```
#CAR PRICE PREDICTION


#IMPORTING LIBRARIES


import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import linear_model
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split


#LOADING THE DATASET


cars = pd.read_csv("https://raw.githubusercontent.com/amankharwal/Website-data/master/CarPrice.csv")


# head
cars.head()
```

	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocatic
0	1	3	alfa-romero giulia	gas	std	two	convertible	rwd	fro
1	2	3	alfa-romero stelvio	gas	std	two	convertible	rwd	fro
2	3	1	alfa-romero Quadrifoglio	gas	std	two	hatchback	rwd	fro
3	4	2	audi 100 ls	gas	std	four	sedan	fwd	fro
4	5	2	audi 100ls	gas	std	four	sedan	4wd	fro

5 rows × 26 columns

```
cars.tail()
```

	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation
200	201	-1	volvo 145e (sw)	gas	std	four	sedan	rwd	front
201	202	-1	volvo 144ea	gas	turbo	four	sedan	rwd	front
202	203	-1	volvo 244dl	gas	std	four	sedan	rwd	front
203	204	-1	volvo 246	diesel	turbo	four	sedan	rwd	front
204	205	-1	volvo 264gl	gas	turbo	four	sedan	rwd	front

5 rows × 26 columns

```
#DATA PREPARATION


# SPLITTING INTO X AND Y
X = cars.loc[:, ['symboling', 'fueltype', 'aspiration', 'doornumber',
       'carbody', 'drivewheel', 'enginelocation', 'wheelbase', 'carlength',
       'carwidth', 'carheight', 'curbweight', 'enginetype', 'cylindernumber',
       'enginesize', 'fuelsystem', 'boreratio', 'stroke', 'compressionratio',
       'horsepower', 'peakrpm', 'citympg', 'highwaympg']]
```

```
y = cars['price']


# CREATING DUMMY VARIABLES FOR ALL CATEGORICAL VARIABLES
cars_categorical = X.select_dtypes(include=['object'])
cars_categorical.head()
```

	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	enginetype	cylindernumber
0	gas	std	two	convertible	rwd	front	dohc	four
1	gas	std	two	convertible	rwd	front	dohc	four
2	gas	std	two	hatchback	rwd	front	ohcv	six
3	gas	std	four	sedan	fwd	front	ohc	four
4	gas	std	four	sedan	4wd	front	ohc	five

```
# CONVERT INTO DUMMIES
cars_dummies = pd.get_dummies(cars_categorical, drop_first=True)
cars_dummies.head()
```

	fueltype_gas	aspiration_turbo	doornumber_two	carbody_hardtop	carbody_hatchback	carbody_sedan	c
0	1	0	1	0	0	0	
1	1	0	1	0	0	0	
2	1	0	1	0	1	0	
3	1	0	0	0	0	1	
4	1	0	0	0	0	1	

5 rows × 29 columns

```
# DROP CATEGORICAL VARIABLES
X = X.drop(list(cars_categorical.columns), axis=1)


# CONCAT DUMMY VARIABLES WITH X
X = pd.concat([X, cars_dummies], axis=1)


# SCALING THE FEATURES
from sklearn.preprocessing import scale
cols = X.columns
X = pd.DataFrame(scale(X))
X.columns = cols
X.columns
```

```
Index(['symboling', 'wheelbase', 'carlength', 'carwidth', 'carheight',
       'curbweight', 'enginesize', 'boreratio', 'stroke', 'compressionratio',
       'horsepower', 'peakrpm', 'citympg', 'highwaympg', 'fueltype_gas',
       'aspiration_turbo', 'doornumber_two', 'carbody_hardtop',
       'carbody_hatchback', 'carbody_sedan', 'carbody_wagon', 'drivewheel_fwd',
       'drivewheel_rwd', 'enginelocation_rear', 'enginetype_dohcv',
       'enginetype_l', 'enginetype_ohc', 'enginetype_ohcf', 'enginetype_ohcv',
       'enginetype_rotor', 'cylindernumber_five', 'cylindernumber_four',
       'cylindernumber_six', 'cylindernumber_three', 'cylindernumber_twelve',
       'cylindernumber_two', 'fuelsystem_2bbl', 'fuelsystem_4bbl',
       'fuelsystem_idi', 'fuelsystem_mfi', 'fuelsystem_mpfi',
       'fuelsystem_spdi', 'fuelsystem_spfi'],
      dtype='object')
```

```
# SPLIT INTO TRAIN AND TEST
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,test_size = 0.3, random_state=100)


#TRAINING THE MODEL


lm = LinearRegression()
```

```
lm.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

+ Code + Text

```
print(lm.coef_)
print(lm.intercept_)
```

```
[ 3.10948137e+02  8.36453448e+02 -3.43260665e+02  1.46810706e+03
  2.72600381e+02  5.87779180e+02  9.22894020e+03 -2.55869078e+03
 -2.47902728e+03 -4.60781831e+02  1.78214309e+03  7.54540184e+02
 -1.24778464e+03  1.64373073e+03 -5.10899847e+02  7.50918876e+02
  1.19818860e+02 -8.10723498e+02 -1.68110837e+03 -1.29400048e+03
 -9.29406051e+02 -3.25156860e+02  3.82124154e+02  7.23233783e+02
 -1.30457362e+02  8.20729163e+01  2.60653437e+03  1.05033915e+03
 -7.91711232e+02  1.23615776e+03  6.02211054e+01  1.51957444e+03
 -4.11635324e+00  1.10565452e+03 -1.63423128e+03  1.23615776e+03
 -9.39263854e+00  5.89877229e+01  5.10899847e+02  2.56734680e-25
 -4.56241487e+01 -6.15650841e+02 -1.27481262e-27]
13614.94107409846
```

```
y_pred = lm.predict(X_test)
```

```
from sklearn.metrics import r2_score

print('The Accuracy is :',r2_score(y_true=y_test, y_pred=y_pred)*100,'%')
```

```
The Accuracy is : 84.29496993045859 %
```