```
import numpy as np
import pandas as pd
```

```
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
     /kaggle/input/automobile-dataset/Automobile_data.csv
```

```
df=pd.read_csv('/kaggle/input/automobile-dataset/Automobile_data.csv')
```

```
df
```

```
df.head(5)
```

```
df.info()
```

```
list_cate=[]
list_num=[]
for i in list(df.columns):

    if df[i].dtype=='object':

        list_cate.append(i)
    else:
        list_num.append(i)
```

```
list_cate
```

```
df['normalized-losses'].unique()
```

```
df['normalized-losses']=df['normalized-losses'].str.replace('?','145')
```

```
df['normalized-losses']=df['normalized-losses'].astype('int64')
```

```
df['make'].unique()
```

```
df['fuel-type'].unique()
```

```
df['bore'].unique()
```

```
df['bore']=df['bore'].str.replace('?','3.35')
```

```
df['bore']=df['bore'].astype('float')
```

```
df['stroke'].unique()
```

```
df['stroke']=df['stroke'].str.replace('?','3.15')
```

```
df['stroke']=df['stroke'].astype('float')
```

```
df['horsepower'].unique()
```

```
df['horsepower']=df['horsepower'].str.replace('?','130')
```

```
df['horsepower']=df['horsepower'].astype('int64')
```

```
df['peak-rpm'].unique()
```

```
df['peak-rpm']=df['peak-rpm'].str.replace('?','4500')
df['peak-rpm']=df['peak-rpm'].astype('int64')
```

```
df['price']=df['price'].str.replace('?','16000')
df['price']=df['price'].astype('int64')
```

```
df.info()
```

```
#now preprocess the categorical features
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
for i in list(df.columns):
    if df[i].dtype=='object':
        df[i]=le.fit_transform(df[i])
```

```
df
```

```
y=df['price']
x=df.drop(['price'],axis=1)
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=0,test_size=0.2)
```

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
lr=LinearRegression()
lr.fit(x_train,y_train)
pred_1=lr.predict(x_test)
score_1=r2_score(y_test,pred_1)
```

```
score_1
```

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.scatterplot(x=y_test,y=pred_1)
```

```
#its a decent fit line
```

```
from sklearn.ensemble import RandomForestRegressor
rfg=RandomForestRegressor()
rfg.fit(x_train,y_train)
pred_2=rfg.predict(x_test)
score_2=r2_score(y_test,pred_2)
```

```
score_2
```

```
sns.scatterplot(x=y_test,y=pred_2)
```

```
#slightly better than previous one
```

```
from sklearn.ensemble import GradientBoostingRegressor
gbr=GradientBoostingRegressor()
gbr.fit(x_train,y_train)
pred_3=gbr.predict(x_test)
score_3=r2_score(y_test,pred_3)
```

```
score_3
```

```
sns.scatterplot(x=y_test,y=pred_3)
```

```
from sklearn.svm import SVR
svm=SVR()
svm.fit(x_train,y_train)
pred_4=svm.predict(x_test)
score_4=r2_score(y_test,pred_4)
```

```
score_4
```

```
sns.scatterplot(x=y_test,y=pred_4)
```

```
#svm regressor gives the worst fit line
#from all the models randomforest gives the best fit line
```