```
import pandas as pd
from sklearn import linear_model, metrics
```

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

```
df.head()
```

```
   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
```

Next steps: Generate code with df | View recommended plots | New interactive sheet

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

```
                  0
YearsExperience   0
         Salary   0

dtype: int64
```

```
df.columns
```

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

```
#split the dataset
x = df.iloc[:,:-1].values
y = df.iloc[:,-1].values
```

```
x
```

```
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],
       [ 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]])
```

```
y
```

```
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.])
```

```
#Diving dataset into training and testing ( Data Preparation)
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.4,random_state
```

```
print("size of: \n")
print("x_train: ", x_train.shape)
print("y_train: ", y_train.shape)
print("x_test: ", x_test.shape)
print("y_test: ", y_test.shape)
```

```
size of: 

x_train:  (18, 1)
y_train:  (18,)
x_test:  (12, 1)
y_test:  (12,)
```

```
#model selection
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(x_train,y_train)
```

▾ LinearRegression ⓘ ?

```
LinearRegression()
```

```
print("model coefficient (slop = ) :" , model.coef_)
print("model intercept : ", model.intercept_)
```

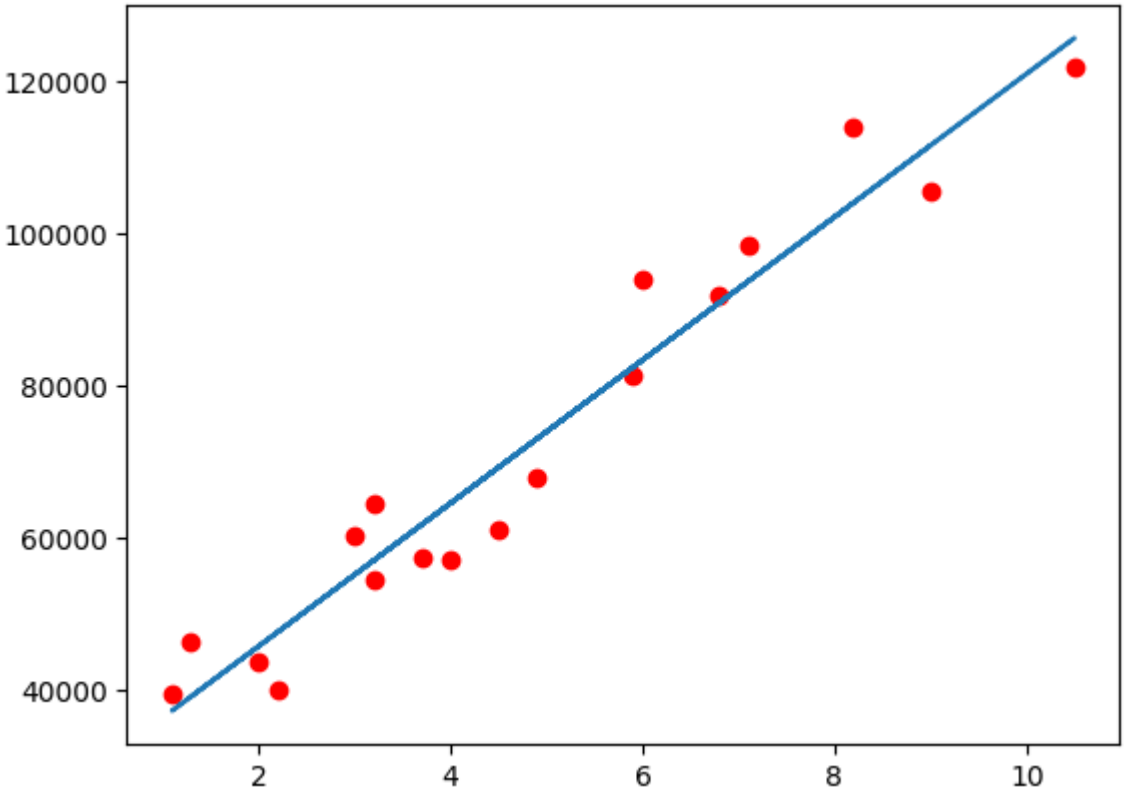
```
model coefficient (slop = ) : [9407.35651165]
model intercept :  26838.99589395697
```

```
#model predictions
predicted = model.predict(x_test)
print("Predicted Salaries :", predicted)
print("Actual Salaries :", y_test)
```

```
Predicted Salaries : [ 40950.03066143 123734.76796394  65409.15759172  6352
 116208.88275462 108682.9975453  117149.61840578  64468.42194055
  76697.98540569 101157.11233598  54120.32977774  74816.51410337]
Actual Salaries : [ 37731. 122391.  57081.  63218. 116969. 109431. 112635.
 101302.  56642.  66029.]
```

```
#line plotting
import matplotlib.pyplot as plt
plt.scatter(x_train,y_train,color='red')
plt.plot(x_train,model.predict(x_train))
```

```
[<matplotlib.lines.Line2D at 0x7c2eaf0ab7f0>]
```



```
model
```

```
▾ LinearRegression ⓘ ?
```

```
LinearRegression()
```

```
#calculate the matrics here
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

```
print( f"Mean Squared Error: {mean_squared_error(y_test,predicted)}")
```

```
Mean Squared Error: 25234929.18660526
```

```
print( f"Mean Absolute Error: {mean_absolute_error(y_test,predicted)}")
```

```
Mean Absolute Error: 3811.8240947365243
```

```
print( f"r2 score: {r2_score(y_test,predicted)}")
```

```
r2 score: 0.9679117063698979
```

Start coding or generate with AI.