LINEAR REGRESSION: WHEATHER DATASET

In [76]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn. metrics import mean_squared_error ,r2_score
from sklearn.linear_model import LinearRegression
```

In [77]:
```
data=pd.read_csv('weather[1].csv')

data.head(4)

data.describe()
```

Out[77]:

	Temperature_c	Humidity	Wind_Speed_kmh	Wind_Bearing_degrees	Visibility_km	Pressure_millibars	Rain
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	11.857208	0.735272	10.813565	185.979700	10.372964	1004.896421	0.891200
std	9.512244	0.195498	6.972569	107.469909	4.178680	109.617120	0.311404
min	-21.822222	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	4.655556	0.600000	5.747700	112.000000	8.287475	1011.890000	1.000000
50%	11.561111	0.780000	9.933700	179.000000	10.046400	1016.500000	1.000000
75%	18.822222	0.900000	14.103600	290.000000	14.876400	1021.110000	1.000000
max	38.838889	1.000000	53.242700	359.000000	16.100000	1045.380000	1.000000

In [78]:
```
X=data[['Humidity']]
y=data['Temperature_c']

X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=0.2,random_state=42)

model=LinearRegression()

model.fit(X_train,y_train)
```

Out[78]:
```
▾ LinearRegression
LinearRegression()
```

In [79]:
```
y_pred=model.predict(X_test)

mse=mean_squared_error(y_test,y_pred)
R=r2_score(y_test,y_pred)

print(mse)
print(R)
```

```
53.91967468983329
0.3992921200155686
```

LINEAR REGRESSION:Covid-19 Daily Cases Dataset

In [111…
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.metrics import mean_squared_error,r2_score
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [104…
```
data=pd.read_csv('covid_19_clean_complete[1].csv')
data.head(5)
data.describe()
```

Out[104]:

	Lat	Long	Confirmed	Deaths	Recovered	Active
count	49068.000000	49068.000000	4.906800e+04	49068.000000	4.906800e+04	4.906800e+04
mean	21.433730	23.528236	1.688490e+04	884.179160	7.915713e+03	8.085012e+03
std	24.950320	70.442740	1.273002e+05	6313.584411	5.480092e+04	7.625890e+04
min	-51.796300	-135.000000	0.000000e+00	0.000000	0.000000e+00	-1.400000e+01
25%	7.873054	-15.310100	4.000000e+00	0.000000	0.000000e+00	0.000000e+00
50%	23.634500	21.745300	1.680000e+02	2.000000	2.900000e+01	2.600000e+01
75%	41.204380	80.771797	1.518250e+03	30.000000	6.660000e+02	6.060000e+02
max	71.706900	178.065000	4.290259e+06	148011.000000	1.846641e+06	2.816444e+06

In [105…
```
X=data[['Confirmed']]
y=data['Deaths']
```

In [106…
```
print(X,y)
```

```
       Confirmed
0              0
1              0
2              0
3              0
4              0
...          ...
49063        865
49064       1691
49065        354
49066       7235
49067        505

[49068 rows x 1 columns] 0          0
1          0
2          0
3          0
4          0
        ...
49063     14
49064    483
49065      7
49066     60
49067     12
Name: Deaths, Length: 49068, dtype: int64
```

In [107…
```
X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=0.2,random_state=42)
```

In [108…
```
model=LinearRegression()
```

In [109…
```
model.fit(X_train,y_train)
```

Out[109]:
```
▾ LinearRegression
LinearRegression()
```

In [110…
```
y_pred=model.predict(X_test)

mse=mean_squared_error(y_test,y_pred)
R=r2_score(y_test,y_pred)
print(mse,R)
```

```
6635395.310517764 0.8328043477308983
```

LINEAR REGREESION:Iris dataset

In [113…
```
import numpy as np
import pandas as pd
import matplotlib. pyplot as plt
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [119…
```
data=pd.read_csv('iris.data[1].csv')
data.head(5)
data.describe()
```

Out[119]:

	5.1	3.5	1.4	0.2
count	149.000000	149.000000	149.000000	149.000000
mean	5.848322	3.051007	3.774497	1.205369
std	0.828594	0.433499	1.759651	0.761292
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.400000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [120…
```
X=data[['3.5']]
y=data['1.4']
```

In [121…
```
X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=0.2,random_state=42)
```

In [122…
```
model=LinearRegression()
```

In [123…
```
model.fit(X_train,y_train)
```

Out[123]:
```
▾ LinearRegression
LinearRegression()
```

In [125…
```
y_pred=model.predict(X_test)
mse=mean_squared_error(y_test,y_pred)

print(mse)
```

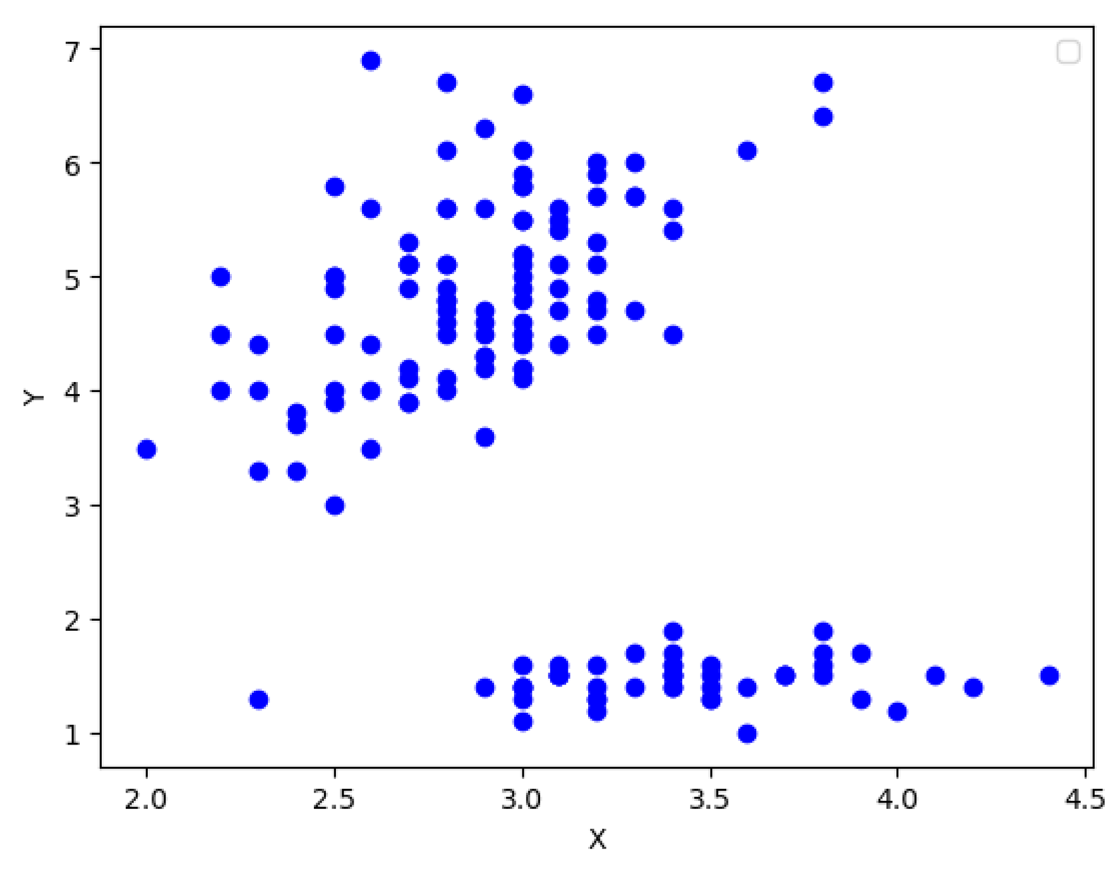
```
2.845216852745706
```

In [128…
```
plt.scatter(X, y, color='blue', label='Iris Data ')

# Plot the regression line

plt.xlabel('X')
plt.ylabel('Y')
plt.legend('')

# Show the plot
plt.show()
```



LINEArEGRESSION:studentrecord

In [146…
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.metrics import mean_squared_error,r2_score

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [147…
```
data=pd.read_csv('studentrecord.csv')
data.head(5)
data.describe()
```

Out[147]:

	subject	weight	age	bloodfat
count	11.000000	11.000000	11.000000	11.000000
mean	6.000000	68.454545	38.090909	328.545455
std	3.316625	14.995757	13.692732	84.575840
min	1.000000	27.000000	20.000000	190.000000
25%	3.500000	67.000000	26.500000	275.500000
50%	6.000000	72.000000	36.000000	354.000000
75%	8.500000	75.500000	49.000000	393.500000
max	11.000000	84.000000	57.000000	451.000000

In [148…
```
X=data[['age']]
y=data['bloodfat']
```

In [149…
```
X_train,X_test,y_train,y_test = train_test_split(X,y,train_size=0.2,random_state=42)
```

In [150…
```
model=LinearRegression()
model.fit(X_train,y_train)
```

Out[150]:
```
▾ LinearRegression
LinearRegression()
```

In [151…
```
y_pred=model.predict(X_train)
mse=mean_squared_error(y_train,y_pred)
mse2=r2_score(y_train,y_pred)
print(mse,mse2)
```

```
0.0 1.0
```

INEAR REGRESSION:Task dataset

In [153…
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

In [182…
```
data=pd.read_csv('Dataset[1].csv')
data.head(7)
data.describe()
```

Out[182]:

	Quantity	UnitPrice	CustomerID
count	541909.000000	541909.000000	406829.000000
mean	9.552250	4.611114	15287.690570
std	218.081158	96.759853	1713.600303
min	-80995.000000	-11062.060000	12346.000000
25%	1.000000	1.250000	13953.000000
50%	3.000000	2.080000	15152.000000
75%	10.000000	4.130000	16791.000000
max	80995.000000	38970.000000	18287.000000

In [190…
```
X=data[['UnitPrice']]
y=data['Quantity']
```

In [191…
```
X_train,X_test,y_train,y_test = train_test_split(X,y,train_size=0.2,random_state=42)
```

In [192…
```
model=LinearRegression()
model.fit(X_train,y_train)
```

Out[192]:
```
▾ LinearRegression
LinearRegression()
```

In [193…
```
y_pred=model.predict(X_train)
mse=mean_squared_error(y_train,y_pred)

print(mse)
```

```
61838.113753699006
```

LINEAR REGRESSION:Salary dataset

In [203…
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

In [205…
```
data=pd.read_csv('Salary_dataset[1].csv')
data.head(2)
data.describe()
```

Out[205]:

	Unnamed: 0	YearsExperience	Salary
count	30.000000	30.000000	30.000000
mean	14.500000	5.413333	76004.000000
std	8.803408	2.837888	27414.429785
min	0.000000	1.200000	37732.000000
25%	7.250000	3.300000	56721.750000
50%	14.500000	4.800000	65238.000000
75%	21.750000	7.800000	100545.750000
max	29.000000	10.600000	122392.000000

In [198…
```
X=data[['YearsExperience']]
y=data['Salary']
```

In [199…
```
X_train,X_test,y_train,y_test = train_test_split(X,y,train_size=0.2,random_state=42)
```

In [200…
```
model=LinearRegression()
model.fit(X_train,y_train)
```

Out[200]:
```
▾ LinearRegression
LinearRegression()
```

In [201…
```
y_pred=model.predict(X_train)
mse=mean_squared_error(y_train,y_pred)

print(mse)
```

```
31753229.30836155
```

In [ ]: