Title: Predicting Employee Salary Based on Experience

Problem Statement:

Background:

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.

Dataset Description:

The dataset consists of the following columns: 1.Experience_Years: Number of years of experience the employee has. 2.Salary: Salary of the employee (in dollars).

Importing Libraries

In [53]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Import DataSet

In [2]:
```
emp_sal=pd.read_csv("salary_exp.csv")
emp_sal
```

Out[2]:

	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 Understanding

In [3]:

```
emp_sal.head()
```

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

Initial Check Up

In [4]:
```
emp_sal.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
```

In [6]:
```
emp_sal.describe()
```

Out[6]:

	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

In [7]:
```
emp_sal.shape
```

Out[7]: (40, 2)

Asking Questions to the Data

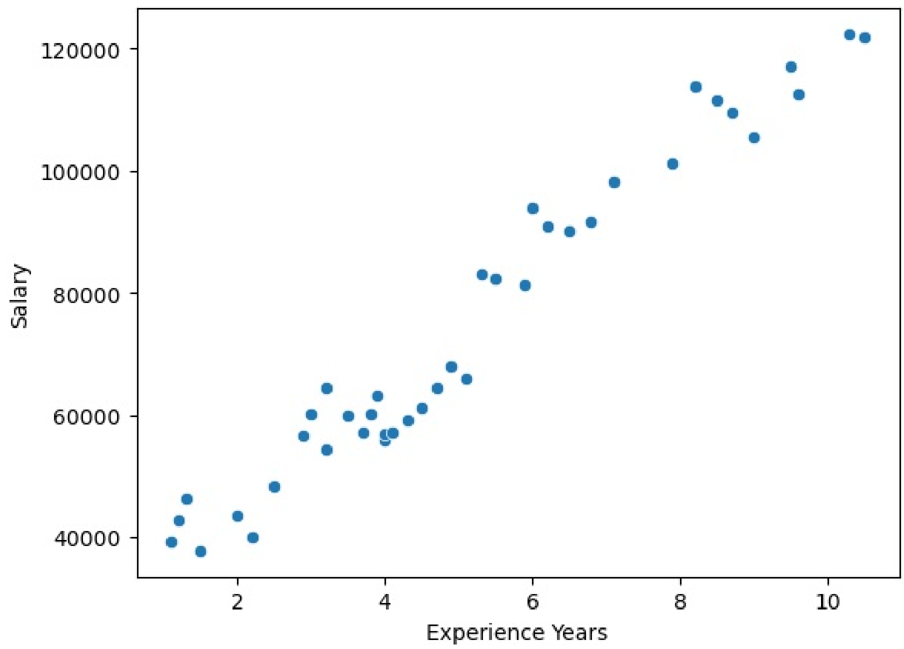
1.What is the highest Salary of employee and how many years of experience employee has. 2.Years of experience greater than 7. 3.Which Experience year has Salary is equals to 46205. 4.Maximum salary of an employee 5.Average salary of an employee 6.Maxmimum years of experience 7.Average year experience of an employee 8.How many null values are in the emp_sal

Data Visualization

In [8]:
```
#1.What is the highest Salary of employee and how many years of experience employee has.
#4.Maximun salary of an employee
#7.Average years of experience

sns.scatterplot(emp_sal,x="Experience Years",y="Salary")
```

Out[8]: <Axes: xlabel='Experience Years', ylabel='Salary'>



In [49]:
```python
#2.Years of experience greater than 7.
emp_sal[emp_sal['Experience Years']>7]
```

Out[49]:

	Experience Years	Salary
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

In [50]:
```python
#3.Which Experience year has Salary is equals to 46205.
emp_sal[emp_sal['Salary']==46205]
```

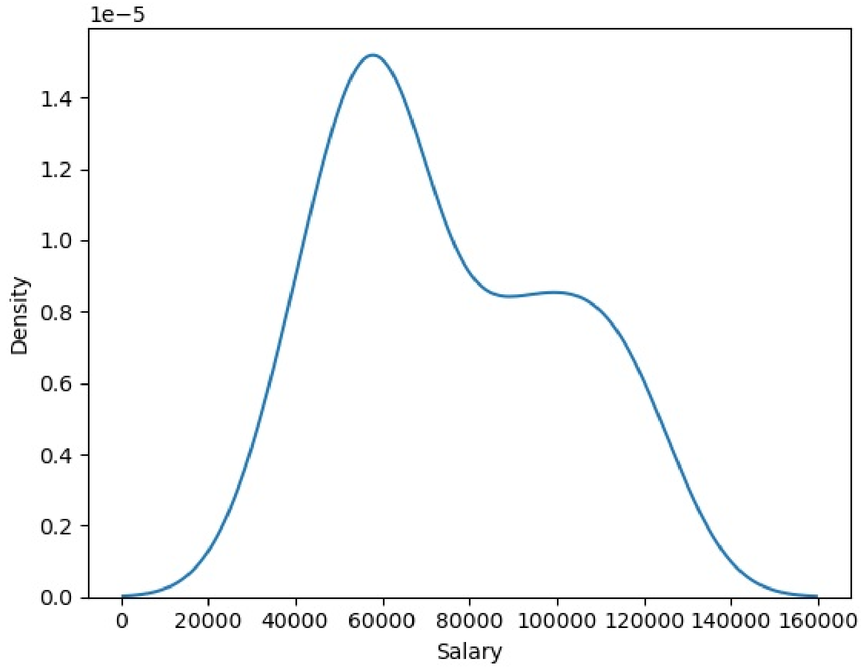
Out[50]:

	Experience Years	Salary
2	1.3	46205

In [11]:
```python
#5. Average salary of an employee
sns.kdeplot(emp_sal,x="Salary")
```

```
C:\Users\sindhu\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is dep
recated 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):
```

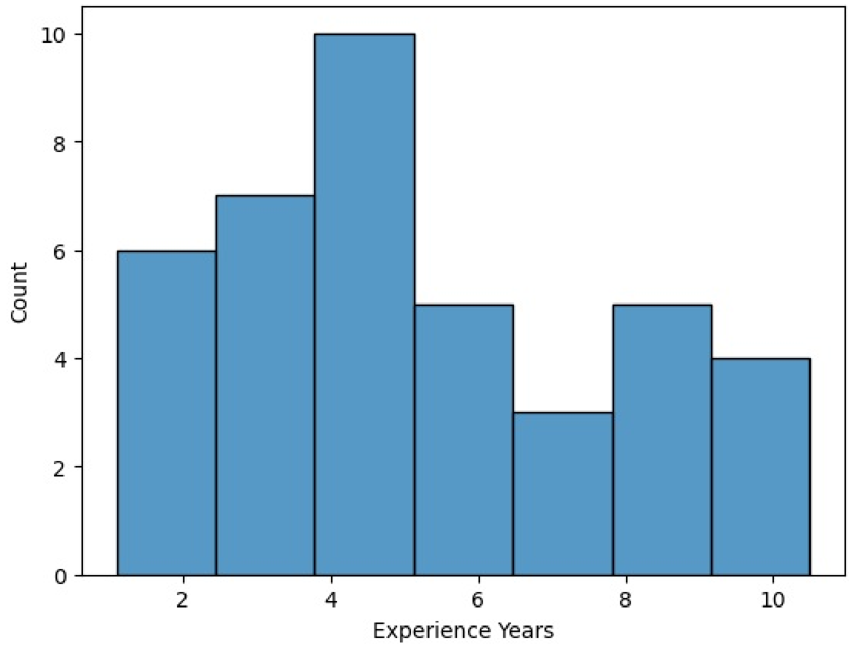
Out[11]: <Axes: xlabel='Salary', ylabel='Density'>



In [12]:
```
#7.Average experience of an employee
sns.histplot(emp_sal,x='Experience Years')
```

```
C:\Users\sindhu\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is dep
recated 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[12]: `<Axes: xlabel='Experience Years', ylabel='Count'>`



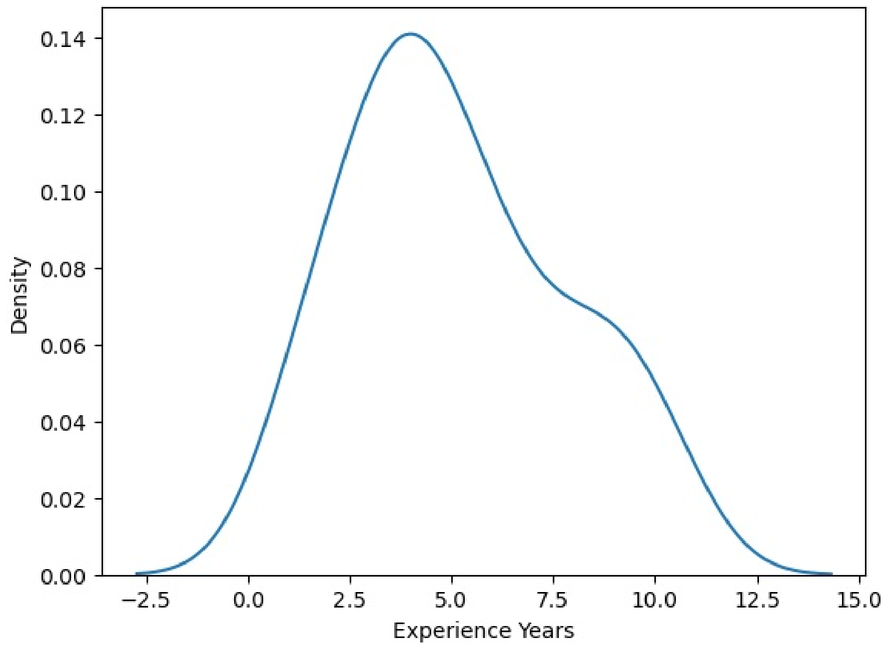
In [13]:
```
#8.How many null values are in the emp_sal
emp_sal.isnull().count()
```

Out[13]:
```
Experience Years    40
Salary              40
dtype: int64
```

In [14]:
```
sns.kdeplot(data=emp_sal,x='Experience Years')
```

```
C:\Users\sindhu\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is dep
recated 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):
```

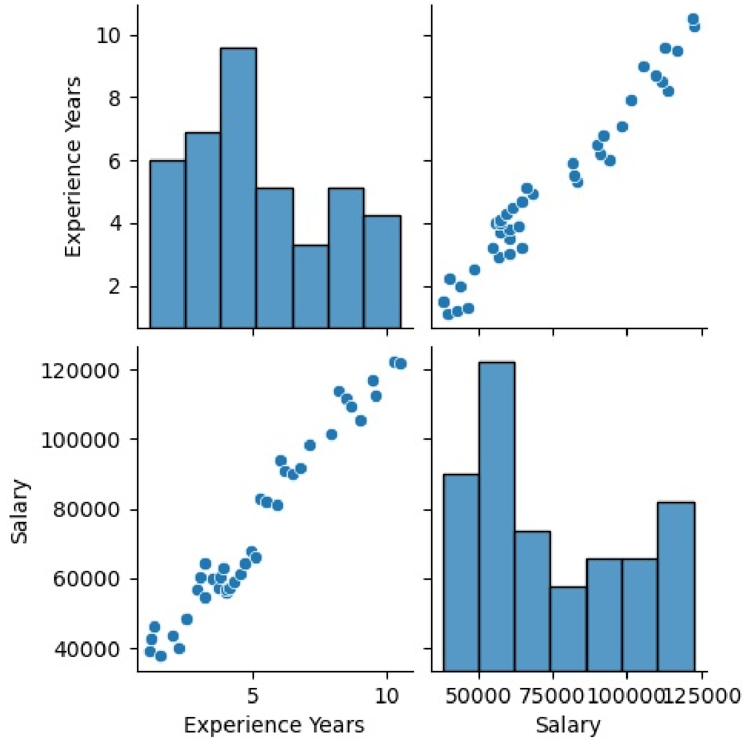
Out[14]: `<Axes: xlabel='Experience Years', ylabel='Density'>`



In [15]:
```
sns.pairplot(emp_sal)
```

```
C:\Users\sindhu\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is dep
recated 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):
C:\Users\sindhu\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is dep
recated 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[15]: <seaborn.axisgrid.PairGrid at 0x2f5f8476b10>



ML

Linear Regression

Linear regression is a supervised machine learning method that provides a linear relationship between an independent variable and a dependent variable to predict the outcome of future events.

In [17]:
```
# correlation
```

```
emp_sal.corr()
```

Out[17]:

	Experience Years	Salary
Experience Years	1.000000	0.977692
Salary	0.977692	1.000000

In [18]:
```
emp_sal.max()
```

Out[18]:
```
Experience Years        10.5
Salary              122391.0
dtype: float64
```

In [19]:
```
emp_sal.min()
```

Out[19]:
```
Experience Years       1.1
Salary             37731.0
dtype: float64
```

In [20]:
```
X=emp_sal.iloc[:,0:1]
y=emp_sal.iloc[:,-1]
X.head()
```

Out[20]:

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

In [21]:
```
y.head()
```

Out[21]:
```
0    39343
1    42774
2    46205
3    37731
4    43525
Name: Salary, dtype: int64
```

Train and Test

In [ ]:
```
```

In [22]:
```
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 [23]:
```
X_train.head()
```

Out[23]:

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

In [24]:
```
X_test
```

Out[24]:

	Experience Years
27	6.2
9	3.2
14	3.9
0	1.1
2	1.3
30	7.1
13	3.8
36	9.5

In [25]:
```
y_test
```

Out[25]:
```
27     91000
9      54445
14     63218
0      39343
2      46205
30     98273
13     60200
36    116969
Name: Salary, dtype: int64
```

In [26]:
```
X_train.shape
```

Out[26]:
```
(32, 1)
```

In [27]:
```
X_test.shape
```

Out[27]:
```
(8, 1)
```

In [28]:
```
y_train.shape
```

Out[28]:
```
(32,)
```

In [29]:
```
y_test.shape
```

Out[29]:
```
(8,)
```

Model Building

In [30]:
```
from sklearn.linear_model import LinearRegression
Lr=LinearRegression()
Lr
```

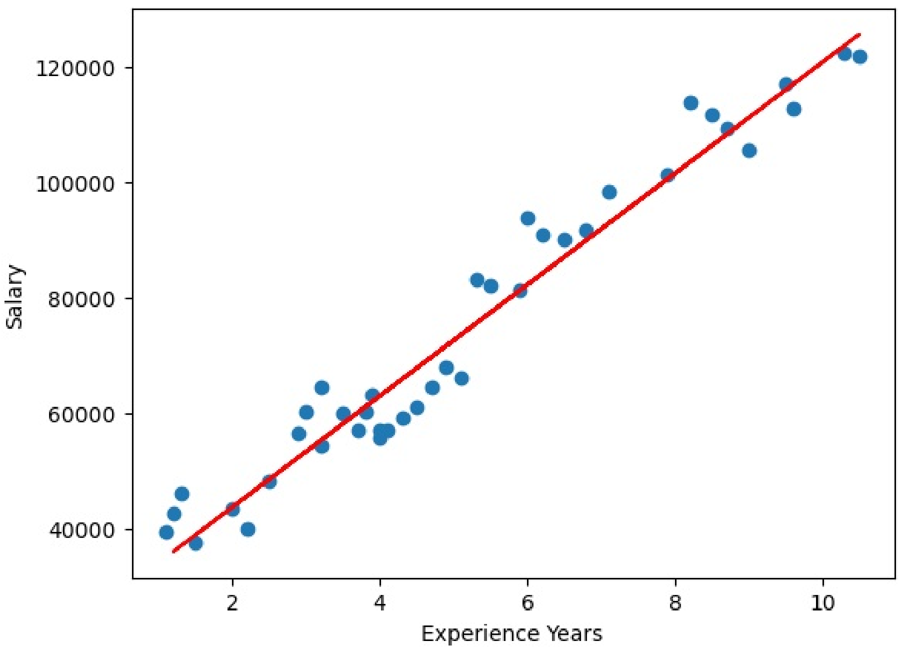
Out[30]:
```
▾ LinearRegression
LinearRegression()
```

In [31]:
```
Lr.fit(X_train,y_train)
```

Out[31]:
```
▾ LinearRegression
LinearRegression()
```

In [32]:
```
#Data visualization
import matplotlib.pyplot as plt
plt.scatter(emp_sal['Experience Years'],emp_sal['Salary'])
plt.plot(X_train,Lr.predict(X_train),color='red')
plt.xlabel("Experience Years")
plt.ylabel("Salary")
```

Out[32]:
```
Text(0, 0.5, 'Salary')
```



#calculating coefficient, Intercept and predicted value

In [33]:
```
Lr.coef_
```
Out[33]:
```
array([9629.89561636])
```
In [34]:
```
Lr.intercept_
```
Out[34]:
```
24469.054538114055
```
In [35]:
```
#predict
Lr.predict(X_test)
```
Out[35]:
```
array([ 84174.40735952,  55284.72051045,  62025.6474419 ,  35061.9397161 ,
        36987.91883938,  92841.31341423,  61062.65788026, 115953.06289349])
```
In [54]:
```
Lr.predict([[3.2]])
```
```
C:\Users\sindhu\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature name
s, but LinearRegression was fitted with feature names
  warnings.warn(
```
Out[54]:
```
array([55284.72051045])
```
The predicted salary of 3.2 is 55284 and the actual value of 3.2 is 64445. The predicted value is decreased by 9161 than actual value.

Model Evaluation

In [37]:
```
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
y_pred=Lr.predict(X_test)
y_pred
```
Out[37]:
```
array([ 84174.40735952,  55284.72051045,  62025.6474419 ,  35061.9397161 ,
        36987.91883938,  92841.31341423,  61062.65788026, 115953.06289349])
```
In [38]:
```
mean_absolute_error(y_test,y_pred)
```
Out[38]:
```
3708.261090762284
```
In [39]:
```
r2_score(y_test,y_pred)
```
Out[39]:
```
0.9655807830897453
```
In [40]:
```
MSE=mean_squared_error(y_test,y_pred)
MSE
```
Out[40]:
```
22909642.289620496
```
In [41]:
```
import pandas as pd
from sklearn.metrics import mean_squared_error
RMSE=MSE**0.5
data_rmse={'Actual (y_test)':y_test,'predicted(y_pred)':y_pred}
df_rmse=pd.DataFrame(data_rmse)
df_rmse.head()
```
Out[41]:

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

In [43]:
```
df_rmse['Actual (y_test)'].sum()
```
Out[43]:
```
569653
```
In [42]:
```
# Total Prediction
df_rmse['predicted(y_pred)'].sum()
```
Out[42]:
```
543391.668055328
```

Insights

In [2]:
```
#1.In salary_exp dataset consists of two columns "Experience Years" and "salary".
#2.Salary - Represents the salary of an employee.
#3.Experience Years - Represents the years of experience.
#4.'salary' dtype - int, 'Experience Years' dtype - float and Both columns has 40 null values.
#5.Average years of experience is 5.1 years of an employee.
```

```
#6.Average amount of salary that employee has $74000.
#7.Highest salary of an employee is $122000.
#8.Maximum years of experience an employee has 10.5
#9.Minimum salary is $37000 and minimun Experience Years        is 1.1 years.
#10.Shape of emp_sal dataset is(40,2) means 40 rowa and 2 columns.
#11.Shape of X_test,y_test is(8,1) and X_train,y_train is(32,1)
#12.iloc is used for integer indexing values 'X' has Experience Years & 'y' has salary column.
#13.corr() tells the relationship between Salary and Experirnce Years.
#14.fit() can be used to fits data into LinearRegression.
#15.LinearRegression helps to determine the relationship between Experience Years and salary.
#16.LinearRegression allows for prediction of salary based on given years of experience by using predict().
#17.Scatter plot to visualize the relationship between Experience Years and salary.
#18.kde plot for both Experience Years and salary to understand data visualization.
#19.Calculate Model Evalution metrics such as MAE,MSE and R2-squared helps in assess model's performance.
#20.The predicted value of 3.2 is decreased by 9161 than actual value of 3.2.
```