Data Preprocessing:

Started with cleaning and preparing the dataset to ensure accuracy and reliability. Dealt with missing values, outliers, and performed necessary transformations to get the data ready for analysis.

Splitting into Training and Test

step of splitting the dataset into training and test sets using train_test_split from sklearn.model_selection. This ensures the model's ability to generalize well to unseen data.

Visualizing Training and Test Plots:

matplotlib to create insightful visualizations. Plotted the training data to understand the model's learning process and then visualized the test data to evaluate its predictive performance.

Utilized sklearn.linear_model:

Implemented Simple Linear Regression using the powerful LinearRegression module from sklearn.linear_model. This allowed for efficient modeling and prediction.

Key Takeaways:

Insights from Training Data: Explored how the model learned from the training data and identified patterns. Evaluation on Test Data: Assessed the model's predictive capabilities on unseen data to ensure its reliability in real-world scenarios.

In [1]:

```
# Importing Libraries

import numpy  as np
import pandas as pd
import  matplotlib.pyplot as plt
```

In [ ]:

```

```

In [5]:

```
#Importing the dataset
dataset=pd.read_csv("salary_Data.csv")
x=dataset.iloc[: ,:-1].values
y=dataset.iloc[: ,1].values
```

In [6]:

```
print(x)
```

```
[[ 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]]
```

In [7]:

```
print(y)
```

```
[ 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.]
```

In [8]:

```
#we dont have to go through data preprocessing step
```

In [11]:

```
#Spliting the dataset  into the Training set and test set

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_st
```

In [12]:
```
print(x_train)
```

```
[[ 9.6]
 [ 4. ]
 [ 5.3]
 [ 7.9]
 [ 2.9]
 [ 5.1]
 [ 3.2]
 [ 4.5]
 [ 8.2]
 [ 6.8]
 [ 1.3]
 [10.5]
 [ 3. ]
 [ 2.2]
 [ 5.9]
 [ 6. ]
 [ 3.7]
 [ 3.2]
 [ 9. ]
 [ 2. ]
 [ 1.1]
 [ 7.1]
 [ 4.9]
 [ 4. ]]
```

In [13]:
```
print(x_test)
```

```
[[ 1.5]
 [10.3]
 [ 4.1]
 [ 3.9]
 [ 9.5]
 [ 8.7]]
```

In [14]:
```
print(y_train)
```

```
[112635.  55794.  83088. 101302.  56642.  66029.  64445.  61111. 113812.
  91738.  46205. 121872.  60150.  39891.  81363.  93940.  57189.  54445.
 105582.  43525.  39343.  98273.  67938.  56957.]
```

In [15]:
```
print(y_test)
```

```
[ 37731. 122391.  57081.  63218. 116969. 109431.]
```

In [17]:
```
#Training the simple linear Regression  model on the Training set

from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
regressor.fit(x_train,y_train)
```

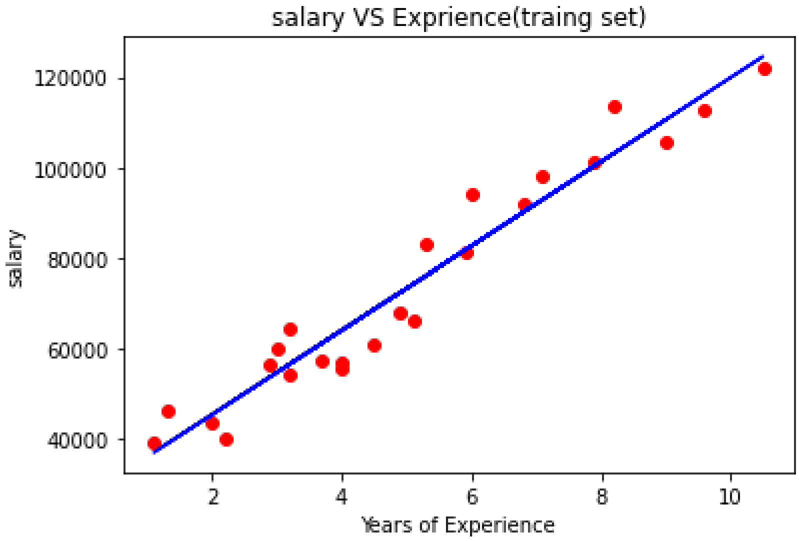
Out[17]:
```
▾ LinearRegression
LinearRegression()
```

Predecting the Test set result

```
In [18]: y_predicted=regressor.predict(x_test)
```

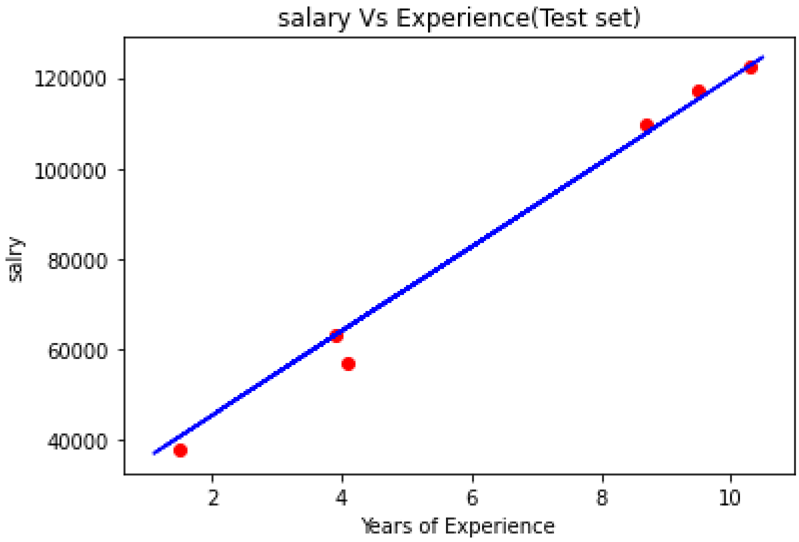
```
In [19]: #visualising training set results

         plt.scatter(x_train,y_train,color='red')
         plt.plot(x_train,regressor.predict(x_train),color='blue')
         plt.title('salary VS Exprience(traing set)')
         plt.xlabel('Years of Experience')
         plt.ylabel('salary')
         plt.show()
```



```
In [20]: #Visualizing the Test set results

         plt.scatter(x_test,y_test,color='red')
         plt.plot(x_train,regressor.predict(x_train),color='blue')
         plt.title('salary Vs Experience(Test set)')
         plt.xlabel('Years of Experience')
         plt.ylabel('salry')
         plt.show()
```



In [ ]: