In [1]:
```
##importing csv file
import pandas as pd
df1=pd.read_csv('C:/Users/USER/Downloads/samplefile.csv')
print(df1.head)
```

```
<bound method NDFrame.head of         c1      c2     c3    c4     c5     c6     c7      c8  c9  c10
0   -2.415  23.335  6.792 -7.23  15.27  -2.76  77.50  102.76  15   0
1    2.947  28.697  6.621 -7.23  15.27  -2.76  77.50  102.76  16   0
2   -4.936  20.814  6.616  0.81  23.31  11.49  91.75  102.76  16   0
3   -3.865  21.885  6.608 -2.41  20.09  -1.78  78.48  102.76  16   0
4    2.681  28.431  6.404 -7.23  15.27  -2.76  77.50  102.76  16   0
..     ...     ...    ...   ...    ...    ...    ...     ...  ..  ...
394 -4.673  21.077  2.791 -7.23  15.27  -2.76  77.50  102.76  36   1
395 -2.398  23.352  2.790 -0.80  21.70  -3.24  77.02  102.76  36   1
396  3.733  29.483  2.790 -7.23  15.27  -2.76  77.50  102.76  36   1
397 -0.986  24.764  2.785  0.81  23.31  11.49  91.75  102.76  36   1
398 -6.194  19.556  2.782 -0.80  21.70  -3.24  77.02  102.76  36   1

[399 rows x 10 columns]>
```

In [2]:
```
df1.head
```

Out[2]:
```
<bound method NDFrame.head of         c1      c2     c3    c4     c5     c6     c7      c8  c9  c10
0   -2.415  23.335  6.792 -7.23  15.27  -2.76  77.50  102.76  15   0
1    2.947  28.697  6.621 -7.23  15.27  -2.76  77.50  102.76  16   0
2   -4.936  20.814  6.616  0.81  23.31  11.49  91.75  102.76  16   0
3   -3.865  21.885  6.608 -2.41  20.09  -1.78  78.48  102.76  16   0
4    2.681  28.431  6.404 -7.23  15.27  -2.76  77.50  102.76  16   0
..     ...     ...    ...   ...    ...    ...    ...     ...  ..  ...
394 -4.673  21.077  2.791 -7.23  15.27  -2.76  77.50  102.76  36   1
395 -2.398  23.352  2.790 -0.80  21.70  -3.24  77.02  102.76  36   1
396  3.733  29.483  2.790 -7.23  15.27  -2.76  77.50  102.76  36   1
397 -0.986  24.764  2.785  0.81  23.31  11.49  91.75  102.76  36   1
398 -6.194  19.556  2.782 -0.80  21.70  -3.24  77.02  102.76  36   1

[399 rows x 10 columns]>
```

In [3]:
```
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.linear_model import LinearRegression
```

In [4]:
```
df1.dropna
```

Out[4]:
```
<bound method DataFrame.dropna of         c1      c2     c3    c4     c5     c6     c7      c8  c9  c10
0   -2.415  23.335  6.792 -7.23  15.27  -2.76  77.50  102.76  15   0
1    2.947  28.697  6.621 -7.23  15.27  -2.76  77.50  102.76  16   0
2   -4.936  20.814  6.616  0.81  23.31  11.49  91.75  102.76  16   0
3   -3.865  21.885  6.608 -2.41  20.09  -1.78  78.48  102.76  16   0
4    2.681  28.431  6.404 -7.23  15.27  -2.76  77.50  102.76  16   0
..     ...     ...    ...   ...    ...    ...    ...     ...  ..  ...
394 -4.673  21.077  2.791 -7.23  15.27  -2.76  77.50  102.76  36   1
395 -2.398  23.352  2.790 -0.80  21.70  -3.24  77.02  102.76  36   1
396  3.733  29.483  2.790 -7.23  15.27  -2.76  77.50  102.76  36   1
397 -0.986  24.764  2.785  0.81  23.31  11.49  91.75  102.76  36   1
398 -6.194  19.556  2.782 -0.80  21.70  -3.24  77.02  102.76  36   1

[399 rows x 10 columns]>
```

In [5]:
```
#printing value of x
X=df1[['c1','c2','c3','c4','c5','c6','c7','c8','c9']]
print(X)
```

```
       c1      c2     c3    c4     c5     c6     c7      c8  c9
0   -2.415  23.335  6.792 -7.23  15.27  -2.76  77.50  102.76  15
1    2.947  28.697  6.621 -7.23  15.27  -2.76  77.50  102.76  16
2   -4.936  20.814  6.616  0.81  23.31  11.49  91.75  102.76  16
3   -3.865  21.885  6.608 -2.41  20.09  -1.78  78.48  102.76  16
4    2.681  28.431  6.404 -7.23  15.27  -2.76  77.50  102.76  16
..     ...     ...    ...   ...    ...    ...    ...     ...  ..
394 -4.673  21.077  2.791 -7.23  15.27  -2.76  77.50  102.76  36
395 -2.398  23.352  2.790 -0.80  21.70  -3.24  77.02  102.76  36
396  3.733  29.483  2.790 -7.23  15.27  -2.76  77.50  102.76  36
397 -0.986  24.764  2.785  0.81  23.31  11.49  91.75  102.76  36
398 -6.194  19.556  2.782 -0.80  21.70  -3.24  77.02  102.76  36

[399 rows x 9 columns]
```

In [6]:
```
#printing value of y
Y=df1['c10']
print(Y)
```

```
0      0
1      0
2      0
3      0
4      0
      ..
394    1
395    1
396    1
397    1
398    1
Name: c10, Length: 399, dtype: int64
```

In [7]:
```
#train test split formula
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)
```

In [8]:
```
print(X_train)
```

```
       c1      c2     c3    c4     c5     c6     c7      c8  c9
23  -2.679  23.071  5.366 -7.23  15.27  -2.76  77.50  102.76  20
254 -4.446  21.304  3.134  0.81  23.31  11.49  91.75  102.76  32
150 -3.220  22.530  3.508 -7.23  15.27  -2.76  77.50  102.76  30
276 -3.979  21.771  3.069 -0.80  21.70  -3.24  77.02  102.76  34
294  1.475  27.225  3.024 -0.80  21.70  -3.24  77.02  102.76  34
..     ...     ...    ...   ...    ...    ...    ...     ...  ..
315  3.252  29.002  2.963  4.02  26.52  -1.45  78.81  102.76  34
158  0.699  26.449  3.471  0.81  23.31  11.49  91.75  102.76  30
25  -4.552  21.198  5.162  4.02  26.52  -1.45  78.81  102.76  20
20   0.483  26.233  5.396 -7.23  15.27  -2.76  77.50  102.76  20
349 -4.080  21.670  2.890 -0.80  21.70  -3.24  77.02  102.76  36

[319 rows x 9 columns]
```

In [9]:
```
#testX
print(X_test)
```

```
       c1      c2     c3    c4     c5     c6     c7      c8  c9
362 -2.224  23.526  2.863 -7.23  15.27  -2.76  77.50  102.76  36
267 -6.842  18.908  3.100 -0.80  21.70  -3.24  77.02  102.76  34
63   6.125  31.875  4.335 -7.23  15.27  -2.76  77.50  102.76  24
54  -6.871  18.879  4.434 -0.80  21.70  -3.24  77.02  102.76  24
298  7.686  33.436  3.006  4.02  26.52  -1.45  78.81  102.76  34
..     ...     ...    ...   ...    ...    ...    ...     ...  ..
300 -6.279  19.471  3.001  0.81  23.31  11.49  91.75  102.76  34
189 -6.460  19.290  3.352  0.81  23.31  11.49  91.75  102.76  30
162 -1.711  24.039  3.456  0.81  23.31  11.49  91.75  102.76  30
246  5.594  31.344  3.155 -7.23  15.27  -2.76  77.50  102.76  32
44   1.715  27.465  4.686 -7.23  15.27  -2.76  77.50  102.76  22

[80 rows x 9 columns]
```

In [10]:
```
#test Y
print(Y_train)
```

```
23     0
254    0
150    0
276    0
294    0
      ..
315    0
158    0
25     0
20     0
349    1
Name: c10, Length: 319, dtype: int64
```

In [34]:
```
#print Y
print(Y_test)
```

```
324    0
155    0
19     0
49     0
297    0
      ..
61     0
62     0
75     0
12     0
76     0
Name: c10, Length: 80, dtype: int64
```

In [ ]:
```
print('Linear Regression:')
print()
reg = LinearRegression()
reg.fit(X_train, Y_train)
Y_pred = reg.predict(X_test)
```

In [ ]:
```
print(Y_pred)
```

In [ ]:
```
import numpy as np
```

In [ ]:
```
print('Accuracy:',reg.score(X_train, Y_train)*100)
print('Mean Absolute Error:', metrics.mean_absolute_error(Y_test, Y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(Y_test, Y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(Y_test, Y_pred)))
#sns.scatterplot(Y_pred, Y_test)
```

In [40]:
```
from sklearn.ensemble import RandomForestRegressor
print('Random Forest Regressor:')
print()
rfr = RandomForestRegressor()
rfr.fit(X_train,Y_train)
Y_pred = rfr.predict(X_test)
print('Accuracy:',rfr.score(X_test, Y_test)*100)
print('Mean Absolute Error:', metrics.mean_absolute_error(Y_test, Y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(Y_test, Y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(Y_test, Y_pred)))
```

```
Random Forest Regressor:

Accuracy: 98.71799687010954
Mean Absolute Error: 0.003999999999999999
Mean Squared Error: 0.0012799999999999995
Root Mean Squared Error: 0.035777087639996624
```

In [ ]:
```
#global green house gas emmision - 78%
#India's green house gas emission -7%
#Total emmission rate - 5.46%
#Bikanar - highest heat emmited
#Amritser and batla - lowest emmiting cities
```

In [ ]:
```
print(Y_test,Y_pred)
```

In [12]:
```
#TITTLE OF THE PROJECT : GLOBAL WARMING ANALAYSIS OF INDIA:An earth hour survey
```

In [ ]:
```

```