1)import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn import metrics

2) #data collection and processing

#loading the data from csv file to pandas df

car\_dataset=pd.read\_csv('C:\\Users\\Amit\\Desktop\\car.csv')

3) #inspecting the first 5 rows of df

car\_dataset.head()

4) car\_dataset.shape

5) car\_dataset.info()

6) car\_dataset.isnull().sum() #checking the number of misiisng

7) #checking the distrubution of categorical data

print(car\_dataset.Fuel\_Type.value\_counts())

print(car\_dataset.Seller\_Type.value\_counts())

print(car\_dataset.Transmission.value\_counts())

8) #encoding "fuel type column"

car\_dataset.replace({'Fuel\_Type':{'Petrol':0,'Diesel':1,'CNG':2}},inplace=True)

#encoding seller type

car\_dataset.replace({'Seller\_Type':{'Dealer':0,'Individual':1}},inplace=True)

#encoding transmission

car\_dataset.replace({'Transmission\_Type':{'Manual':0,'Automatic':1}},inplace=True)

9) car\_dataset.head()

10) #spliting the data into traning and test data

X= car\_dataset.drop(['Car\_Name','Selling\_Price'],axis=1)

Y=car\_dataset['Selling\_Price']

11) print(X)

12) print(Y)

13) #spliting the data and target

X\_train,X\_test,Y\_train,Y\_test=train\_test\_split(X,Y,test\_size=0.1,random\_state=2)

14) #model training

#loading linear regression model

lin\_reg\_model=LinearRegression()

15) from sklearn.preprocessing import LabelEncoder

# Example: Encoding a categorical column 'transmission'

label\_encoder = LabelEncoder()

X\_train['Transmission'] = label\_encoder.fit\_transform(X\_train['Transmission'])

# Then fit the model

lin\_reg\_model.fit(X\_train, Y\_train)

16) #predicition on training model

training\_data\_prediciton=lin\_reg\_model.predict(X\_train)

17) #compare the original value

#calculate r squared erroe

error\_score=metrics.r2\_score(Y\_train,training\_data\_prediciton)

print("R squared error",error\_score)

18) #visualize the actual price and predicted data

plt.scatter(Y\_train,training\_data\_prediciton)

plt.xlabel("actual price")

plt.ylabel("predicted price")

plt.title("actual price vs predicted price")

plt.show()