importing or creatind data

In [51]:
```
import numpy as np
import pandas as pd
data={'Experience': [1,2,3,4,5,6,7,8,9,10],
     'salary' : [45000, 50000, 60000, 65000, 70000, 80000, 85000, 90000, 95000,
df= pd.DataFrame(data)
#print(df)
```

Extracting the Independent (x) and Dependent (y) varibels

In [52]:
```
#x= df.iloc[:,0]    # independent variables
#= df.iloc[:,1]    # Dependent Variables

x=df[['Experience']]
y=df['salary']
```

Train and Test the data

In [53]:
```
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=0
```

instalizing the Train model using Linear Regression

```
DATA PRE-PROCESSING
```

In [54]:
```
from sklearn.linear_model import LinearRegression
# Initializing and training the model
Lr = LinearRegression()
Lr.fit(X_train, y_train)
```

Out[54]:
```
LinearRegression()
```

Making the Predicition

In [55]:
```
y_pred= Lr.predict(X_test)
```

Calculating Mean_squared_Error and R2 squared error

In [56]:

```
from sklearn.metrics import mean_squared_error,r2_score
mse = mean_squared_error(y_pred,y_test)
r2 = r2_score(y_test,y_pred)

#  printing the MSE   and    R2
print (f"mean_squred_error : {mse}")
print  (f"R-Squared_score   :{r2}")
```

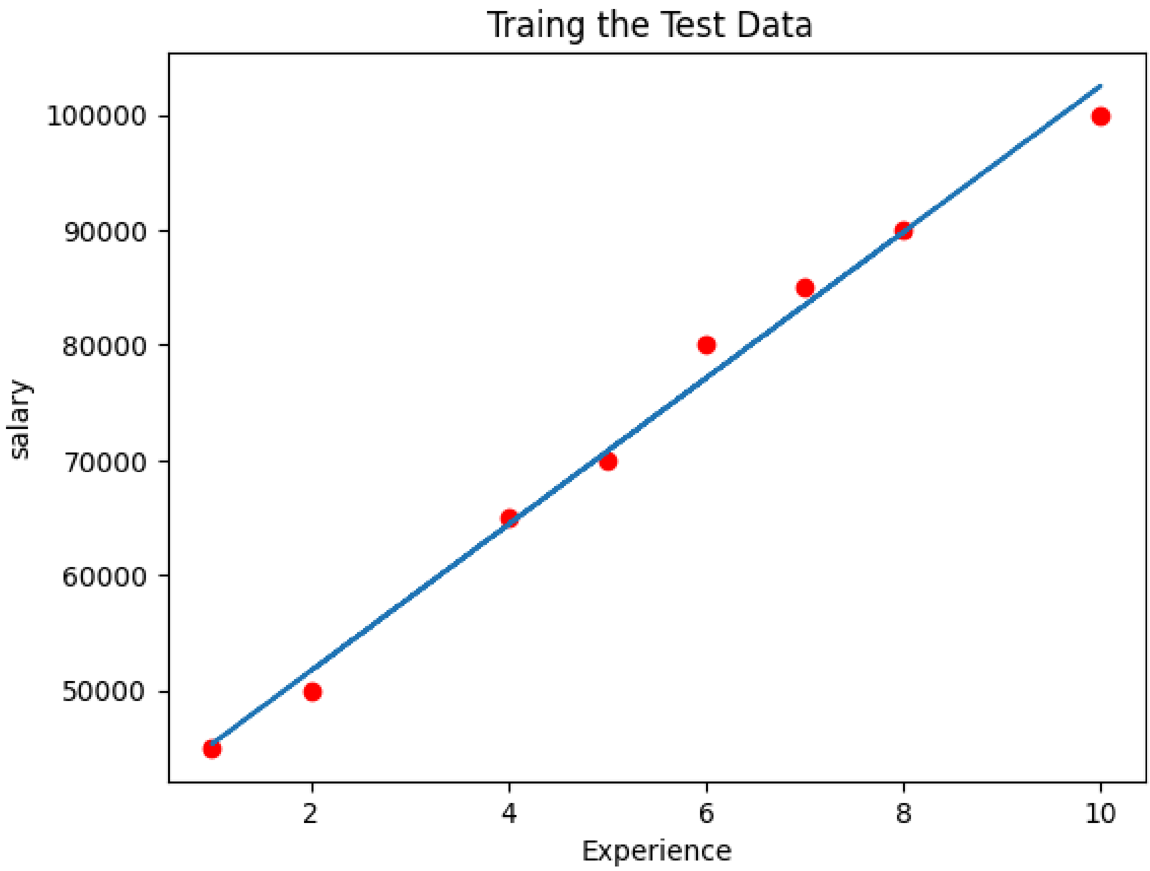
```
mean_squred_error : 2570120.748618475
R-Squared_score   :0.991607768984103
```

In [57]:

```
##    Plotting the Training Data
```

In [58]:

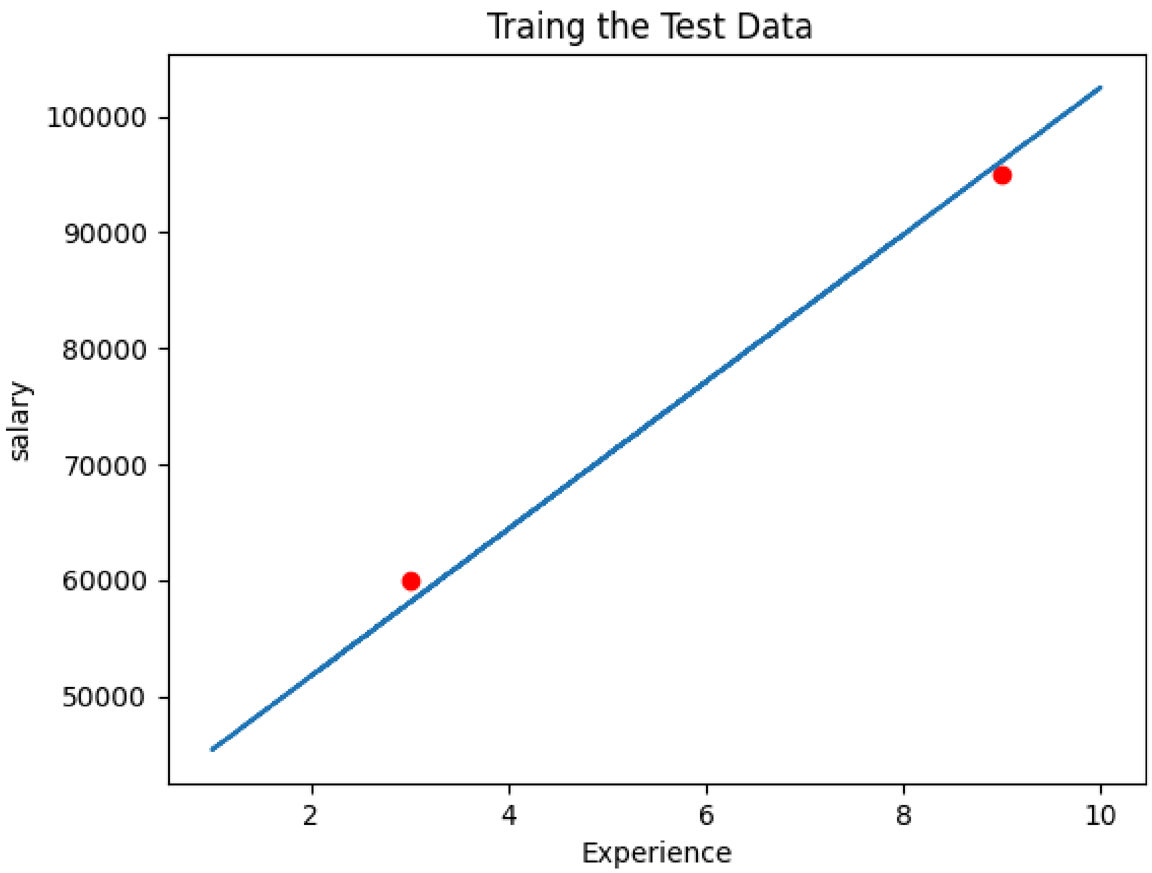
```
import matplotlib .pyplot as pp
pp.scatter(X_train,y_train,color="Red")
pp.plot(X_train,Lr.predict(X_train))
pp.title("Traing the Test Data ")
pp.xlabel('Experience')
pp.ylabel('salary')
pp.show()
```



In [59]:

```
print(df)
pp.scatter(X_test,y_test,color="Red")
pp.plot(X_train,Lr.predict(X_train))
pp.title("Traing the Test Data ")
pp.xlabel('Experience')
pp.ylabel('salary')
pp.show()
```

```
   Experience  salary
0           1   45000
1           2   50000
2           3   60000
3           4   65000
4           5   70000
5           6   80000
6           7   85000
7           8   90000
8           9   95000
9          10  100000
```



```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

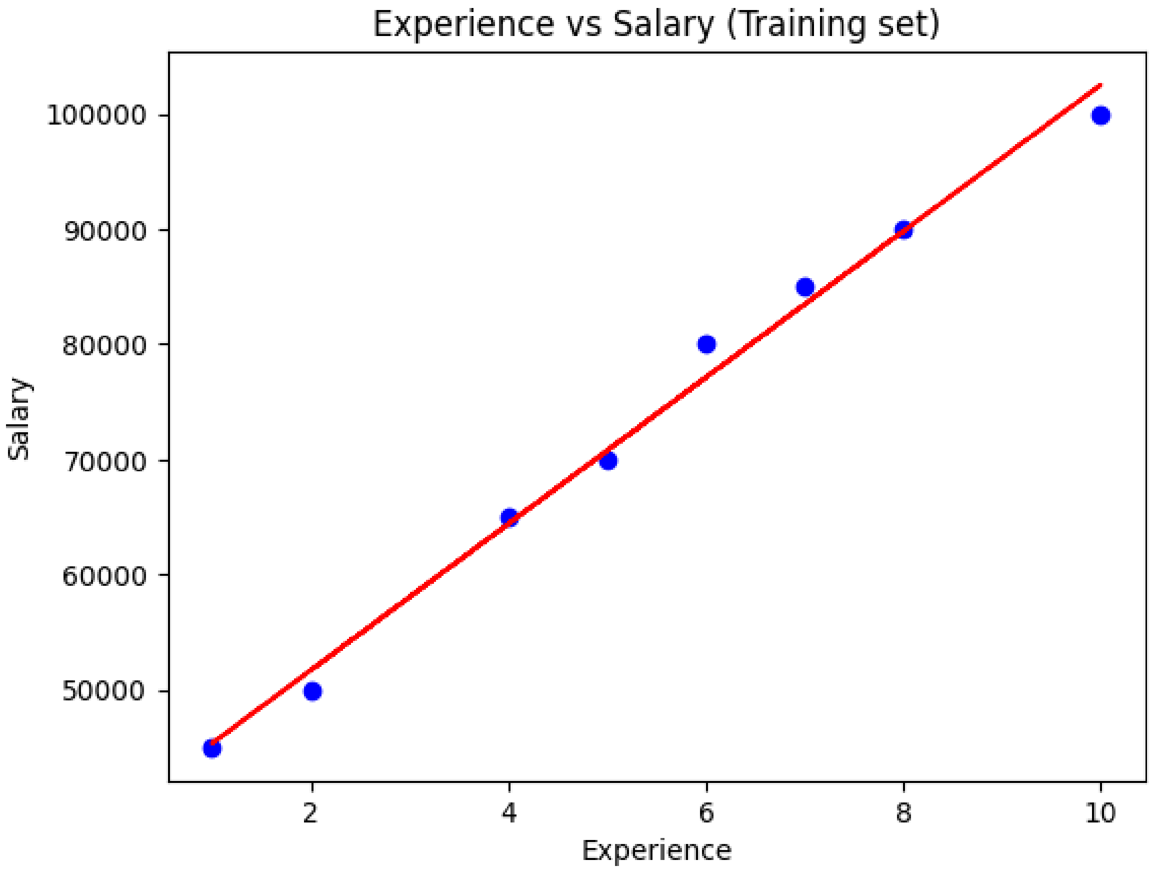
```
In [ ]:
```

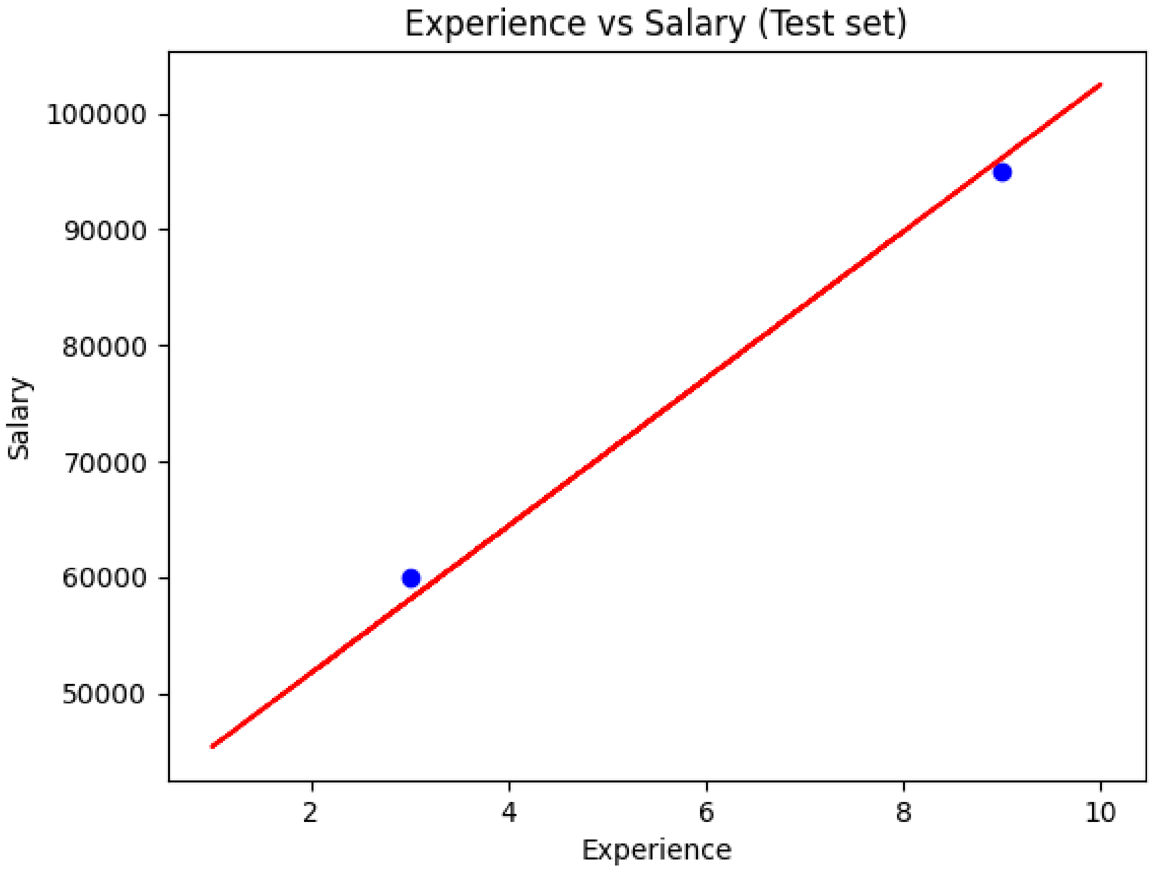
From AI in Linear Regression

```
In [17]: import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

# Creating a simple dataset
```

```
data = {
    'Experience': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'Salary': [45000, 50000, 60000, 65000, 70000, 80000, 85000, 90000, 95000, 10
}
df = pd.DataFrame(data)

# Extracting independent and dependent variables
X = df[['Experience']]  # Independent variable
y = df['Salary']        # Dependent variable

# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_

# Initializing and training the model
model = LinearRegression()
model.fit(X_train, y_train)

# Making predictions
y_pred = model.predict(X_test)

# Calculating mean squared error and R-squared score
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse}')
print(f'R-squared Score: {r2}')

# Plotting the training data
plt.scatter(X_train, y_train, color='blue')
plt.plot(X_train, model.predict(X_train), color='red')
plt.title('Experience vs Salary (Training set)')
plt.xlabel('Experience')
plt.ylabel('Salary')
plt.show()

# Plotting the test data and predictions
plt.scatter(X_test, y_test, color='blue')
plt.plot(X_train, model.predict(X_train), color='red')
plt.title('Experience vs Salary (Test set)')
plt.xlabel('Experience')
plt.ylabel('Salary')
plt.show()
```

```
Mean Squared Error: 2570120.748618475
R-squared Score: 0.991607768984103
```


Experience vs Salary (Training set)
Salary
100000
90000
80000
70000
60000
50000
2
4
6
8
10
Experience


Experience vs Salary (Test set)
Salary
100000
90000
80000
70000
60000
50000
2
4
6
8
10
Experience