Servo Mechanism

A servo mechanism is an electromagnetic device that converts electricity into precise controlled motion by use of negative feedback mechanisms. Servos can be used to generate linear or circular motion, depending on their type.

Dataset:

The source of data source is from github repository by YBI Foundation.

[1]:
```
#import libraries
import pandas as pd
```

[2]:
```
import numpy as np
```

[3]:
```
#import datset
servo=pd.read_csv('Servo Mechanism.csv')
```

[4]:
```
servo.head()
```

[4]:

	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

[5]:
```
#describe data
servo.describe()
```

[5]:

	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

[6]:
```
#data visualization
servo.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: 6.7+ KB
```

[7]:
```
#data preprocessing
servo.shape
```

[7]:
```
(167, 5)
```

[8]:
```
servo[['Motor']].value_counts()
```

[8]:
```
Motor
C       40
A       36
B       36
E       33
D       22
dtype: int64
```

[9]:
```
servo.replace({'Motor':{'A':0,'B':1,'C':2,'D':3,'E':4}},inplace=True)
```

[10]:
```
servo.replace({'Screw':{'A':0,'B':1,'C':2,'D':3,'E':4}},inplace=True)
```

[11]:
```
#define taget(y) and feature(X)
```

[12]:
```
servo.columns
```

[12]:
```
Index(['Motor', 'Screw', 'Pgain', 'Vgain', 'Class'],
dtype='object')
```

[13]:
```
y=servo['Class']
```

[14]:
```
y.shape
```

[14]: (167,)

[15]:
```
y
```

[15]:
```
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
```

[16]:
```
X=servo[['Motor','Screw','Pgain','Vgain']]
```

[17]:
```
X.shape
```

[17]: (167, 4)

[18]:
```
X
```

[18]:

	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 x 4 columns]
```

[19]:
```
#train test split
```

[20]:
```
from sklearn.model_selection import train_test_split
```

[21]:
```
X_train, X_test,y_train,y_test=train_test_split(X,y, test_size=0.
↪3,random_state=2529)
```

[22]:
```
X_train.shape, X_test.shape,y_train.shape,y_test.shape
```

[22]:
```
((116, 4), (51, 4), (116,) , (51,))
```

[23]:
```
#modelling
from sklearn.linear_model import LinearRegression
```

[24]:
```
LR=LinearRegression()
```

[25]:
```
LR.fit(X_train,y_train)
```

[25]:
```
LinearRegression()
```

[26]:
```
#model prediction
```

[27]:
```
y_pred=LR.predict(X_test)
```

[28]:
```
y_pred
```

[28]:
```
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])
```

[29]:
```
y_pred.shape
```

[29]: (51,)

[30]:
```
#model evaluation
from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score
```

[31]:
```
mean_squared_error(y_test,y_pred)
```

[31]: 66.03589175595566

[32]:
```
mean_absolute_error(y_test,y_pred)
```

[32]: 7.190539677251239

[33]:
```
r2_score(y_test,y_pred)
```

[33]: 0.6807245170563926

Conclusion:

The linear regression algorithm is been applied to the data set. And all the useful data is been extracted. The string values will be categorial and converted to float.