1.Importing Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.axes as ax
```

2.Loading Data

In [2]:
```
data = pd.read_csv(r'C:\Users\G.SAI KRISHNA\Desktop\ML_Projects\ML_GFG\train.csv')
```

In [3]:
```
data.head()
```

Out[3]:

	x	y
0	24.0	21.549452
1	50.0	47.464463
2	15.0	17.218656
3	38.0	36.586398
4	87.0	87.288984

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 700 entries, 0 to 699
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   x       700 non-null    float64
 1   y       699 non-null    float64
dtypes: float64(2)
memory usage: 11.1 KB
```

3.Data Preprocessing

In [5]:
```
#Handling Null Values
data = data.dropna(axis=0)
```

In [6]:
```
data['x'].max()
```

Out[6]: 100.0

In [7]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 699 entries, 0 to 699
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   x       699 non-null    float64
 1   y       699 non-null    float64
dtypes: float64(2)
memory usage: 16.4 KB
```

4.Data Splitting

In [8]:
```
train_input = np.array(data.iloc[0:500,0]).reshape(500,1)
train_output = np.array(data.iloc[0:500,1]).reshape(500,1)

test_input = np.array(data.iloc[500:700,0]).reshape(199,1)
test_output = np.array(data.iloc[500:700,1]).reshape(199,1)
```

In [9]:
```
train_input.shape
```

Out[9]: (500, 1)

In [10]:
```
train_output.shape
```

Out[10]: (500, 1)

In [11]:
```
test_input.shape
```

Out[11]: (199, 1)

In [12]:
```
test_output.shape
```

Out[12]: (199, 1)

5.Linear Regression

Train the Data

In [13]:
```
from sklearn.linear_model import LinearRegression

linear_regressor = LinearRegression()
linear_regressor.fit(train_input,train_output)
```

Out[13]: LinearRegression()

Predicting Test output

In [14]:
```
predicted_value = linear_regressor.predict(test_input)
```

In [15]:
```
predicted_value
```

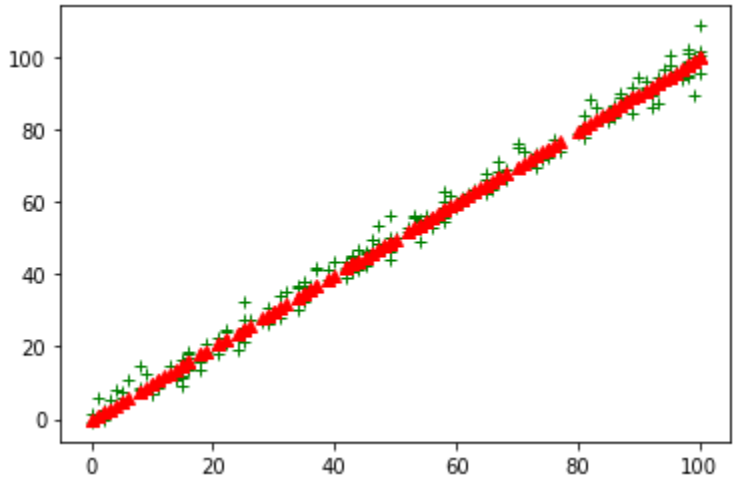
Out[15]:
```
array([[ 4.76896187],
       [60.82492252],
       [46.81093236],
       [97.86189652],
       [29.7939443 ],
       [62.82692111],
       [-0.23603461],
       [99.86389511],
       [17.78195274],
       [29.7939443 ],
       [97.86189652],
       [15.77995414],
       [21.78594992],
       [54.81892673],
       [42.80693517],
       [74.83891268],
       [90.85490143],
       [45.80993306],
       [84.84890565],
```

In [16]:
```
test_output
```

```
       [ 14.81106804],
       [ 48.09368034],
       [ 42.29145672],
       [ 52.73389794],
       [ 36.72396986],
       [ 28.64535198],
       [ 62.16675273],
       [ 95.58459518],
       [ 66.04325304],
       [ 99.9566225 ],
       [ 46.14941984],
       [ 89.13754963],
       [ 69.71787806],
       [ 12.31736648],
       [ 78.20296268],
       [ 71.30995371],
       [ 81.45544709],
       [ 58.59500642],
       [ 94.62509374],
       [ 88.60376995],
```

Visualizing Model Performance

In [17]:
```
plt.figure()
plt.plot(test_input,test_output,'+',color="green")
plt.plot(test_input,predicted_value,'^',color="red")
plt.show()
```



In [18]:
```
from sklearn.metrics import mean_squared_error

error=mean_squared_error(test_output,predicted_value)
error
```

Out[18]: 8.03003159183537

In [19]:
```
print("Accuracy : "+str(100 - error)+"%")
```

```
Accuracy : 91.96996840816463%
```

/