
```
#training Data Score (Accuracy)
```

In [ ]:
```
reg.score(features_train,labels_train)
```

Out[73]:
```
0.9590995552307341
```

In [ ]:
```
#testing Data Score (Accuracy)
```

In [ ]:
```
reg.score(features_test,labels_test)
```

Out[75]:
```
0.9407739954625256
```

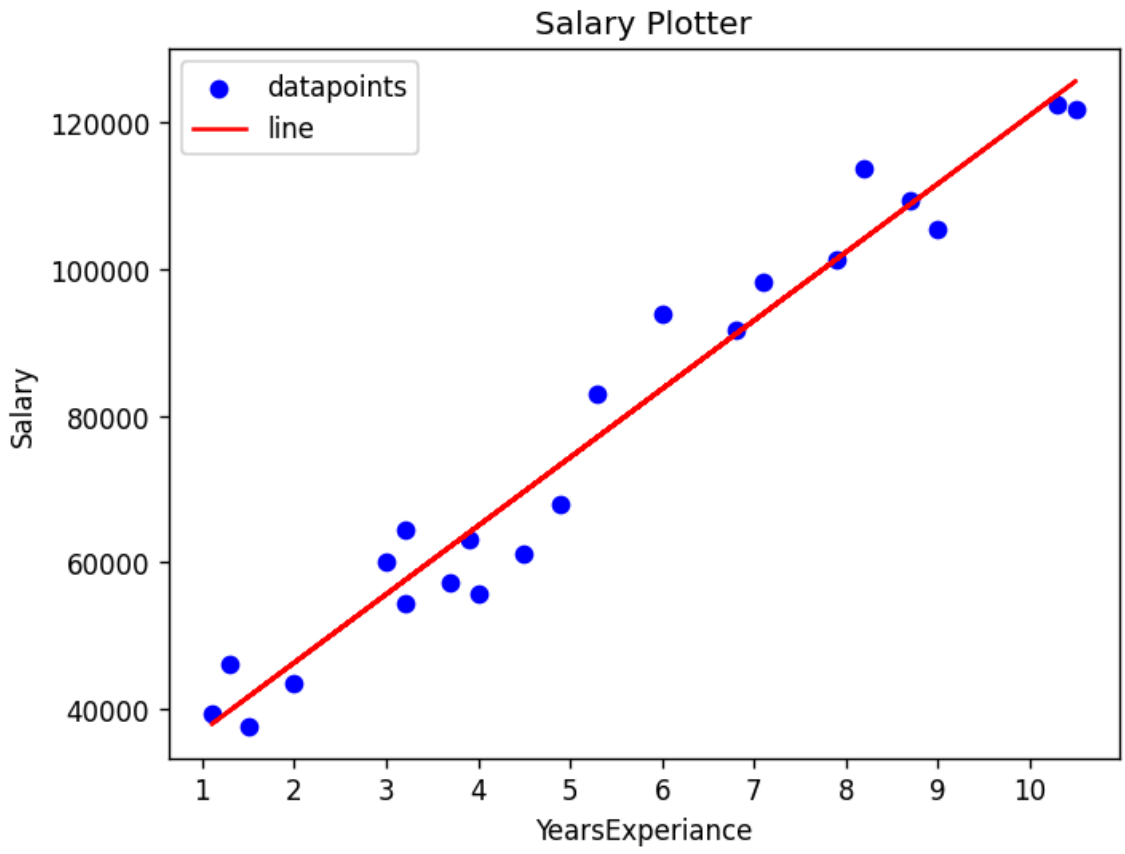
In [ ]:
```
score=reg.predict([[10.4]])
print(score)
```

```
[124765.31270481]
```

In [ ]:
```
#visulization
```

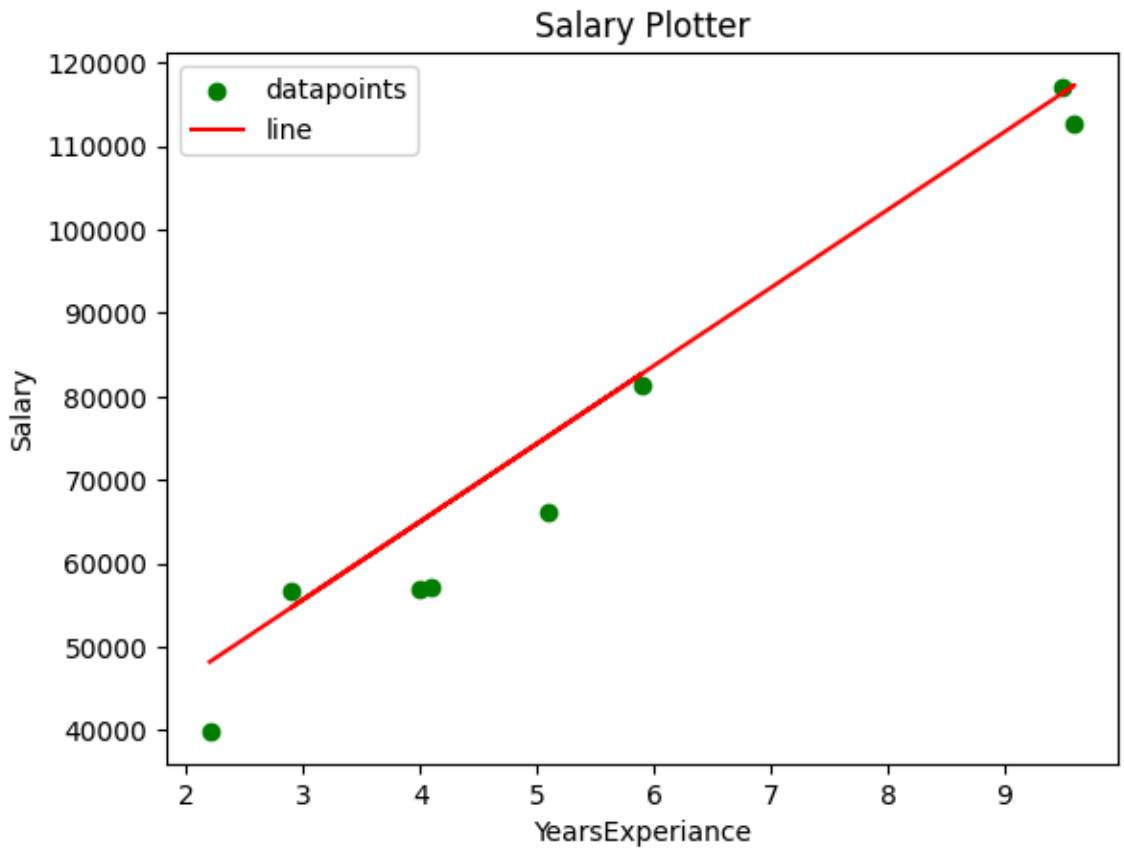
In [ ]:
```
plt.figure(dpi=120)
plt.title("Salary Plotter")
plt.ylabel("Salary")
plt.xlabel("YearsExperiance")
plt.xticks(range(0,12))
plt.scatter(features_train,labels_train, color="blue",label="datapoints")
plt.plot(features_train,reg.predict(features_train),color="red",label="line
plt.legend()
```

Out[78]:
```
<matplotlib.legend.Legend at 0x78d531ff6380>
```



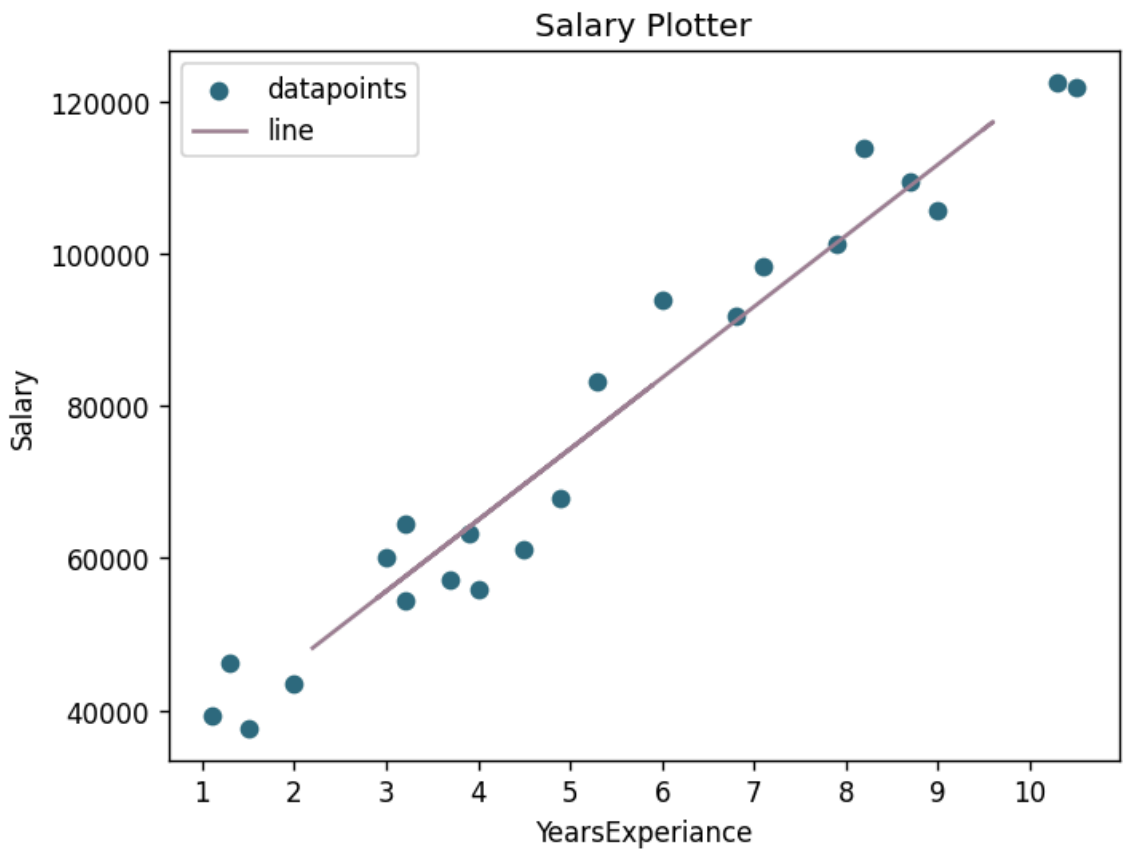
In [ ]:
```
plt.figure(dpi=100)
plt.title("Salary Plotter")
plt.ylabel("Salary")
plt.xlabel("YearsExperiance")
plt.xticks(range(0,12))
plt.scatter(features_test,labels_test, color="green",label="datapoints")
plt.plot(features_test,reg.predict(features_test),color="red",label="line")
plt.legend()
```

Out[79]: <matplotlib.legend.Legend at 0x78d5311170d0>



In [ ]:
```
plt.figure(dpi=120)
plt.title("Salary Plotter")
plt.ylabel("Salary")
plt.xlabel("YearsExperiance")
plt.xticks(range(0,12))
plt.scatter(features_train,labels_train, color="#2d6d7d",label="datapoints"
plt.plot(features_test,reg.predict(features_test),color="#9c8194",label="li
plt.legend()
```

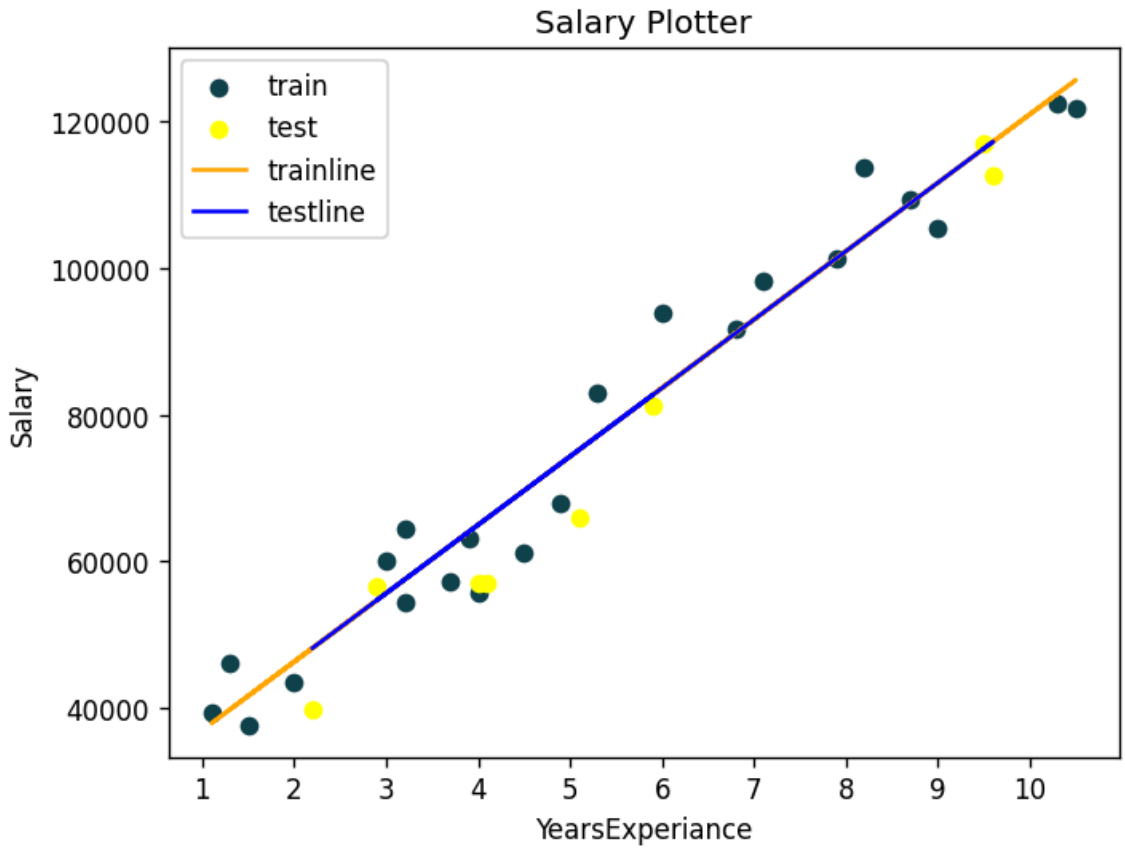
Out[80]: <matplotlib.legend.Legend at 0x78d531485660>



In [ ]:
```
plt.figure(dpi=120)
plt.title("Salary Plotter")
plt.ylabel("Salary")
plt.xlabel("YearsExperiance")
plt.xticks(range(0,12))
plt.scatter(features_train,labels_train, color="#12414d",label="train")
plt.scatter(features_test,labels_test,color="yellow",label="test")
plt.plot(features_train,reg.predict(features_train),color="orange",label="t
plt.plot(features_test,reg.predict(features_test),color="blue",label="testl

plt.legend()
```

Out[81]:
```
<matplotlib.legend.Legend at 0x78d53109fa90>
```



In [ ]:
```
c1=features_test.ravel()
c2=labels_test.ravel()
c3=reg.predict(features_test).ravel()
c4=(labels_test-reg.predict(features_test)).ravel()
df2=pd.DataFrame({"YearsExperiance":c1,"ActualValue":c2,"PredictedValue":c3
```

In [ ]:
```
print(df2)
```

```
   YearsExperiance  ActualValue  PredictedValue  Differences
0              9.5     116969.0   116361.741351   607.258649
1              9.6     112635.0   117295.471502 -4660.471502
2              5.1      66029.0    75277.614733 -9248.614733
3              2.9      56642.0    54735.551424  1906.448576
4              5.9      81363.0    82747.455937 -1384.455937
5              4.1      57081.0    65940.313229 -8859.313229
6              4.0      56957.0    65006.583079 -8049.583079
7              2.2      39891.0    48199.440372 -8308.440372
```

In [ ]: