import library

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

import data

```
df.pd.read_csv('https//GitHub.com/YBI.foundation/dataset/raw/main/MPG.csv')
df.read()
df.nunique()
```

data preprocessing

```
df.info()
df.describe()
df.corr()
```

remove missing values

```
df=df.dropna()
df.info()
```

data visualization

```
sns.pairplot(df,x_vars=['displacement','horizantal','wight','accelaration','mpg'],y_vars=['mpg']);
sns.regplot(x='displacement',y='mpg'data=df);
```

define target varible y and feature x

```
df.columns
y=df['mpg']
y.shape
x=df[['displacement','horizantal','waight']]
x.shape
x
```

scaling data
predict

```
from sklearn.prepocessing import standerdscaler
ss=standerdscaler()
x=ss.fit_termsform(x)
x
pd.dataframe(x).describe()
```

train test spilt data

```
from sklearn.model_selection import train_test_spilt
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((274,4),(118,4),(274),(118O))
```

predict test data

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

model accuracy

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

polynominal regression

```
from sklearn.preprocessing import polynominalfeatures
poly=polynomkinalfeatures(degree=2,interaction_only=true,include_bias=false)
lr.fit(x_train2,y_train)
lr.intercept_
ir.coef
y_pred_poly=lr.predict(x_test2)
```

model accuracy

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