ASSIGNMENT - LINEAR REGRESSION

Importing Libraries

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

Data

In [2]:
```
df=pd.read_csv(r"C:\Users\alika\Downloads\Salary_dataset.csv")
df.head(5)
```

Out[2]:

	Unnamed: 0	YearsExperience	Salary
0	0	1.2	39344.0
1	1	1.4	46206.0
2	2	1.6	37732.0
3	3	2.1	43526.0
4	4	2.3	39892.0

Dropping unnecessary columns

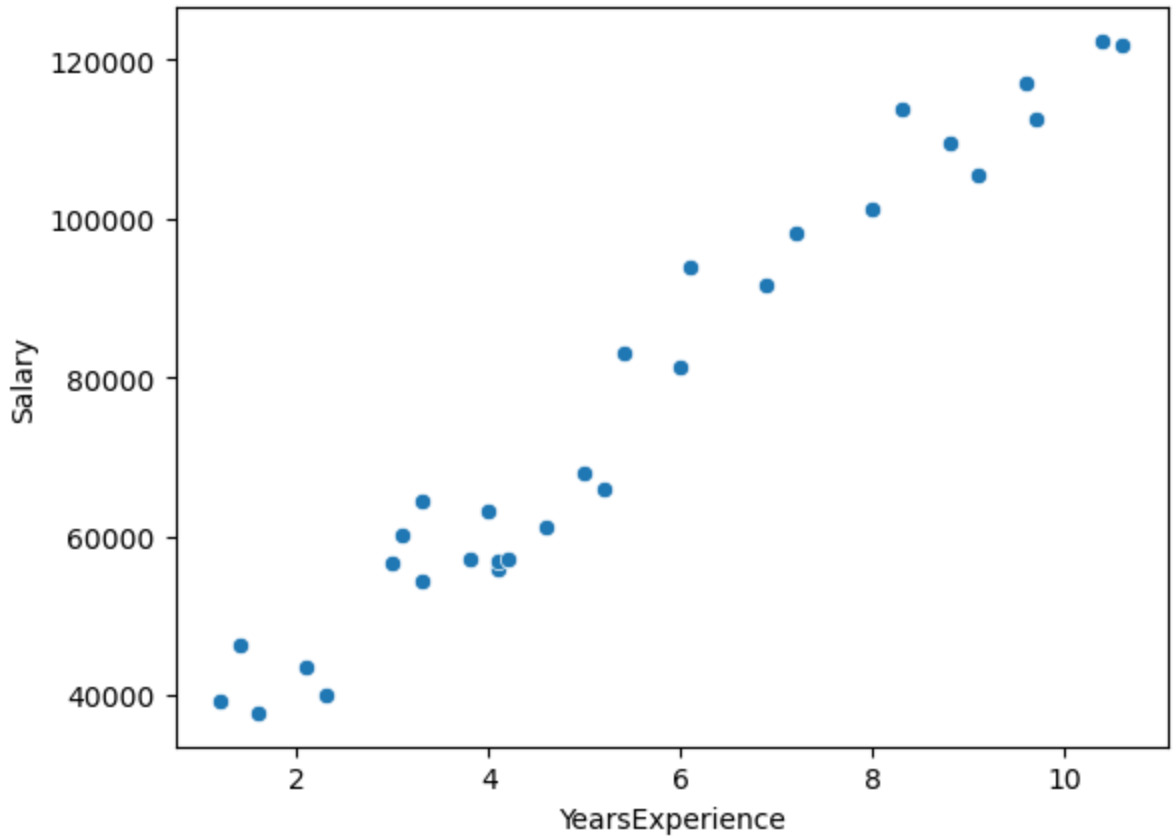
In [3]:
```
df=df.drop(columns='Unnamed: 0', axis=1)
df.head(5)
```

Out[3]:

	YearsExperience	Salary
0	1.2	39344.0
1	1.4	46206.0
2	1.6	37732.0
3	2.1	43526.0
4	2.3	39892.0

Scatterplot

In [4]:
```
sns.scatterplot(x='YearsExperience',y='Salary',data=df)
plt.show()
```



Test vs Train Data

In [5]:
```
df_train,df_test=train_test_split(df,test_size=0.2,random_state=72)
```

In [6]:
```
df_train
```

Out[6]:

	YearsExperience	Salary
27	9.7	112636.0
9	3.8	57190.0
2	1.6	37732.0
11	4.1	55795.0
21	7.2	98274.0
0	1.2	39344.0
26	9.6	116970.0
1	1.4	46206.0
29	10.6	121873.0
28	10.4	122392.0
18	6.0	81364.0
16	5.2	66030.0
8	3.3	64446.0
12	4.1	56958.0
13	4.2	57082.0
7	3.3	54446.0
23	8.3	113813.0
15	5.0	67939.0
25	9.1	105583.0
5	3.0	56643.0
10	4.0	63219.0
14	4.6	61112.0
19	6.1	93941.0
24	8.8	109432.0

In [7]:
```
df_test
```

Out[7]:

	YearsExperience	Salary
22	8.0	101303.0
20	6.9	91739.0
6	3.1	60151.0
3	2.1	43526.0
17	5.4	83089.0
4	2.3	39892.0

Linear Regression

In [8]:
```
from sklearn.linear_model import LinearRegression
```

In [9]:
```
model=LinearRegression()
```

In [10]:
```
df_input_train=df_train[['YearsExperience']]
df_target_train=df_train['Salary']
```

In [11]:
```
model.fit(df_input_train,df_target_train)
```

Out[11]:
```
LinearRegression()
```

In [12]:
```
df_input_test=df_test[['YearsExperience']]
df_input_test
```

Out[12]:

	YearsExperience
22	8.0
20	6.9
6	3.1
3	2.1
17	5.4
4	2.3

Prediction

In [13]:
```
prediction=model.predict(df_input_test)
prediction
```

Out[13]:
```
array([99950.73980663, 89633.01278755, 53989.95581256, 44610.20397704,
       75563.38503427, 46486.15434414])
```

In [14]:
```
df_test_target=df_test['Salary'].tolist()
df_test_target
```

Out[14]:
```
[101303.0, 91739.0, 60151.0, 43526.0, 83089.0, 39892.0]
```

In [15]:
```
from sklearn.metrics import mean_squared_error, r2_score
```

In [16]:
```
mse=mean_squared_error(prediction,df_test_target)
mse
```

Out[16]:
```
24252584.27330477
```

Root Mean Squared Error

In [17]:
```
rmse=np.sqrt(mse)
rmse
```

Out[17]:
```
4924.6912871067125
```

R-Squared

In [18]:
```
r2=r2_score(df_test_target,prediction)
r2
```

Out[18]:
```
0.9562771755752736
```

In [19]:
```
prediction=prediction.tolist()
```

In [20]:
```
prediction
```

Out[20]:
```
[99950.73980662727,
 89633.01278755131,
 53989.955812561646,
 44610.20397703805,
 75563.38503426593,
 46486.15434414278]
```

In [21]:
```
df_test_target
```

Out[21]:
```
[101303.0, 91739.0, 60151.0, 43526.0, 83089.0, 39892.0]
```

Comparison

In [22]:
```
comparison=pd.DataFrame({'actual': df_test_target, 'prediction': prediction})
comparison
```

Out[22]:

	actual	prediction
0	101303.0	99950.739807
1	91739.0	89633.012788
2	60151.0	53989.955813
3	43526.0	44610.203977
4	83089.0	75563.385034
5	39892.0	46486.154344

In [23]:
```
list=[]
```

In [24]:
```
for i in range (len(comparison)):
    list.append(i+1)
```

In [25]:
```
list
```

Out[25]:
```
[1, 2, 3, 4, 5, 6]
```

In [26]:
```
comparison['s.no.']=list
```

In [27]:
```
comparison
```

Out[27]:

	actual	prediction	s.no.
0	101303.0	99950.739807	1
1	91739.0	89633.012788	2
2	60151.0	53989.955813	3
3	43526.0	44610.203977	4
4	83089.0	75563.385034	5
5	39892.0	46486.154344	6

Plotting Actual vs Predicted values

In [28]:
```
sns.barplot(x='s.no.', y='actual', data=comparison, color='blue', label='Actual', a
sns.barplot(x='s.no.', y='prediction', data=comparison, color='green', label='Predi
```

Out[28]:
```
<Axes: xlabel='s.no.', ylabel='prediction'>
```