```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

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

df.head()

df.nunique()

df.info()

df.describe()

df.corr()

df = df.dropna()

df.info()

sns.pairplot(df,x_vars=['displacement','horsepower','weight','acceleration','mpg'], y_vars=['mpg']);

sns.regplot(x = 'displacement', y = 'mpg', data = df);

df.columns

y = df['mpg']

y.shape

X = df[['displacement','horsepower','weight','acceleration']]

X.shape

X

from sklearn.preprocessing import StandardScaler

ss = StandardScaler()

X = ss.fit_transform(X)

X

pd.DataFrame(X).describe()

from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.7, random_state = 2529)

X_train.shape, X_test.shape, y_train.shape, y_test.shape

from sklearn.linear_model import LinearRegression

lr = LinearRegression()

lr.fit(X_train, y_train)

lr.intercept_

lr.coef_
```

Predict Test Data

```
y_pred =lr.predict(X_test)

y_pred
```

Model Accuracy

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

mean_absolute_error(y_test,y_pred)

mean_absolute_percentage_error(y_test, y_pred)

r2_score(y_test,y_pred)
```

Polynomial Regression

```
from sklearn.preprocessing import PolynomialFeatures

poly = PolynomialFeatures(degree=2,interaction_only=True, include_bias=False)

X_train2 = poly.fit_transform(X_train)

X_test2 = poly.fit_transform(X_test)

lr.fit(X_train2, y_train)

lr.intercept_

lr.coef_

y_pred_poly = lr.predict(X_test2)
```

Model Accuracy

```
mean_absolute_error(y_test,y_pred_poly)
mean_absolute_percentage_error(y_test,y_pred_poly)
r2_score(y_test, y_pred_poly)
```