ybi-internship-project-2

September 11, 2024

#

SERVO PREDICTION USING LINEAR REGRESSION

0.1 OBJECTIVE

- **To import and analyse the data**
- **Then visualize the data using various plot**
- **Now clean and preprocess the data**
- **Then split train-teat data and perform modeling**

0.2 Data Source

The data is downloaded from https://github.com/YBI-Foundation/Dataset/raw/main/Servo%20Mechanism.csv

This is in the CSV format

1 IMPORTING THE LIBRARIES

[1]:

```
import pandas as pd
import numpy as np

import seaborn as sns
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings('ignore')

from sklearn.model_selection import train_test_split

from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

from sklearn.linear_model import LinearRegression
```

2 IMPORTING DATA

[2]:

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

[3]:

```
df.head()
```

[3]:

```
  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
```

3 DESCRIBE DATA

[4]:

```
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
```

[5]:

```
df.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]:

```
df.columns
```

[6]:

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

[7]:
```
df.shape
```

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

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

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

[9]:
```
df[['Screw']].value_counts()
```

[9]:
```
Screw
A        42
B        35
C        31
D        30
E        29
dtype: int64
```

[10]:
```
numerical_features = df.select_dtypes(include = [np.number]).columns
categorical_features = df.select_dtypes(include = [np.object]).columns
```

[11]:
```
numerical_features
```

[11]:
```
Index(['Pgain', 'Vgain', 'Class'], dtype='object')
```

[12]:
```
categorical_features
```

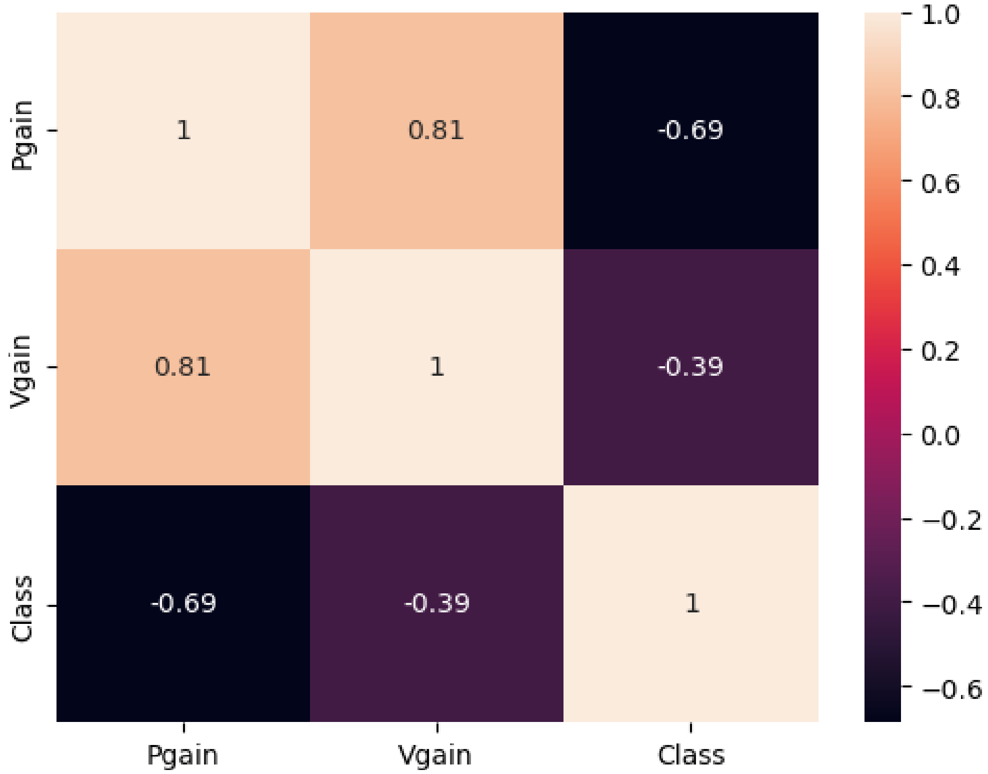
[12]:
```
Index(['Motor', 'Screw'], dtype='object')
```

4 DATA VISUALIZATION

4.1 Heatmap Correlation

[13]:
```
df.corr()
#plotting the correlation
plt.figure(1)
sns.heatmap(df.corr(), annot = True)
```

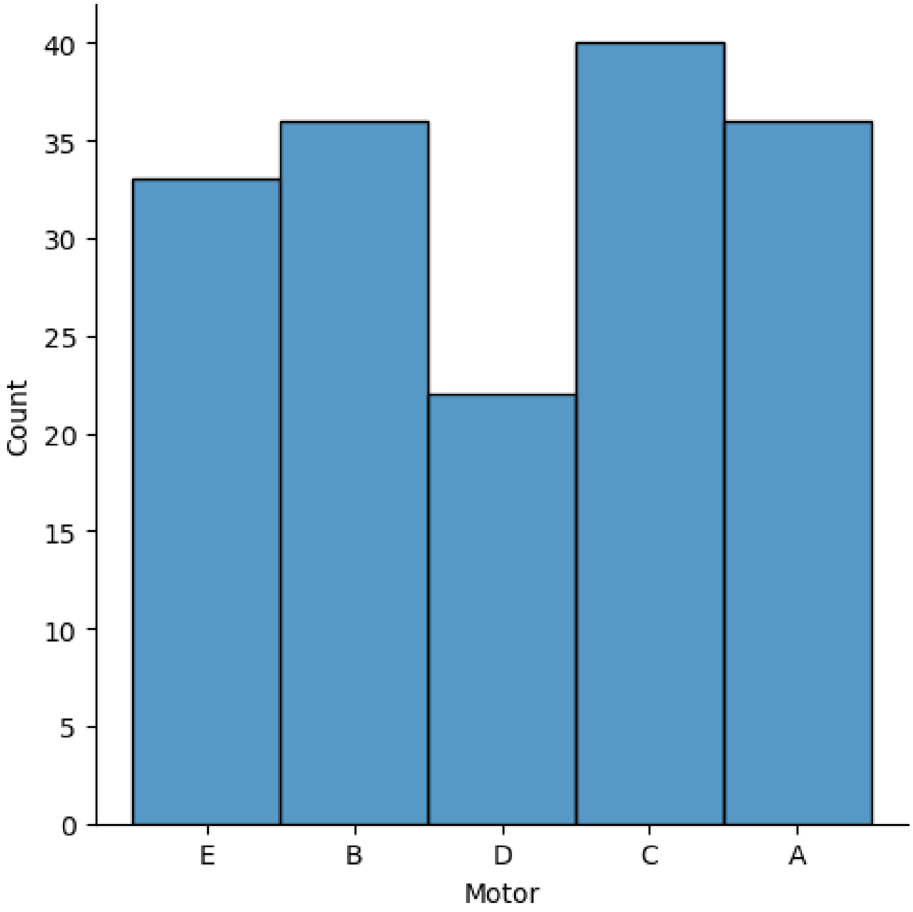
[13]:
```
<Axes: >
```



4.2 Univariate Analysis

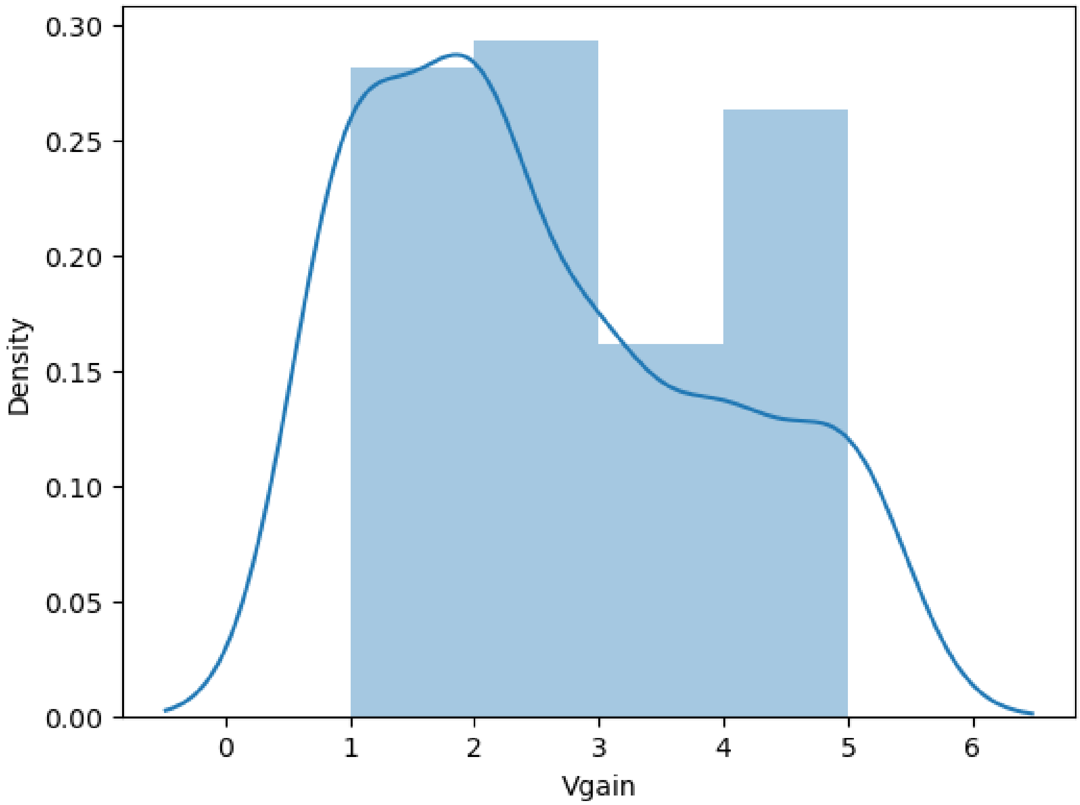
[14]:
```
sns.displot(df.Motor)
```

[14]:
```
<seaborn.axisgrid.FacetGrid at 0x78977b050250>
```



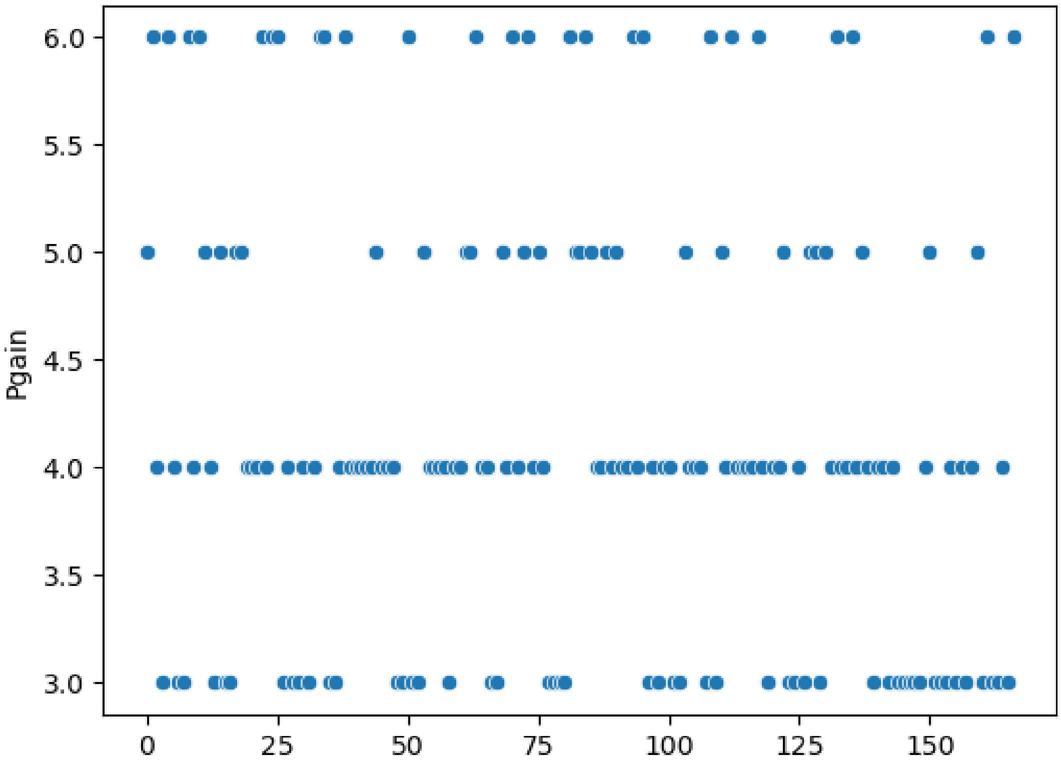
```
[15]: sns.distplot(df['Vgain'])
```

```
[15]: <Axes: xlabel='Vgain', ylabel='Density'>
```



[16]:
```
sns.scatterplot(df.Pgain)
```

[16]:
```
<Axes: ylabel='Pgain'>
```



4.3 Multivariate Analysis

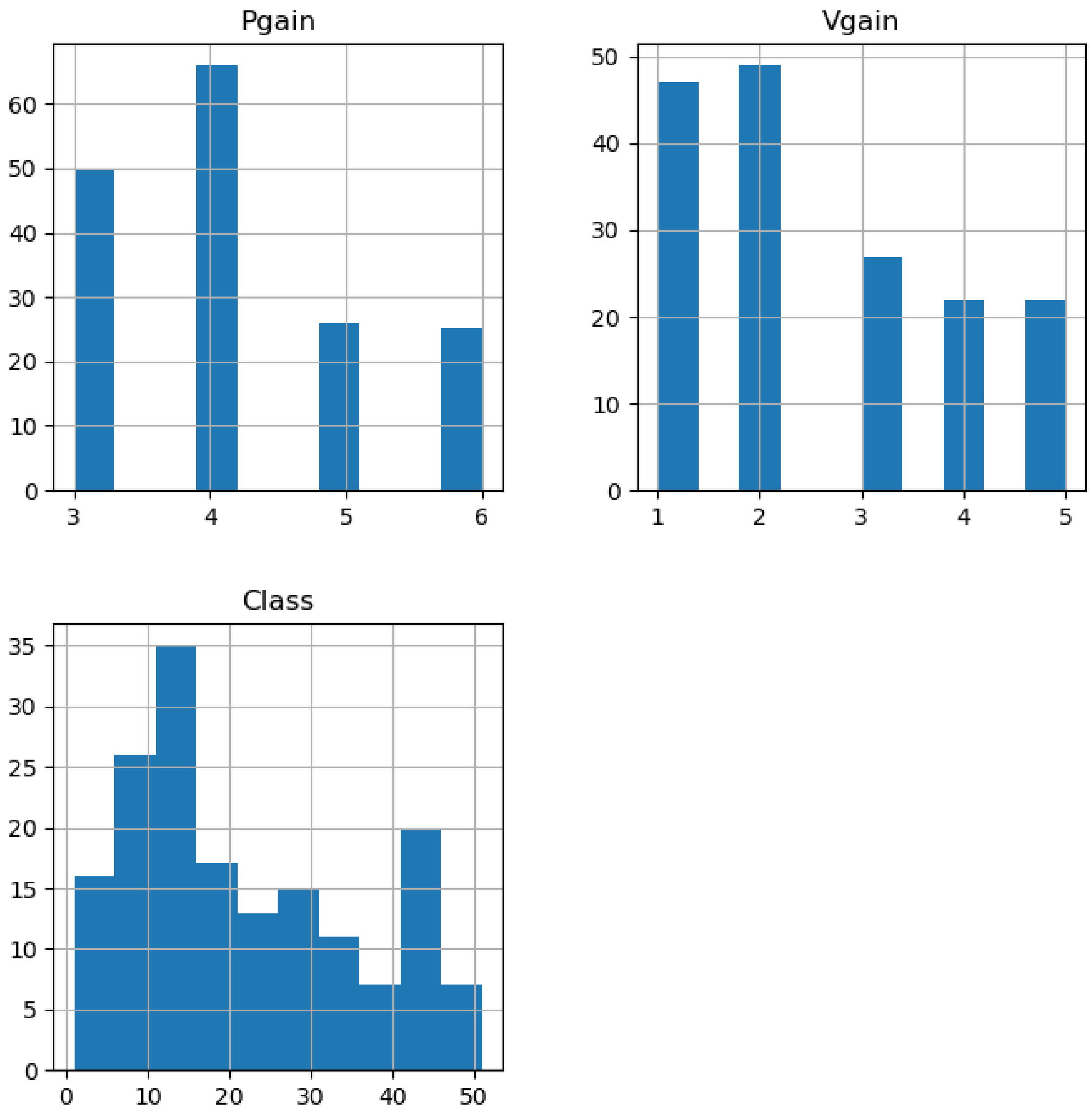
[17]:
```
plt.figure()

import plotly.express as px

fig = px.scatter_3d(df, x='Motor', y='Vgain', z='Screw',#hue='Pgain')

color='Pgain')

fig.show()
```

```
<Figure size 640x480 with 0 Axes>
```

4.4 Histogram Analysis

[18]:
```
df.hist(figsize=(8,8))
```

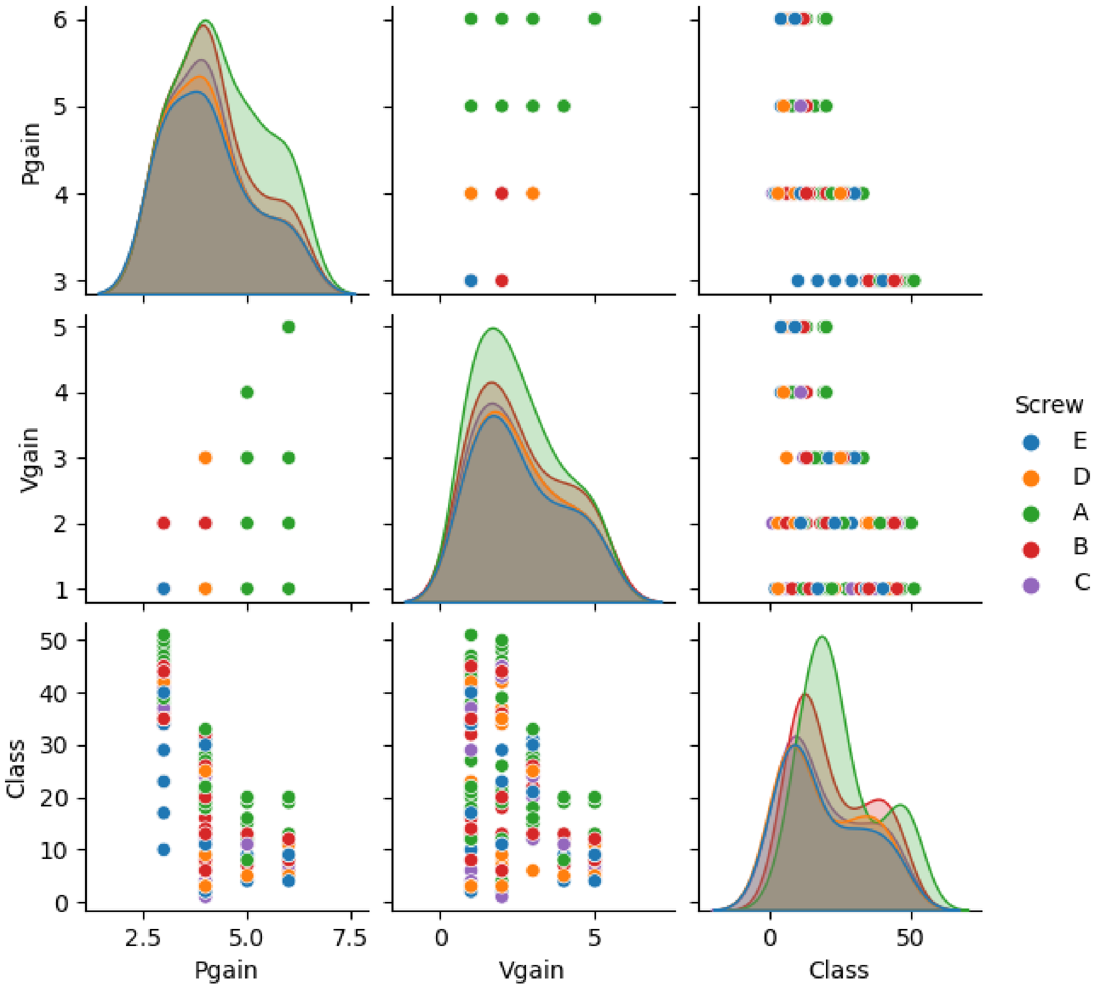
[18]:
```
array([[<Axes: title={'center': 'Pgain'}>,
        <Axes: title={'center': 'Vgain'}>],
       [<Axes: title={'center': 'Class'}>, <Axes: >]], dtype=object)
```



4.5 Pairplot Analysis

```
[19]: sns.pairplot(df,hue='Screw',size=2)
```

```
[19]: <seaborn.axisgrid.PairGrid at 0x789778c978e0>
```



5 DATA PREPROCESSING

[20]:
```
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)
```

[21]:
```
df.head()
```

[21]:
```
   Motor  Screw  Pgain  Vgain  Class
0      4      4      5      4      4
1      1      3      6      5     11
2      3      3      4      3      6
3      1      0      3      2     48
4      3      1      6      5      6
```

6 Define Target Variable (y) and Feature Variables (X)

[22]:
```
y = df['Class']
x = df[['Motor', 'Screw', 'Pgain', 'Vgain']]
```

7 TRAIN-TEST SPLIT

[23]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,random_state=2529)
```

[24]:
```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

[24]:
```
((125, 4), (42, 4), (125,), (42,))
```

8 MODELING

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

[26]:
```
lr.fit(x_train,y_train)
```

[26]:
```
LinearRegression()
```

9 PREDICTION

[27]:
```
y_pred = lr.predict(x_test)
```

[28]:
```
y_pred.shape
```

[28]:
```
(42,)
```

[29]:
```
y_pred
```

[29]:
```
array([25.56654851, 31.24940687, 19.04876502, 24.47852818, 39.37546387,
       24.37566459, 11.92192754, 19.94086363, 41.22109189, 41.9215261 ,
       13.76755557, 25.9225073 , 17.203137  , 15.50606268, 22.23125126,
       21.48770634, -4.92349976, 31.10343258, 32.9490606 ,  0.46057329,
       34.2402287 , 32.24862639, 34.53901401, 30.10421306, 18.94164411,
        7.03978758, 29.21211445, 16.9500418 , 24.77731349,  4.04896574,
       31.05774247, 23.93090499, 35.93988243, 31.94984108, 11.81480663,
        3.60420614, 22.63290015, 33.09503489, 16.65125649, 38.23027006,
       28.55737035, 21.78649166])
```

10 MODEL EVALUATION

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

```
[30]: 60.211561593192876
```

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

```
[31]: 6.735955685746795
```

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

```
[32]: 0.706476624263817
```

```
[ ]:
```