Problem Statement

Predicting Employee Salary Based on Experience

Problem Description

In the corporate world, employee compensation is a crucial factor for both the employers and the employees. Determining a fair and competitive salary based on an employee's experience is important for maintaining job satisfaction, motivation, and retention. This dataset contains data on employees years of experience and their corresponding salaries

Objective

The objective of this analysis is to build a predictive model that can accurately forecast an employee's salary based on their years of experience. This model will help in understanding the salary trends related to experience and assist companies in establishing fair compensation practices.

Imports

In [2]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

Get the Data

In [3]:
```python
salary=pd.read_csv("salary_exp.csv")
salary
```

Out[3]:

	Experience Years	Salary
0	1.1	39343
1	1.2	42774
2	1.3	46205
3	1.5	37731
4	2.0	43525
5	2.2	39891
6	2.5	48266
7	2.9	56642
8	3.0	60150
9	3.2	54445
10	3.2	64445
11	3.5	60000
12	3.7	57189
13	3.8	60200
14	3.9	63218
15	4.0	55794
16	4.0	56957
17	4.1	57081
18	4.3	59095
19	4.5	61111
20	4.7	64500
21	4.9	67938
22	5.1	66029
23	5.3	83088
24	5.5	82200
25	5.9	81363
26	6.0	93940
27	6.2	91000
28	6.5	90000
29	6.8	91738
30	7.1	98273
31	7.9	101302
32	8.2	113812
33	8.5	111620
34	8.7	109431
35	9.0	105582
36	9.5	116969
37	9.6	112635
38	10.3	122391
39	10.5	121872

Data description

1.Experience_Years: Number of years of experience the employee has. 2.Salary: Salary of the employee (in dollars).

Intial Check Up

checkout head

In [4]:
```python
salary.head()
```

Out[4]:

	Experience Years	Salary
0	1.1	39343
1	1.2	42774
2	1.3	46205
3	1.5	37731
4	2.0	43525

check out tail

In [7]:
```python
salary.tail()
```

Out[7]:

	Experience Years	Salary
35	9.0	105582
36	9.5	116969
37	9.6	112635
38	10.3	122391
39	10.5	121872

In [ ]:

In [8]:
```python
salary.shape
```

Out[8]: (40, 2)

we have 40 rows of data and 2 columnscheck out description

In [10]:
```python
salary.describe()
```

Out[10]:

	Experience Years	Salary
count	40.000000	40.000000
mean	5.152500	74743.625000
std	2.663715	25947.122885
min	1.100000	37731.000000
25%	3.200000	56878.250000
50%	4.600000	64472.500000
75%	6.875000	95023.250000
max	10.500000	122391.000000

The total count of years of experience is 40. Mean is 5, Minimum is 1 yaer and Maximum is 10 years.check out information

In [11]:
```python
salary.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40 entries, 0 to 39
Data columns (total 2 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Experience Years  40 non-null     float64
 1   Salary            40 non-null     int64
dtypes: float64(1), int64(1)
memory usage: 772.0 bytes
```

Data Analysis Asking Questions 1.Maximum experience 2.Minimum experience 3.Highest salary 4.Lowest salary 5.Years wise salary 1.Maximum experience

In [12]:
```python
salary["Experience Years"].max
```

Out[12]:
```
<bound method Series.max of 0      1.1
1      1.2
2      1.3
3      1.5
4      2.0
5      2.2
6      2.5
7      2.9
8      3.0
9      3.2
10     3.2
11     3.5
12     3.7
13     3.8
14     3.9
15     4.0
16     4.0
17     4.1
18     4.3
19     4.5
20     4.7
21     4.9
22     5.1
23     5.3
24     5.5
25     5.9
26     6.0
27     6.2
28     6.5
29     6.8
30     7.1
31     7.9
32     8.2
33     8.5
34     8.7
35     9.0
36     9.5
37     9.6
38    10.3
39    10.5
Name: Experience Years, dtype: float64>
```

2.Minimum experience

In [16]:
```python
salary['Experience Years'].min
```

Out[16]:
```
<bound method Series.min of 0      1.1
1      1.2
2      1.3
3      1.5
4      2.0
5      2.2
6      2.5
7      2.9
8      3.0
9      3.2
10     3.2
11     3.5
12     3.7
13     3.8
14     3.9
15     4.0
16     4.0
17     4.1
18     4.3
19     4.5
20     4.7
21     4.9
22     5.1
23     5.3
24     5.5
25     5.9
26     6.0
27     6.2
28     6.5
29     6.8
30     7.1
31     7.9
32     8.2
33     8.5
34     8.7
35     9.0
36     9.5
37     9.6
38    10.3
39    10.5
Name: Experience Years, dtype: float64>
```

Histplot

In [17]:
```python
sns.histplot(salary,x='Experience Years')
```

```
C:\Users\ajaykanth\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[17]: <Axes: xlabel='Experience Years', ylabel='Count'>

In [ ]:
```python
## The 5 years of experience is highest where the 7 years of experience is low.
```

3.Highest salary

In [19]:
```python
salary['Salary'].max()
```

Out[19]: 122391

4.Lowest salary

In [20]:
```python
salary['Salary'].min()
```

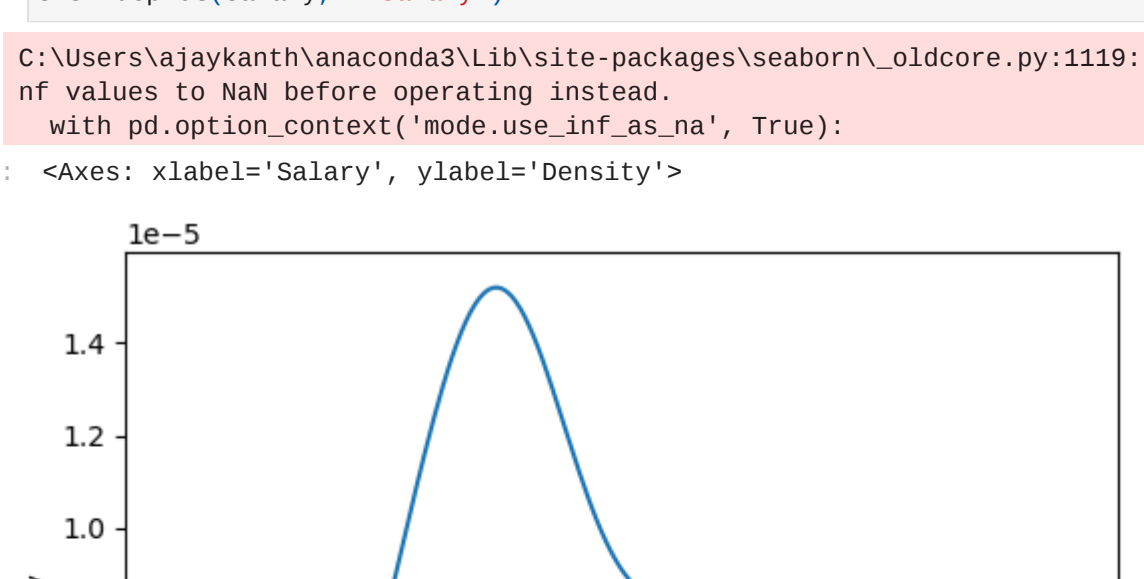
Out[20]: 37731

kdeplot

In [22]:
```python
sns.kdeplot(salary,x='Salary')
```

```
C:\Users\ajaykanth\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

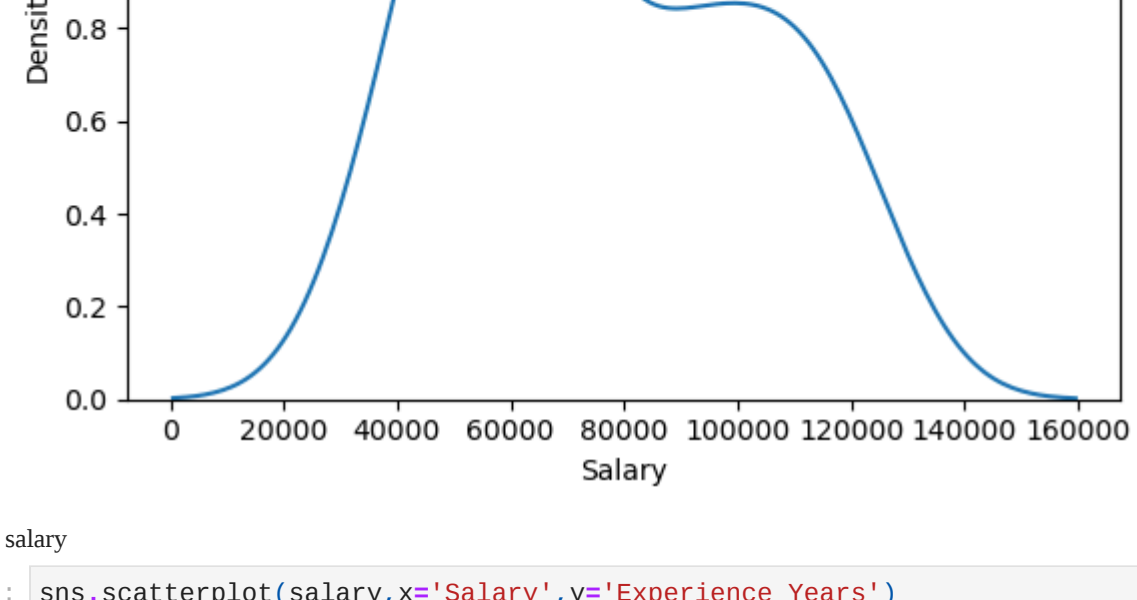
Out[22]: <Axes: xlabel='Salary', ylabel='Density'>



5.Years wise salary

In [25]:
```python
sns.scatterplot(salary,x='Salary',y='Experience Years')
```

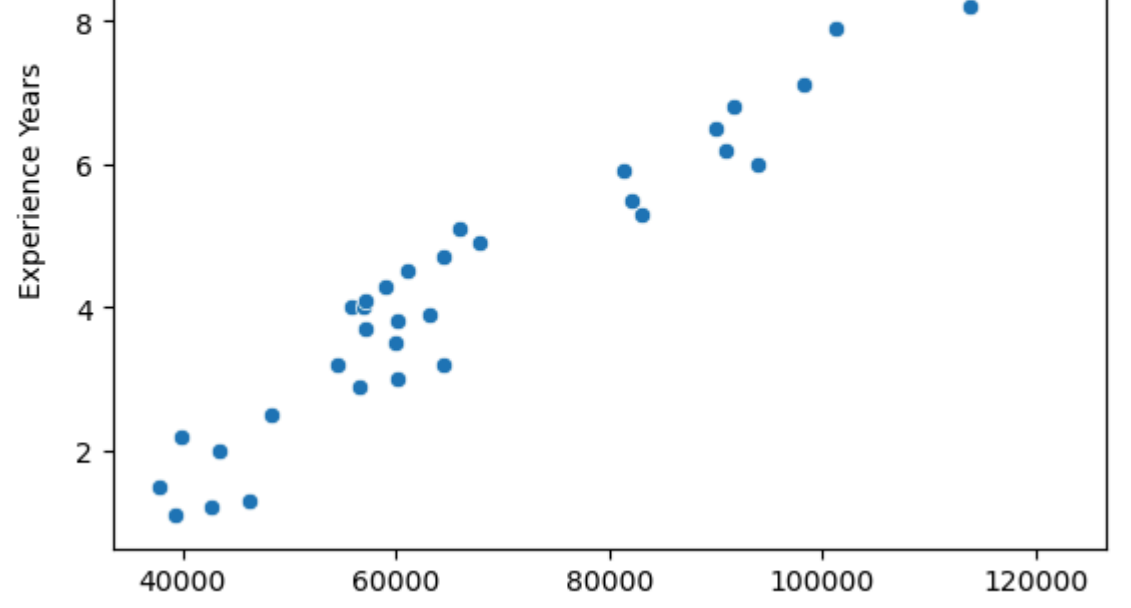
Out[25]: <Axes: xlabel='Salary', ylabel='Experience Years'>



Data Analysis

In [26]:
```python
sns.scatterplot(salary,x='Salary',y='Experience Years')
```

Out[26]: <Axes: xlabel='Salary', ylabel='Experience Years'>



In [ ]:
```python
## From the above visualization data is linear positive slope.
```

Correlation

In [27]:
```python
salary.corr()
```

Out[27]:

	Experience Years	Salary
Experience Years	1.000000	0.977692
Salary	0.977692	1.000000

One Dimention Cloumns

In [29]:
```python
X=salary.iloc[:,0:1]
y=salary.iloc[:,-1]
```

In [30]:
```python
X.head()
```

Out[30]:

	Experience Years
0	1.1
1	1.2
2	1.3
3	1.5
4	2.0

In [31]:
```python
y.head()
```

Out[31]:
```
0    39343
1    42774
2    46205
3    37731
4    43525
Name: Salary, dtype: int64
```

Train and Test

In [32]:
```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=2)
```

In [34]:
```python
X_train.head()
```

Out[34]:

	Experience Years
17	4.1
37	9.6
38	10.3
29	6.8
24	5.5

In [35]:
```python
X_train.shape
```

Out[35]: (32, 1)

In [36]:
```python
X_test.head()
```

Out[36]:

	Experience Years
27	6.2
9	3.2
14	3.9
0	1.1
2	1.3

In [37]:
```python
X_test.shape
```

Out[37]: (8, 1)

In [38]:
```python
y_train.head()
```

Out[38]:
```
17     57081
37    112635
38    122391
29     91738
24     82200
Name: Salary, dtype: int64
```

In [39]:
```python
y_train.shape
```

Out[39]: (32,)

In [40]:
```python
y_test.head()
```

Out[40]:
```
27    91000
9     54445
14    63218
0     39343
2     46205
Name: Salary, dtype: int64
```

In [41]:
```python
y_test.shape
```

Out[41]: (8,)

Model Building

In [42]:
```python
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
```

fit() method

In [43]:
```python
lr.fit(X_train,y_train)
```

Out[43]:
```
▾ LinearRegression
LinearRegression()
```

Coefficient

In [44]:
```python
lr.coef_
```

Out[44]: array([9629.89561636])

Interception

In [45]:
```python
lr.intercept_
```

Out[45]: 24469.054538114055

Prediction

In [46]:
```python
lr.predict([[6.9]])
```

```
C:\Users\ajaykanth\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[46]: array([90915.33429096])

In [47]:
```python
lr.predict([[10]])
```

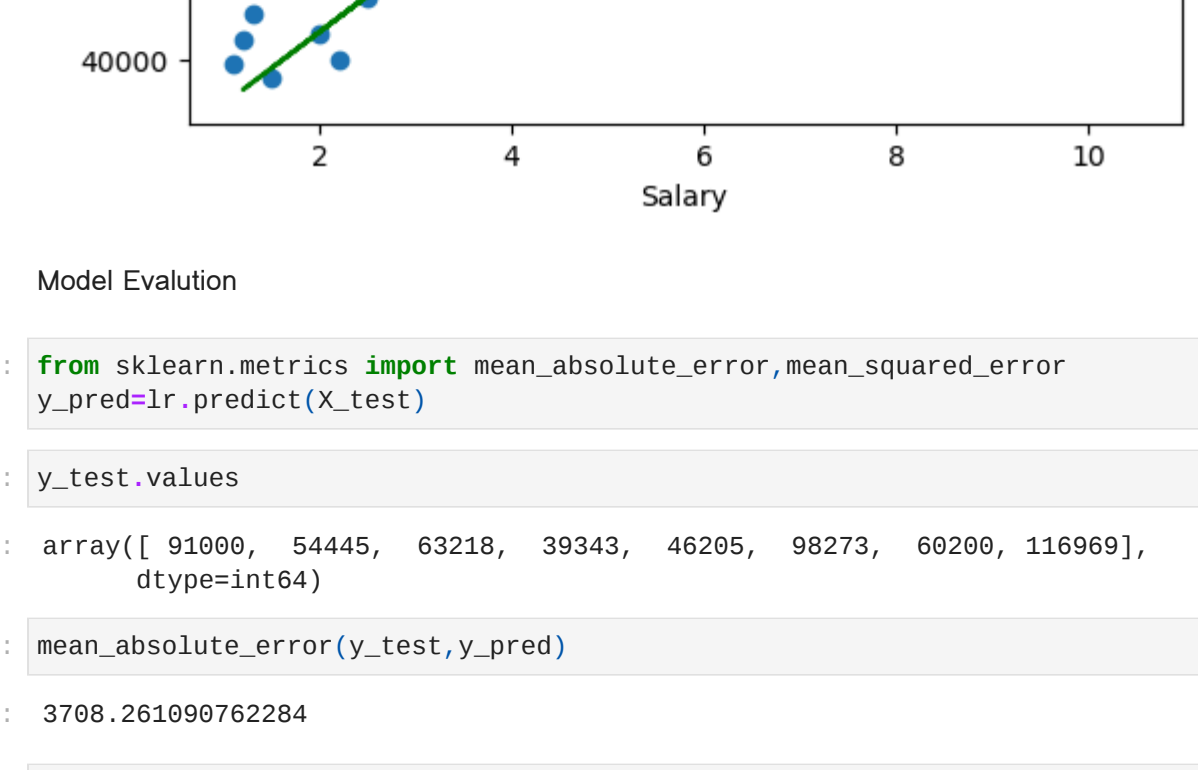
```
C:\Users\ajaykanth\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[47]: array([120768.01070166])

Linear Plot Marking

In [49]:
```python
plt.scatter(salary['Experience Years'],salary['Salary'])
plt.plot(X_train,lr.predict(X_train),color='green')
plt.xlabel('Salary')
plt.ylabel('Experience Years')
```

Out[49]: Text(0, 0.5, 'Experience Years')



Model Evalution

In [50]:
```python
from sklearn.metrics import mean_absolute_error,mean_squared_error
y_pred=lr.predict(X_test)
```

In [51]:
```python
y_test.values
```

Out[51]:
```
array([ 91000,  54445,  63218,  39343,  46205,  98273,  60200, 116969],
      dtype=int64)
```

In [52]:
```python
mean_absolute_error(y_test,y_pred)
```

Out[52]: 3708.261090762284

In [53]:
```python
mean_squared_error(y_test,y_pred)
```

Out[53]: 22909642.289620496

Accurate Prediction

In [54]:
```python
from sklearn.metrics import mean_squared_error
MSE=mean_squared_error(y_test,y_pred)
RMSE=MSE*0.5
data_rmse={'Actual(y_test)':y_test,'predicted(y_test)':y_pred}
df_rmse=pd.DataFrame(data_rmse)
df_rmse.head()
```

Out[54]:

	Actual(y_test)	predicted(y_test)
27	91000	84174.407360
9	54445	55284.720510
14	63218	62025.647442
0	39343	35061.939716
2	46205	36987.918839

In [ ]:
```
INSIGTHS
1.First we had imported the necessary libraries for our program i.e pandas,numpy, seaborn and matplotlib.
2. Next we had imported the salary_exp csv y using pandas.
3.We had intial check up for our data set like head,tail,description and information.
4.the dataset has 40 rows of data and 2 columns( salary and experienced years).
5.The total count of years of experience is 40. Mean is 5, Minimum is 1 yaer and Maximum is 10 years.
6.As for our data analysis the maximun experienced years is 39 and minimum experienced years is 1.
7.The 5 years of experience is highest where the 7 years of experience is low.
8. The highest salary is 12 lakhs and the lowest salary is 37 thousand.
9.The count of 50000 thousand salary is high.
10.By using sacatter plot the value of x increases as the value of y also increases we can say that our data is linear.
11.By using iloc() one dimention column is created for salary and experienced years.
12.Importing train_test_split for training the x and y.
13.We have to check shape the of x and y for to fit data into linear regression.
14.We have to import and create instance for linear regression.
15. By using fit() we had fit the x and values for model evalution.
16.lr.coef_ find the coefficient of salary and experienced years.
17.lr.intercept find the interception of y.
18. By using predict() predicyions for model is done to calcualte actual values.
19. By using plot method linear line had drawn.
```

Ok. By using the metrics we had calculated the absolute error and square error.
Ok.At last the Actual value and predicted value is compared.