Linear Regression

Steps Include

- 1. Import Libraries
- 1. Import Dataset and Preprocessing
- 1. Exploratory Data Analysis
- 1. Divide the dataset into test and train
- 1. Training the Linear Regression Model
- 1. Testing the Linear Regression Model
- 1. Evaluate Predictions

1. Import Libraries

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

2. Import Dataset and Preprocessing

In [2]:
```
df=pd.read_csv("IceCreamData.csv")
df.head()
```

Out[2]:

	Temperature	Revenue
0	24.566884	534.799028
1	26.005191	625.190122
2	27.790554	660.632289
3	20.595335	487.706960
4	11.503498	316.240194

In [3]:
```
df.shape
```

Out[3]:
```
(500, 2)
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Temperature  500 non-null    float64
 1   Revenue      500 non-null    float64
dtypes: float64(2)
memory usage: 7.9 KB
```

In [12]:
```
df.isnull().sum()
```

Out[12]:
```
Temperature    0
Revenue        0
dtype: int64
```

In [9]:
```
df.describe().round(2)
```

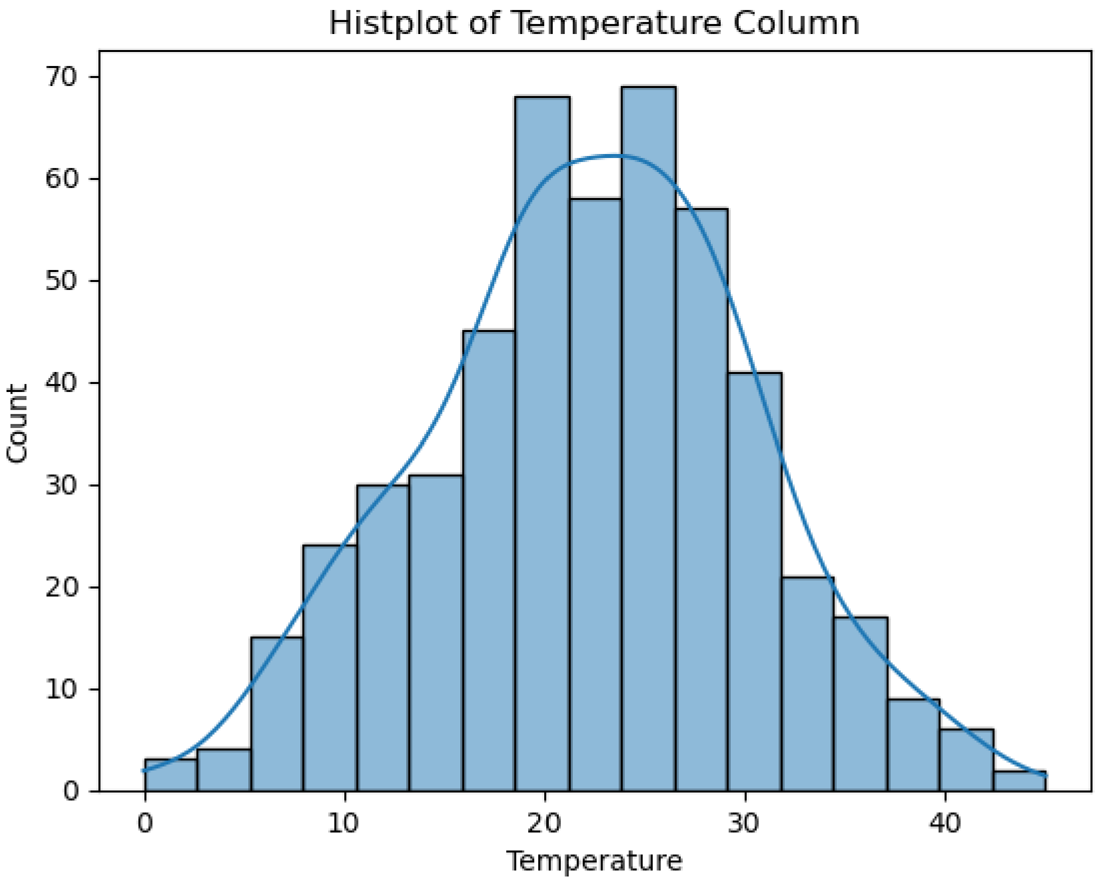
Out[9]:

	Temperature	Revenue
count	500.00	500.00
mean	22.23	521.57
std	8.10	175.40
min	0.00	10.00
25%	17.12	405.56
50%	22.39	529.37
75%	27.74	642.26
max	45.00	1000.00

3. Exploratory Data Analysis

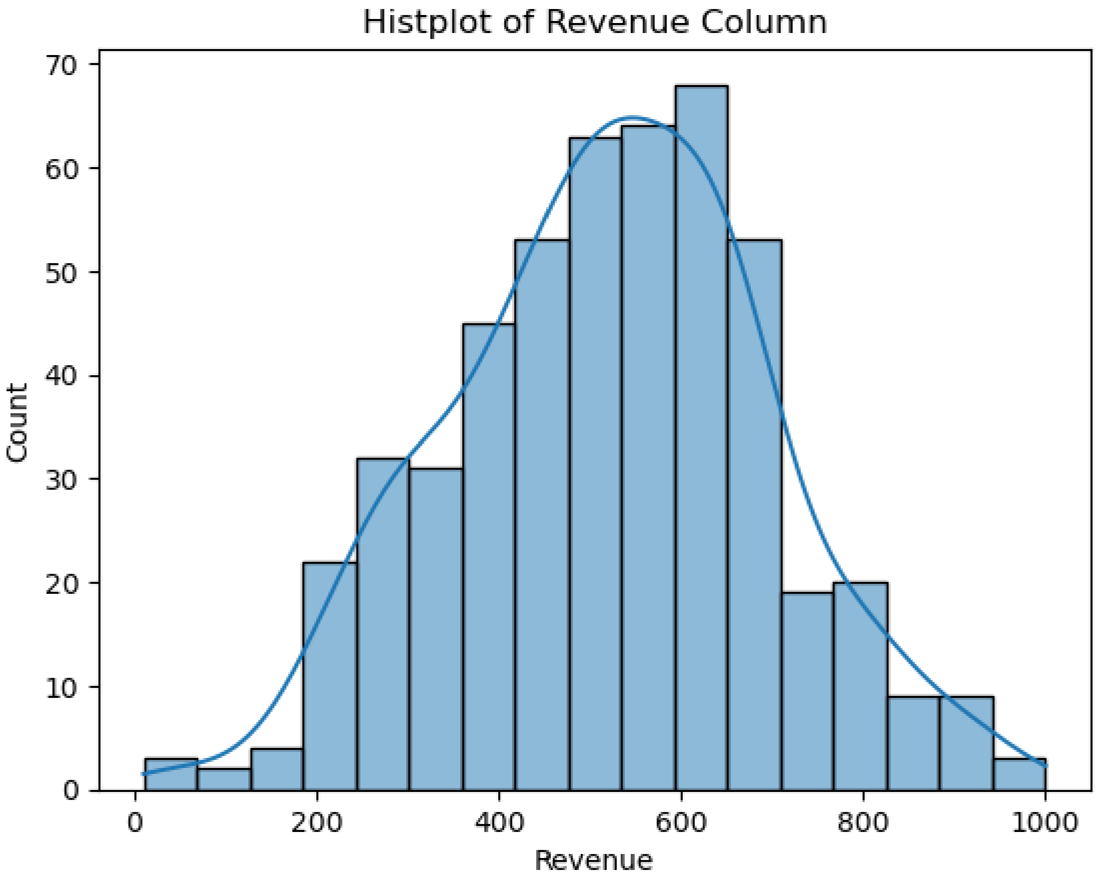
In [83]:
```
plt.title("Histplot of Temperature Column")
sns.histplot(data=df,x="Temperature",kde=True)
```

Out[83]:
```
<Axes: title={'center': 'Histplot of Temperature Column'}, xlabel='Temperat
ure', ylabel='Count'>
```



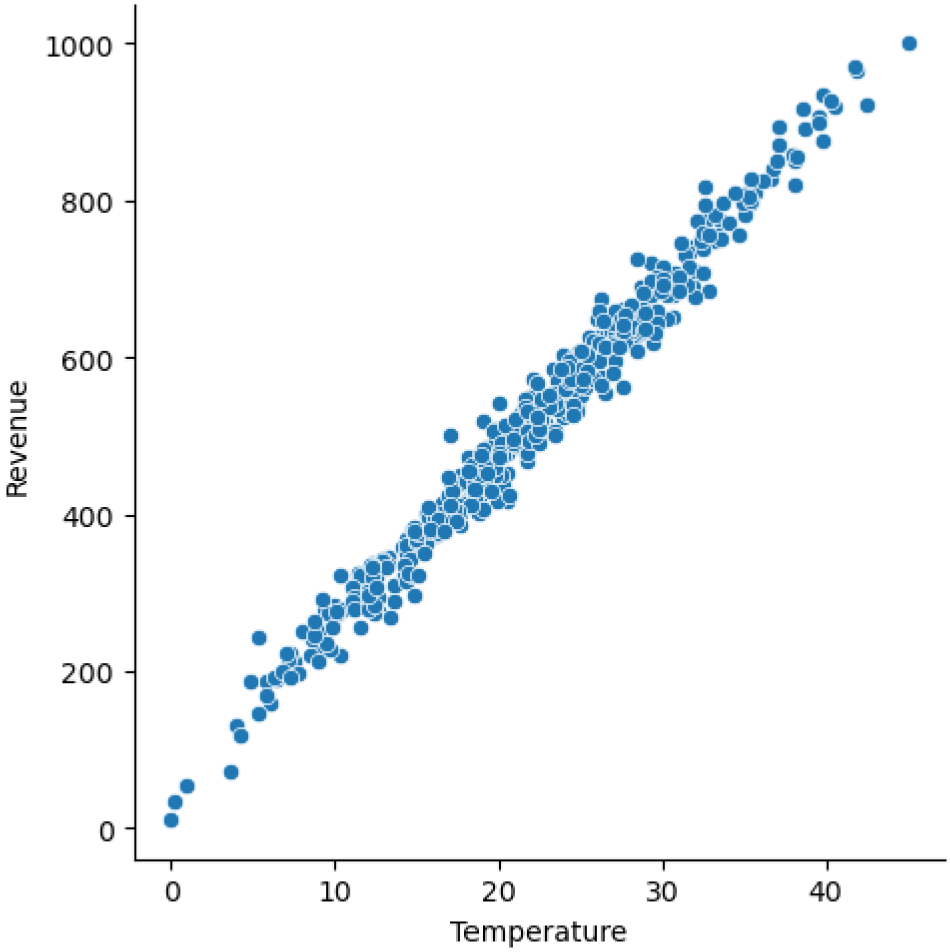
In [84]:
```
plt.title("Histplot of Revenue Column")
sns.histplot(data=df,x="Revenue",kde=True)
```

Out[84]: <Axes: title={'center': 'Histplot of Revenue Column'}, xlabel='Revenue', ylabel='Count'>



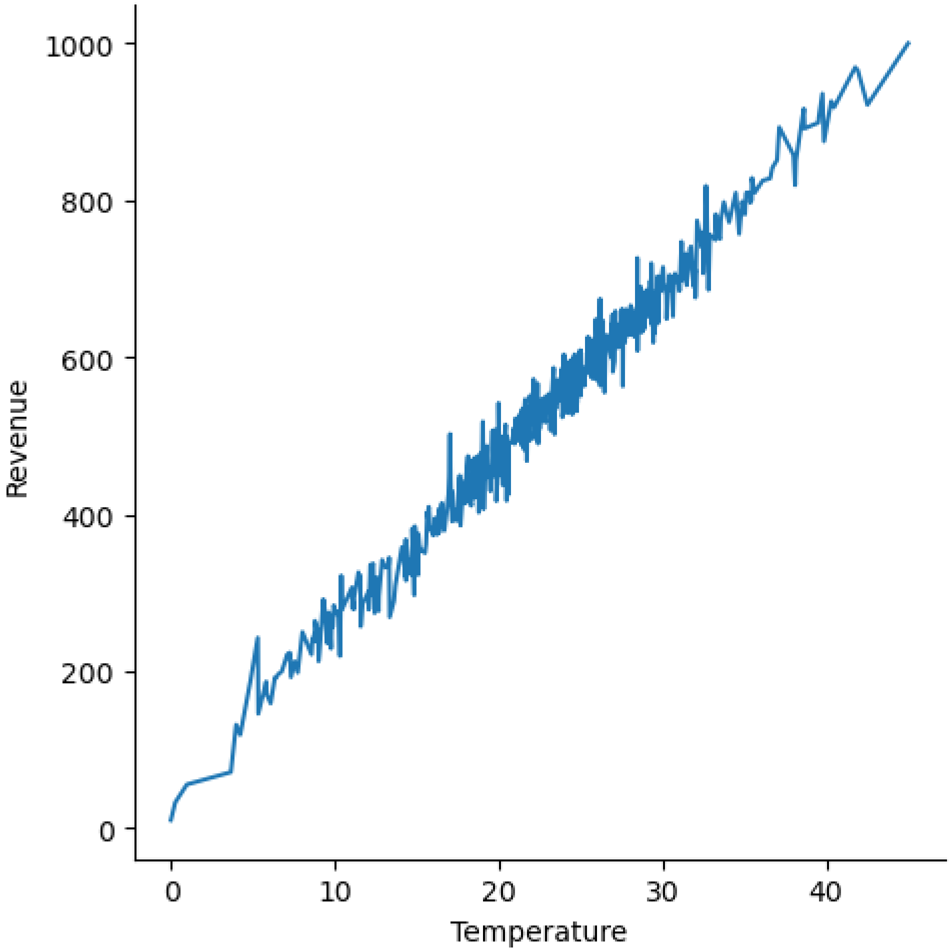
In [87]:
```
# plt.title("Scatter Plot between Temperature and Revenue")
sns.relplot(data=df,x="Temperature",y="Revenue")
```

Out[87]: <seaborn.axisgrid.FacetGrid at 0x16932b090>



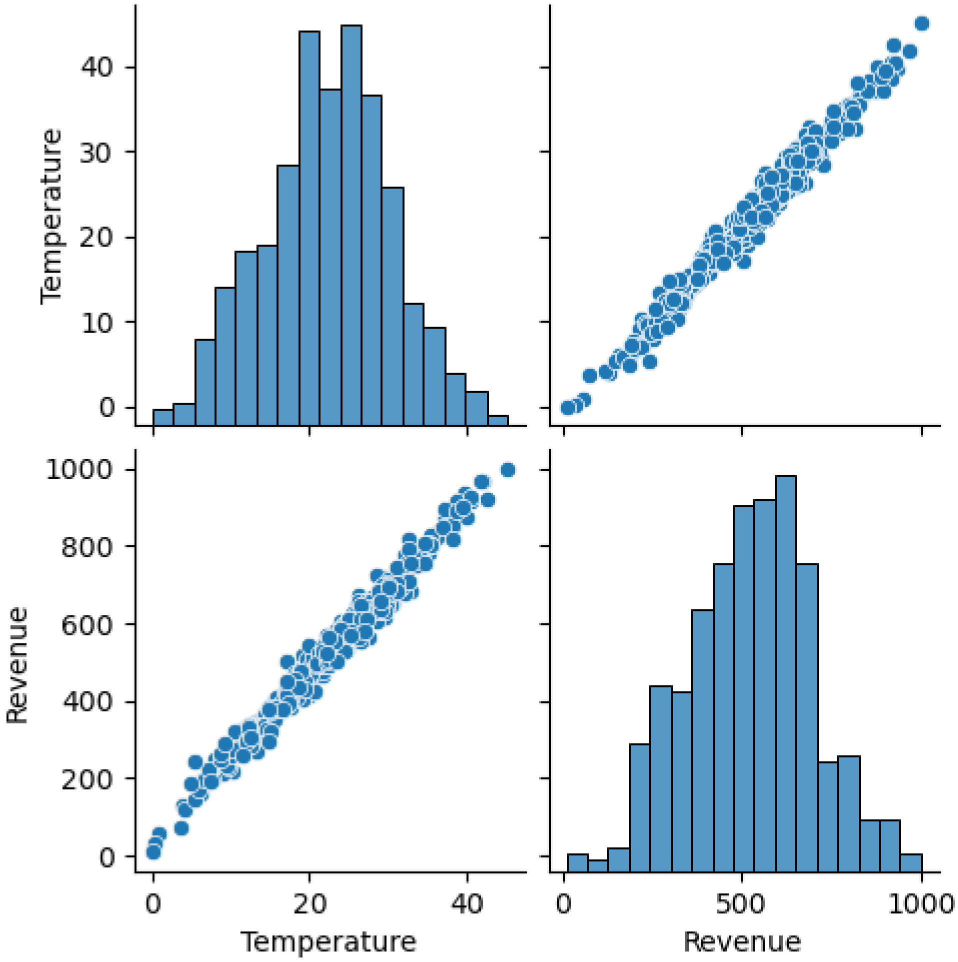
In [11]:
```
sns.relplot(data=df,x="Temperature",y="Revenue",kind="line")
```

Out[11]: <seaborn.axisgrid.FacetGrid at 0x155db8710>



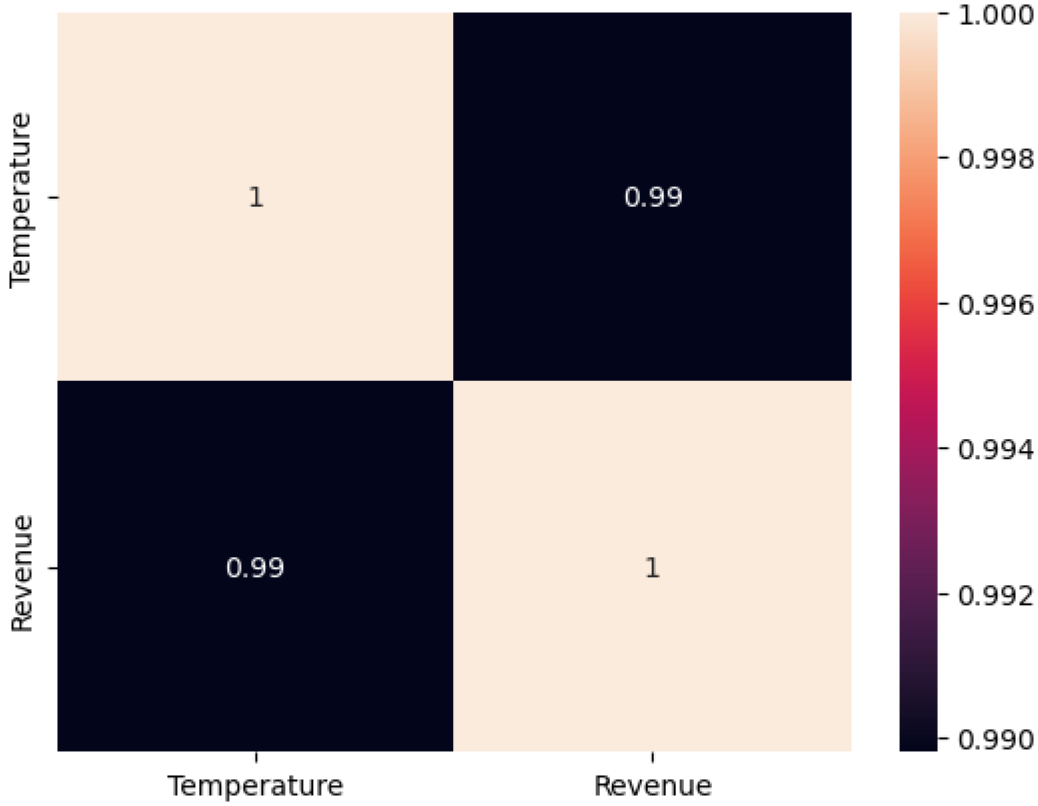
In [91]:
```
sns.pairplot(df)
```

Out[91]:
```
<seaborn.axisgrid.PairGrid at 0x16b1ef350>
```



In [15]:
```
corr=df.corr()
sns.heatmap(corr,annot=True)
```

Out[15]: <Axes: >



4. Divide the Dataset into Train and Test

In [16]:
```
from sklearn.model_selection import train_test_split
```

In [17]:
```
df.head()
```

Out[17]:

	Temperature	Revenue
0	24.566884	534.799028
1	26.005191	625.190122
2	27.790554	660.632289
3	20.595335	487.706960
4	11.503498	316.240194

In [49]:
```
df_numpy=np.array(df)
```

In [50]:
```
df_numpy.shape
```

Out[50]:
```
(500, 2)
```

In [52]:
```
x=df_numpy[:,:-1]
y=df_numpy[:,-1]
```

In [53]:
```
x.shape,y.shape
```

Out[53]:
```
((500, 1), (500,))
```

In [55]:
```
x_train.shape
```

Out[55]:
```
(400, 1)
```

In [56]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

In [57]:
```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[57]:
```
((400, 1), (100, 1), (400,), (100,))
```

5. Training the Linear Regression Model

In [59]:
```
from sklearn.linear_model import LinearRegression
```

In [61]:
```
le=LinearRegression()
```

In [62]:
```
le.fit(x_train,y_train)
```

Out[62]:
```
▾ LinearRegression
LinearRegression()
```

6. Testing the Linear Regression Model

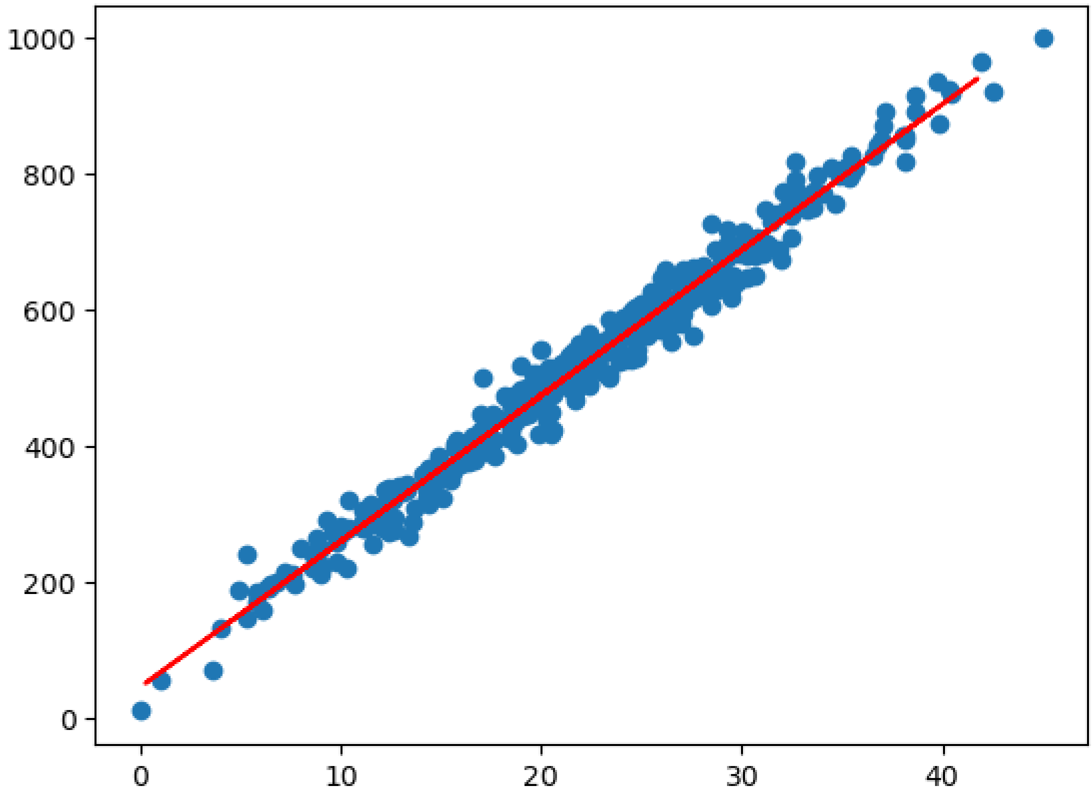
In [63]:
```
y_pred=le.predict(x_test)
```

In [69]:
```
le.score(x_test,y_test)
```

Out[69]:
```
0.9809948176474016
```

In [82]:
```
plt.scatter(x_train,y_train)
plt.plot(x_test,y_pred,color="red")
```

Out[82]:
```
[<matplotlib.lines.Line2D at 0x16a8539d0>]
```



7. Evaluate Predictions

In [66]:
```
from sklearn.metrics import accuracy_score,mean_squared_error,mean_absolute_
```

In [76]:
```
Acc = le.score(x_test,y_test)
```

In [72]:
```
MAE = mean_absolute_error(y_test,y_pred)
```

In [73]:
```
MSE = mean_squared_error(y_test,y_pred)
```

In [74]:
```
RMSE = np.sqrt(MSE)
```

In [77]:
```
R2 = r2_score(y_test,y_pred)
```

In [92]:
```
print("Accuracy Score is : ",Acc)
print("Mean Absolute Error is : ",MAE)
print("Mean Squared Error is : ",MSE)
```

```
print("Root Mean Squared Error is : ",RMSE)
print("R2 Score is : ",R2)
```

```
Accuracy Score is :  0.9809948176474016
Mean Absolute Error is :  19.47569835356527
Mean Squared Error is :  576.6401232652831
Root Mean Squared Error is :  24.013332198286918
R2 Score is :  0.9809948176474016
```