Double-click (or enter) to edit

```
import pandas as pd
```

```
import numpy as np
```

Import CSV as Dataframe

```
df = pd.read_csv(r'https://github.com/YBI-Foundation/Dataset/raw/main/Servo%20M
```

Get the First Five Rows of Dataframe

```
df.head()
```

	Motor	Screw	Pgain	Vgain	Class
0	E	E	5	4	4
1	B	D	6	5	11
2	D	D	4	3	6
3	B	A	3	2	48
4	D	B	6	5	6

Get summary Statistics

```
df.describe()
```

Get columns Names

```
df.shape
```

```
(167, 5)
```

Get Categories and Counts Of Categorical Variables

```
df[['Motor']].value_counts()
```

```
Motor
C        40
A        36
B        36
E        33
D        22
dtype: int64
```

```
df[['Screw']].value_counts()
```

```
Screw
A        42
B        35
C        31
D        30
E        29
dtype: int64
```

Get Encoding of Categorical Features

```
df.replace({'Motor':{'A':0,'B':1,'C':2,'D':3,'E':4}}, inplace=True)
```

```
df.replace({'Screw':{'A':0,'B':1,'C':2,'D':3,'E':4}}, inplace=True)
```

Define y(dependent or label or target variable) and x (independent or features or attribute Variable)

```
y= df['Class']
```

```
y.shape
```

```
(167,)
```

```
y
```

```
0       4
1      11
2       6
3      48
4       6
       ..
162    44
163    40
164    25
165    44
166    20
Name: Class, Length: 167, dtype: int64
```

```
x = df[['Motor','Screw','Pgain','Vgain']]
```

```
x.shape
```

```
(167, 4)
```

```
x
```

	Motor	Screw	Pgain	Vgain
0	4	4	5	4
1	1	3	6	5
2	3	3	4	3
3	1	0	3	2
4	3	1	6	5
...	...	...	...	...
162	1	2	3	2
163	1	4	3	1
164	2	3	4	3
165	0	1	3	2
166	0	0	6	5

167 rows × 4 columns

Get Train Test Split

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.3, rando
```

```
x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

```
((116, 4), (51, 4), (116,), (51,))
```

Get model Train

```
from sklearn.linear_model import LinearRegression
```

```
lr = LinearRegression()
```

```
lr.fit(x_train, y_train)
```

```
LinearRegression()
```

Get Model Prediction

```
y_pred = lr.predict(x_test)
```

```
y_pred.shape
```

```
(51,)
```

```
y_pred
```

```
array([24.55945258, 30.98765106, 18.54485477, 25.51524243, 38.56082023,
       23.52007775, 11.61947065, 20.03335614, 40.60404401, 41.7009556 ,
       13.66269443, 26.01242807, 16.50163099, 16.54663453, 21.92598051,
       22.52570646, -5.46449561, 30.68912392, 32.7323477 ,  1.41282941,
       33.97718702, 31.63543611, 33.52806048, 30.04133887, 19.38557109,
        6.49364826, 28.5528375 , 17.04382017, 25.06611589,  3.50411229,
       30.59606128, 23.67067716, 35.72188367, 32.08456265, 12.46018697,
        3.6547117 , 23.47201865, 33.03087484, 17.49294672, 37.61450804,
       27.54898855, 22.07657992, 11.51387478,  9.470651  , 30.53852451,
       28.64590014, 33.67865989,  4.60102388, 24.1198037 , 21.13026773,
       25.71390094])
```

Get Model Evaluation

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

```
mean_squared_error(y_test, y_pred)
```

```
66.03589175595563
```

```
mean_absolute_error(y_test, y_pred)
```

```
7.190539677251235
```

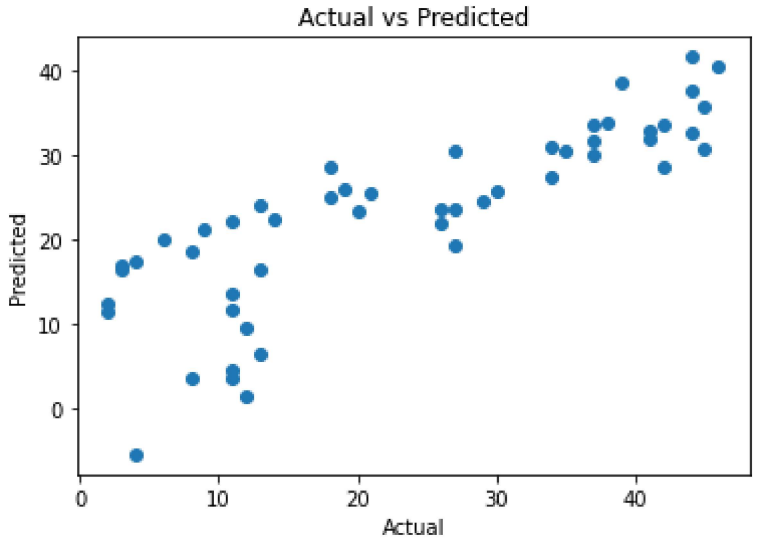
```
r2_score(y_test, y_pred)
```

```
0.6807245170563927
```

Get Visualization of Actual VS Predicted Results

```
import matplotlib.pyplot as plt
plt.scatter(y_test, y_pred)
plt.xlabel("Actual")
plt.ylabel("Predicted")
plt.title("Actual vs Predicted")
plt.show
```

```
<function matplotlib.pyplot.show>
```



Get Future Predictions

1. extract a random row using sample function
2. Separate x and y
3. Standardize x
4. Predict

```
x_new = df.sample(1)
```

```
x_new
```

	Motor	Screw	Pgain	Vgain	Class
17	1	2	5	4	9

```
x_new.shape
```

```
(1, 5)
```

```
x_new = x_new.drop('Class',axis = 1)
```

```
x_new
```

	Motor	Screw	Pgain	Vgain
17	1	2	5	4

```
x_new.shape
```

```
(1, 4)
```

```
y_pred_new = lr.predict(x_new)
```

```
y_pred_new
```

```
array([16.65223039])
```