## Experiment-2

Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

### In [1]:

```
import pandas as pd
import numpy as np
```

### In [2]:

```
df_salary=pd.read_csv('Salary_Data.csv')
df_salary.head()
```

## Out[2]:

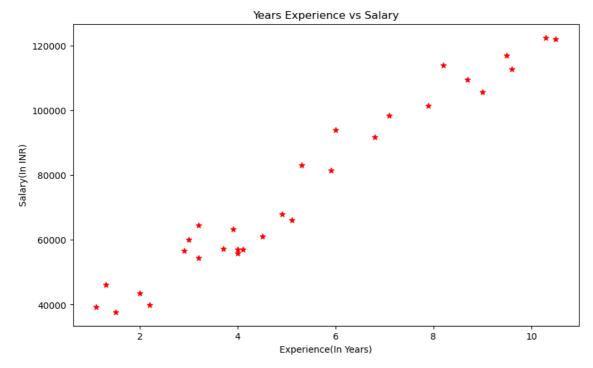
	YearsExperience	Salary		
0	1.1	39343.0		
1	1.3	46205.0		
2	1.5	37731.0		
3	2.0	43525.0		
4	2.2	39891.0		

## In [3]:

import matplotlib.pyplot as plt

## In [4]:

```
plt.figure(figsize=(10,6))
plt.scatter(x='YearsExperience',y='Salary',data=df_salary,marker='*',color='red')
plt.title('Years Experience vs Salary')
plt.xlabel('Experience(In Years)')
plt.ylabel('Salary(In INR)')
plt.show()
```



# In [5]:

```
X=df_salary.drop('Salary',axis=1)
X.head(5)
```

## Out[5]:

Years	Experience
0	1.1
1	1.3
2	1.5
3	2.0
4	22

```
In [6]:
y=df_salary[['Salary']]
y.head(5)
Out[6]:
     Salary
 0 39343.0
 1 46205.0
 2 37731.0
 3 43525.0
 4 39891.0
In [7]:
from sklearn.model_selection import train_test_split
In [9]:
ain, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=101)
In [19]:
from sklearn.preprocessing import StandardScaler
In [21]:
sc_X=StandardScaler()
sc_y=StandardScaler()
In [25]:
X_train=sc_X.fit_transform(X_train)
In [26]:
y_train=sc_y.fit_transform(y_train)
In [27]:
from sklearn.linear_model import LinearRegression
In [28]:
regressor=LinearRegression()
```

#### In [29]:

```
regressor.fit(X_train,y_train)
```

## Out[29]:

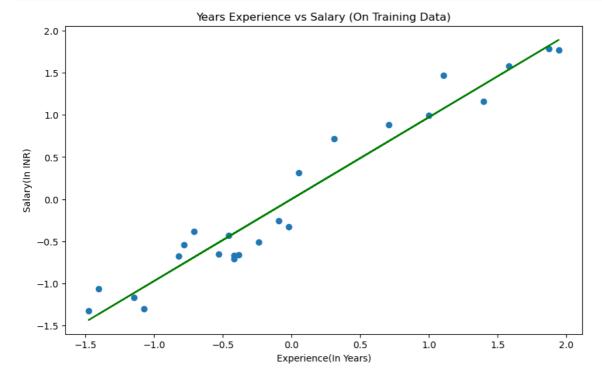
```
LinearRegression
LinearRegression()
```

#### In [30]:

```
y_pred_train=regressor.predict(X_train)
```

#### In [31]:

```
plt.figure(figsize=(10,6))
plt.scatter(X_train,y_train)
plt.plot(X_train,y_pred_train,color='green')
plt.title('Years Experience vs Salary (On Training Data)')
plt.xlabel('Experience(In Years)')
plt.ylabel('Salary(In INR)')
plt.show()
```



## In [32]:

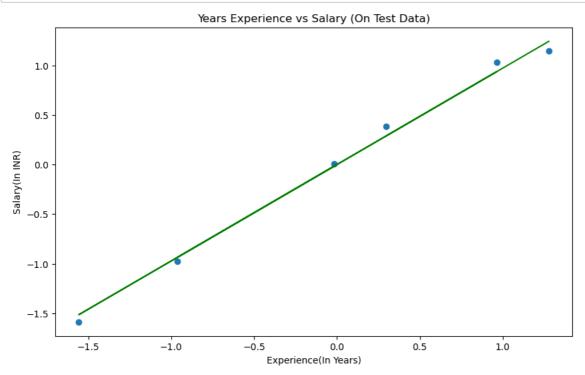
```
X_test=sc_X.fit_transform(X_test)
y_test=sc_y.fit_transform(y_test)
```

#### In [33]:

```
y_pred_test=regressor.predict(X_test)
```

# In [34]:

```
plt.figure(figsize=(10,6))
plt.scatter(X_test,y_test)
plt.plot(X_test,y_pred_test,color='green')
plt.title('Years Experience vs Salary (On Test Data)')
plt.xlabel('Experience(In Years)')
plt.ylabel('Salary(In INR)')
plt.show()
```



# In [35]:

from sklearn.metrics import r2\_score

### In [36]:

```
score=r2_score(y_test,y_pred_test)
```

## In [37]:

score

### Out[37]:

0.994287174293827

Ι	n [ ]:				