


```
import pandas as pd

import numpy as np


df = pd.read_csv('https://github.com/YBI-Foundation/Dataset/raw/main/Servo%20Mechanism.csv')

df.describe()
```



	Pgain	Vgain	Class
count	167.000000	167.000000	167.000000
mean	4.155689	2.538922	21.173653
std	1.017770	1.369850	13.908038
min	3.000000	1.000000	1.000000
25%	3.000000	1.000000	10.500000
50%	4.000000	2.000000	18.000000
75%	5.000000	4.000000	33.500000
max	6.000000	5.000000	51.000000

```
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

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 167 entries, 0 to 166
Data columns (total 5 columns):
#   Column  Non-Null Count  Dtype
---  -
0   Motor   167 non-null     object
1   Screw   167 non-null     object
2   Pgain   167 non-null     int64
3   Vgain   167 non-null     int64
4   Class   167 non-null     int64
dtypes: int64(3), object(2)
memory usage: 6.6+ KB
```

```
df.shape
```

(167, 5)

```
df.columns
```

Index(['Motor', 'Screw', 'Pgain', 'Vgain', 'Class'], dtype='object')

```
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
```

```
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)
```

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

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

```
x.shape
y.shape
```

```
(167,)
```

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

```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.3,random_state=2529)
```

```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
```

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

```
LinearRegression
LinearRegression()
```

```
model.intercept_
```

```
77.90491786252491
```

```
model.coef_
```

```
array([ -1.67743858, -2.63837066, -16.18303123,  6.39878192])
```

```
y_predict=model.predict(x_test)
```

```
from sklearn.metrics import mean_absolute_percentage_error
```

```
mean_absolute_percentage_error(y_test,y_predict)
```

```
0.590558586252214
```

✓ 0s completed at 12:48 PM

