In [339]: import pandas as pd import numpy as np import matplotlib as plt In [340]: car=pd.read_csv('carresalevaluep.csv') In [341]: car.head() Unnamed: 0 Price kms_driven fuel_type Out[341]: Hyundai Santro Xing 40 Mahindra Jeep CL550 Mahindra 2006 425000 Diesel 2 2 Hyundai Grand (10) Hyundai 2014 325000 28000 Petrol 3 Ford EcoSport Titanium Ford 2014 575000 Ford Figo Ford 2012 175003 41000 In [342]: car.shape Out[342]: (816, 7)In [343]: car.info() <class 'pandas.core.frame.DataFrame'> RangeIndex: 816 entries, 0 to 815 Data columns (total 7 columns): # Column Non-Null Count Dtype 0 Unnamed: 0 816 non-null int64 1 name 816 non-null object 2 company 816 non-null object 3 year 816 non-null int64 4 Price 816 non-null int64 5 kms_driven 816 non-null int64 6 fuel_type 816 non-null object dtypes: int64(4), object(3) memory usage: 44.8+ KB In [344]: Price kms_driven fuel_type Out[344]: 45000 Hyundai Santro Xing Hyundai 2007 Petrol 1 Mahindra Jeep CL550 Mahindra 2006 Diesel 2 Hyundai Grand i10 Hyundai 2014 325000 29000 Petrol 3 3 Ford EcoSport Titanium Ford 2014 575000 36000 Diesel Ford Figo Ford 2012 175000 811 811 Maruti Suzuki Ritz Maruti 2011 270000 50000 Petrol Tata Indica V2 Tata 2009 110000 30000 Diesel 813 813 Toyota Corolla Altis Toyota 2009 300000 132000 Petrol Tata Zest XM Tata 2018 260000 27000 814 814 Diesel Mahindra Quanto CB Mahindra 2013 390000 815 815 40000 Diesal 816 rows × 7 columns In [345]: car[fuel_type].unique() array(['Petrol', 'Diesel', 'LPG'], dtype=object) Out[345]: In [346]: car[year].unique() array([2007, 2006, 2014, 2012, 2013, 2016, 2015, 2010, 2017, 2008, 2018, Out[346]: 2011, 2019, 2009, 2005, 2000, 2003, 2004, 1995, 2002, 2001], dtype=int64) In [347]: backup*car.copy()

In [346]:	car['y	/ear'].uniqu	ue()										
Out[346]:	array([2007, 2006, 2014, 2012, 2013, 2016, 2015, 2010, 2017, 2008, 2018, 2011, 2019, 2009, 2005, 2000, 2003, 2004, 1995, 2002, 2001], dtype=int64)												
In [347]:	backup=car.copy()												
In [348]:	car.h	ead()											
Out[348]:	Unn	named: 0	1	name	company	year	Price k	cms_driven f	uel_type				
out[540].	0	0	Hyundai Santro	Xing	Hyundai	2007	80000	45000	Petrol				
	1	1 1	Mahindra Jeep C	L550 I	Mahindra	2006	425000	40	Diesel				
	2	2	Hyundai Gran	d i10	Hyundai	2014	325000	28000	Petrol				
	3		ord EcoSport Tita		Ford		575000	36000	Diesel				
	4	4	Ford	Figo	Ford	2012	175000	41000	Diesel				
In [349];	car.reset_index(drop=True)												
Out[349]:	U	Innamed: 0		name	compan	y yea	r Price	kms_driven	fuel_type				
	0	0	Hyundai Sani	4.00.0		200	7 80000	45000	Petrol				
	1	1	Mahindra Jeep					40	Diesel				
	2	2	Hyundai Gr					28000	Petrol				
	3		Ford EcoSport T				4 575000	36000	Diesel				
	4	4	Ec	ord Figo	Fon	1 201	2 175000	41000	Diesel				
	811	811	Maruti Suz					50000	Petrol				
	812 813	812 813	Toyota Coro	dica V2 II a Altie				30000 132000	Diesel				
	814	814	1777370000000	Zest XM				27000	Diesel				
	815	815	Mahindra Qu					40000	Diesel				
	816 r	ows × /	columns										
In [350]:	car.d	escribe()											
In [350]:	car.d	escribe()											
In [350]: Out[350]:	1000000	Unnamed: 0			Price		s_driven						
	count	Unnamed: 0	816.000000		000e+02	816	.000000						
	count	Unnamed: 0 816.000000 407.500000	816.000000 2012.444853	4,117	000e+02 176e+05	816 46275	.000000						
	count mean std	Unnamed: 0 816.000000 407.500000 235.703203	816.000000 2012.444853 4.002992	4.1171 4.7518	000e+02 176e+05 844e+05	816 46275 34297	.000000 .531863 .428044						
	count mean std min	Unnamed: 0 816.000000 407.500000 235.703203 0.000000	816.000000 2012.444853 4.002992 1995.000000	4.1171 4.7518 3.0000	000e+02 176e+05 844e+05 000e+04	816 46275 34297	.000000 .531863 .428044 .000000						
	count mean std min 25%	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 203.750000	816.000000 2012.444853 4.002992 1995.000000 2010.000000	4.7518 3.0000 1.7500	000e+02 176e+05 844e+05 000e+04	816 46275 34297 0 27000	.000000 .531863 .428044 .000000 .000000						
	count mean std min 25%	Unnamed: 0 816.000000 407.500000 235.703203 0.000000	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2013.000000	4.1171 4.7518 3.0000 1.7500 2.9999	000e+02 176e+05 844e+05 000e+04 000e+05	816 46275 34297 0 27000 41000	.000000 .531863 .428044 .000000						
	count mean std min 25% 50%	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 203.750000 407.500000 611.250000	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2013.000000	4.117 4.7518 3.0000 1.7500 2.9999 4.9125	000e+02 176e+05 844e+05 000e+04 000e+05 990e+05	816 46275 34297 0 27000 41000 56818	.000000 .531863 .428044 .000000 .000000 .000000						
	count mean std min 25% 50% 75%	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 203.750000 407.500000 611.250000	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2013.000000 2015.000000	4.117 4.7518 3.0000 1.7500 2.9999 4.9125	000e+02 176e+05 844e+05 000e+04 000e+05 990e+05	816 46275 34297 0 27000 41000 56818	.000000 .531863 .428044 .000000 .000000 .000000						
	count mean std min 25% 50% 75% max	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 203.7500000 407.500000 815.000000	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2013.000000 2015.000000	4.117 4.7518 3.0000 1.7500 2.9999 4.9125 8.5000	000e+02 176e+05 844e+05 000e+04 000e+05 990e+05	816 46275 34297 0 27000 41000 56818	.000000 .531863 .428044 .000000 .000000 .000000						
Out[350]:	count mean std min 25% 50% 75% max	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 203.750000 407.500000 611.250000 815.000000	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2013.000000 2015.000000 2019.000000	4.117 4.7518 3.0000 1.7500 2.9999 4.9125 8.5000	000e+02 176e+05 844e+05 000e+04 000e+05 990e+05	816 46275 34297 0 27000 41000 56818	.000000 .531863 .428044 .000000 .000000 .000000						
Out[350]: In [351]:	count mean std min 25% 50% 75% max x=ca y=ca	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 203.750000 407.500000 815.000000 r.drop(colu	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2013.000000 2015.000000 2019.000000	4.1171 4.7518 3.0000 1.7500 2.9999 4.9125 8.5000	000e+02 176e+05 844e+05 000e+04 000e+05 990e+05	816 46275 34297 0 27000 41000 56818	.000000 .531863 .428044 .000000 .000000 .000000						
Out[350]:	count mean std min 25% 50% 75% max x=ca y=ca	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 203.750000 407.500000 815.000000 r.drop(colur['Price']	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2013.000000 2015.000000 2019.000000	4.1171 4.7518 3.0000 1.7500 2.9999 4.9125 8.5000	000e+02 176e+05 844e+05 000e+04 000e+05 990e+05	816 46275 34297 0 27000 41000 56818	.000000 .531863 .428044 .000000 .000000 .000000						
Out[350]: In [351]:	count mean std min 25% 50% 75% max x=ca y=ca	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 203.750000 407.500000 815.000000 r.drop(colu	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2013.000000 2015.000000 2019.000000	4.1171 4.7518 3.0000 1.7500 2.9999 4.9125 8.5000	000e+02 176e+05 844e+05 000e+04 000e+05 990e+05	816 46275 34297 0 27000 41000 56818	.000000 .531863 .428044 .000000 .000000 .000000						
Out[350]: In [351]: In [352]:	count mean std min 25% 50% 75% max x=cal y=cal	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 407.500000 611.2500000 815.000000 r.drop(colur['Price']	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2015.000000 2019.000000 umns='Price')	4.117 4.7518 3.0000 1.7500 2.9999 4.912 8.5000	000e+02 176e+05 344e+05 000e+04 000e+05 990e+05 500e+05 003e+06	816 46275 34297 0 27000 41000 56818 400000	.000000 .531863 .428044 .000000 .000000 .000000 .500000						
Out[350]: In [351]:	count mean std min 25% 50% 75% max x=ca y=cal	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 407.500000 611.2500000 815.0000000 r.drop(colur['Price'] r.drop(colur['Price']	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2015.000000 2015.000000 2019.000000 umns='Price')	4.117' 4.7518 3.0000 1.7500 2.9999 4.912' 8.5000	000e+02 176e+05 344e+05 000e+04 000e+05 990e+05 500e+05 103e+06	816 46275 34297 0 27000 41000 56818 400000	.000000 .531863 .428044 .000000 .000000 .000000 .500000	0.2)					
Out[350]: In [351]: In [352]:	count mean std min 25% 50% 75% max x=ca y=cal	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 407.500000 611.2500000 815.0000000 r.drop(colur['Price'] r.drop(colur['Price']	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2015.000000 2019.000000 umns='Price')	4.117' 4.7518 3.0000 1.7500 2.9999 4.912' 8.5000	000e+02 176e+05 344e+05 000e+04 000e+05 990e+05 500e+05 103e+06	816 46275 34297 0 27000 41000 56818 400000	.000000 .531863 .428044 .000000 .000000 .000000 .500000	0.2)					
Out[350]: In [351]: In [352]:	count mean std min 25% 50% 75% max x=ca y=ca from x_tra	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 203.7500000 407.500000 815.0000000 r.drop(colur['Price'] r.drop(colur['Price'] sklearn.min,x_test,y_	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2015.000000 2019.000000 umns='Price') umns='Price')	4.117** 4.7518 3.0000 1.7500 2.9999 4.912** 8.5000	000e+02 176e+05 844e+05 000e+04 000e+05 990e+05 000e+05 003e+06	816 46275 34297 0 27000 41000 56818 4400000	.000000 .531863 .428044 .000000 .000000 .000000 .500000 .000000	0.2)					
Out[350]: In [351]: In [352]:	count mean std min 25% 50% 75% max x=ca y=ca y=ca from x_tra	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 203.7500000 407.500000 815.0000000 r.drop(colur[Price]) sklearn.min,x_test,y_	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2015.000000 2015.000000 2019.000000 umns='Price')	4.1171 4.7518 3.0000 1.7500 2.9999 4.9121 8.5000	000e+02 176e+05 844e+05 000e+04 1000e+05 090e+05 000e+05 003e+06	816 46275 34297 0 27000 41000 56818 4400000	.000000 .531863 .428044 .000000 .000000 .000000 .500000 .000000	0.2)					
Out[350]: In [351]: In [352]:	count mean std min 25% 50% 75% max x=ca y=cal from from from from the count mean std min 25% 50% 75% max x=ca y=cal from from from from from from from from	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 203.7500000 611.2500000 815.0000000 r.drop(colur['Price'] r.drop(colur['Price'] sklearn.min,x_test,y_ sklearn.min,x_klear	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2015.000000 2019.000000 umns='Price') umns='Price') umns=inetrics import reprocessing	4.117' 4.7518 3.0000 1.7500 2.9999 4.912' 8.5000 n import ir2_sc. import	000e+02 176e+05 844e+05 000e+04 100e+05 90e+05 500e+05 003e+06 ort train_ iest_split	816 46275 34297 0 27000 41000 56818 400000 ttest_s (x,y, te	.000000 .531863 .428044 .000000 .000000 .000000 .500000 .000000						
Out[350]: In [351]: In [352]:	count mean std min 25% 50% 75% max x=ca y=cal from x_tra from from from from from from from from	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 407.500000 611.2500000 815.000000 r.drop(colur['Price'] sklearn.m in,x_test,y_ sklearn.m sklearn.m	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2015.000000 2019.000000 umns='Price') umns='Price') codel_selectiontrain, y_test=	4.117' 4.7518 3.0000 1.7500 2.9999 4.912' 8.5000 import ir2_sciimport make	000e+02 176e+05 344e+05 000e+04 000e+05 990e+05 500e+05 003e+06 ort train_ lest_split	816 46275 34297 0 27000 41000 56818 4000000 ttest_s (x,y, te	.000000 .531863 .428044 .000000 .000000 .000000 .500000 .000000						
Out[350]: In [351]: In [352]:	count mean std min 25% 50% 75% max x=ca y=cal from x_tra from from from from from from from from	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 407.500000 611.2500000 815.000000 r.drop(colur['Price'] sklearn.m in,x_test,y_ sklearn.m sklearn.m	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2013.000000 2015.000000 2019.000000 umns='Price') umns='Price') iodel_selection train, y_test= near_model in tetrics import reprocessing ompose impo	4.117' 4.7518 3.0000 1.7500 2.9999 4.912' 8.5000 import ir2_sciimport make	000e+02 176e+05 344e+05 000e+04 000e+05 990e+05 500e+05 003e+06 ort train_ lest_split	816 46275 34297 0 27000 41000 56818 4000000 ttest_s (x,y, te	.000000 .531863 .428044 .000000 .000000 .000000 .500000 .000000						
Out[350]: In [351]: In [352]:	count mean std min 25% 50% 75% max x=ca y=cal from x_tra from from from from from from from from	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 407.500000 611.250000 815.000000 r.drop(colur['Price'] sklearn.min,x_test,y_ sklearn.pri sklearn.pri sklearn.pri	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2015.000000 2019.000000 umns='Price') umns='Price') codel_selection train, y_test= mear_model in tetrics import teprocessing	4.117' 4.7518 3.0000 1.7500 2.9999 4.912' 8.5000 import ir2_sciimport make	000e+02 176e+05 344e+05 000e+04 000e+05 990e+05 500e+05 003e+06 ort train_ lest_split	816 46275 34297 0 27000 41000 56818 4000000 ttest_s (x,y, te	.000000 .531863 .428044 .000000 .000000 .000000 .500000 .000000						
Out[350]: In [351]: In [352]: In [353]:	count mean std min 25% 50% 75% max x=ca y=cal from x_trainfrom from from from ohe =	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 203.7500000 611.250000 815.0000000 r.drop(colur['Price'] sklearn.min,x_test,y_ sklearn.pr sklearn.pr sklearn.pr	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2015.000000 2019.000000 umns='Price') umns='Price') codel_selection train, y_test= mear_model in tetrics import teprocessing	4.117' 4.7518 3.0000 1.7500 2.9999 4.912' 8.5000 nimport limport rain_1 r2_sc	000e+02 176e+05 344e+05 000e+04 000e+05 000e+05 000e+05 000e+06 000e+06 000e+06	816 46275 34297 0 27000 41000 56818 4000000 ttest_s (x,y, te	.000000 .531863 .428044 .000000 .000000 .000000 .500000 .000000						
Out[350]: In [351]: In [352]: In [353]:	count mean std min 25% 50% 75% max x=ca y=cal from x_tra from from from from from from from from	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 203.7500000 407.500000 815.000000 r.drop(colur[Price]) sklearn.min,x_test,y_ sklearn.min,x_test,y_ sklearn.pi = OneHotE fit(x[['name	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2013.000000 2015.000000 2019.000000 umns='Price') umns='Price') codel_selection train, y_test= near_model in tetrics import treprocessing trep	4.117' 4.7518 3.0000 1.7500 2.9999 4.912' 8.5000 nimport limport rain_1 r2_sc	000e+02 176e+05 344e+05 000e+04 000e+05 000e+05 000e+05 000e+06 000e+06 000e+06	816 46275 34297 0 27000 41000 56818 4000000 ttest_s (x,y, te	.000000 .531863 .428044 .000000 .000000 .000000 .500000 .000000						
Out[350]: In [351]: In [352]: In [353]:	count mean std min 25% 50% 75% max x=ca y=cal from x_tra from from from from from from from from	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 203.7500000 611.250000 815.0000000 r.drop(colur['Price'] sklearn.min,x_test,y_ sklearn.pr sklearn.pr sklearn.pr	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2013.000000 2015.000000 2019.000000 umns='Price') umns='Price') codel_selection train, y_test= near_model in tetrics import treprocessing trep	4.117' 4.7518 3.0000 1.7500 2.9999 4.912' 8.5000 nimport limport rain_1 r2_sc	000e+02 176e+05 844e+05 000e+04 000e+05 000e+05 000e+05 000e+06 000e+06 000e+06	816 46275 34297 0 27000 41000 56818 4000000 ttest_s (x,y, te	.000000 .531863 .428044 .000000 .000000 .000000 .500000 .000000						
Out[350]: In [351]: In [352]: In [354]: In [355]:	count mean std min 25% 50% 75% max x=ca y=cal from x_tra from from from from from from from from	Unnamed: 0 816.000000 407.500000 235.703203 0.000000 203.7500000 407.500000 815.000000 r.drop(colur[Price]) sklearn.min,x_test,y_ sklearn.min,x_test,y_ sklearn.pi = OneHotE fit(x[['name	816.000000 2012.444853 4.002992 1995.000000 2010.000000 2015.000000 2019.000000 2019.000000 umns='Price') umns='Price') todel_selectio_train, y_test= near_model interics import reprocessing oppose imporpose imporpose import incoder() e','company','funcoder() e','company','funcoder()	4.117' 4.7518 3.0000 1.7500 2.9999 4.912' 8.5000 nimport limport rain_1 r2_sc	000e+02 176e+05 844e+05 000e+04 000e+05 000e+05 000e+05 000e+06 000e+06 000e+06	816 46275 34297 0 27000 41000 56818 4000000 ttest_s (x,y, te	.000000 .531863 .428044 .000000 .000000 .000000 .500000 .000000						

Out [356]: [array(['Audi A3 Cabriolet', 'Audi A4 1.8', 'Audi A4 2.0', 'Audi A6 2.0', 'Audi A8', 'Audi Q3 2.0', 'Audi Q5 2.0', 'Audi Q7', 'BMW 3 Series', 'BMW 5 Series', 'BMW 7 Series', 'BMW X1', 'BMW X1 sDrive20d', 'BMW X1 xDrive20d', 'Chevrolet Beat', 'Chevrolet Beat Diesel', 'Chevrolet Beat LS', 'Chevrolet Beat LT', 'Chevrolet Beat PS', 'Chevrolet Cruze LTZ', 'Chevrolet Enjoy', 'Chevrolet Enjoy 1.4', 'Chevrolet Sail 1.2', 'Chevrolet Sail UVA', 'Chevrolet Spark', 'Chevrolet Spark 1.0', 'Chevrolet Spark LS', 'Chevrolet Spark LT', 'Chevrolet Tavera LS', 'Chevrolet Tavera Neo', 'Datsun GO T', 'Datsun Go Plus', 'Datsun Redi GO', 'Fiat Linea Emotion', 'Fiat Petra ELX', 'Fiat Punto Emotion', 'Force Motors Force', 'Force Motors One', 'Ford EcoSport', 'Ford EcoSport Ambiente', 'Ford EcoSport Titanium', 'Ford EcoSport Trend', 'Ford Endeavor 4x4', 'Ford Fiesta', 'Ford Fiesta SXI', 'Ford Figo', 'Ford Figo Diesel', 'Ford Figo Duratorq', 'Ford Figo Petrol', 'Ford Fusion 1.4', 'Ford Ikon 1.3', 'Ford Ikon 1.6', 'Hindustan Motors Ambassador', 'Honda Accord', 'Honda Amaze', 'Honda Amaze 1.2', 'Honda Amaze 1.5', 'Honda Brio', 'Honda Brio V', 'Honda Brio VX', 'Honda City', 'Honda City 1.5', 'Honda City SV', 'Honda City VX', 'Honda City ZX', 'Honda Jazz S', 'Honda Jazz VX', 'Honda Mobilio', 'Honda Mobilio S', 'Honda WR V', 'Hyundai Accent', 'Hyundai Accent Executive', 'Hyundai Accent GLE', 'Hyundai Accent GLX', 'Hyundai Creta', 'Hyundai Creta 1.6', 'Hyundai Elantra 1.8', 'Hyundai Elantra SX', 'Hyundai Elite i20', 'Hyundai Eon', 'Hyundai Eon D', 'Hyundai Eon Era', 'Hyundai Eon Magna', 'Hyundai Eon Sportz', 'Hyundai Fluidic Verna', 'Hyundai Getz', 'Hyundai Getz GLE', 'Hyundai Getz Prime', 'Hyundai Grand i10', 'Hyundai Santro', 'Hyundai Santro AE', 'Hyundai Santro Xing', 'Hyundai Sonata Transform', 'Hyundai Verna', 'Hyundai Verna 1.4', 'Hyundai Verna 1.6', 'Hyundai Verna Fluidic', 'Hyundai Verna Transform', 'Hyundai Verna VGT', 'Hyundai Xcent Base', 'Hyundai Xcent SX', 'Hyundai i10', 'Hyundai i10 Era', 'Hyundai i10 Magna', 'Hyundai i10 Sportz', 'Hyundai i20', 'Hyundai i20 Active', 'Hyundai i20 Asta', 'Hyundai i20 Magna', 'Hyundai i20 Select', 'Hyundai i20 Sportz', 'Jaguar XE XE', 'Jaguar XF 2.2', 'Jeep Wrangler Unlimited', 'Land Rover Freelander', 'Mahindra Bolero DI', 'Mahindra Bolero Power', 'Mahindra Bolero SLE', 'Mahindra Jeep CL550', 'Mahindra Jeep MM', 'Mahindra KUV100', 'Mahindra KUV100 K8', 'Mahindra Logan', 'Mahindra Logan Diesel', 'Mahindra Quanto C4', 'Mahindra Quanto C8', 'Mahindra Scorpio', 'Mahindra Scorpio 2.6', 'Mahindra Scorpio LX', 'Mahindra Scorpio S10', 'Mahindra Scorpio S4', 'Mahindra Scorpio SLE', 'Mahindra Scorpio SLX', 'Mahindra Scorpio VLX', 'Mahindra Scorpio Vlx', 'Mahindra Scorpio W', 'Mahindra TUV300 T4', 'Mahindra TUV300 T8', 'Mahindra Thar CRDe', 'Mahindra XUV500', 'Mahindra XUV500 W10', 'Mahindra XUV500 W6', 'Mahindra XUV500 W8', 'Mahindra Xylo D2', 'Mahindra Xylo E4', 'Mahindra Xylo E8', 'Maruti Suzuki 800', 'Maruti Suzuki A', 'Maruti Suzuki Alto', 'Maruti Suzuki Baleno', 'Maruti Suzuki Celerio', 'Maruti Suzuki Ciaz', 'Maruti Suzuki Dzire', 'Maruti Suzuki Eeco', 'Maruti Suzuki Ertiga', 'Maruti Suzuki Esteem', 'Maruti Suzuki Estilo', 'Maruti Suzuki Maruti', 'Maruti Suzuki Omni', 'Maruti Suzuki Ritz', 'Maruti Suzuki S', 'Maruti Suzuki SX4', 'Maruti Suzuki Stingray', 'Maruti Suzuki Swift', 'Maruti Suzuki Versa', 'Maruti Suzuki Vitara', 'Maruti Suzuki Wagon', 'Maruti Suzuki Zen', 'Mercedes Benz A', 'Mercedes Benz B', 'Mercedes Benz C', 'Mercedes Benz GLA', 'Mini Cooper S', 'Mitsubishi Lancer 1.8', 'Mitsubishi Pajero Sport', 'Nissan Micra XL', 'Nissan Micra XV', 'Nissan Sunny', 'Nissan Sunny XL', 'Nissan Terrano XL', 'Nissan X Trail', 'Renault Duster', 'Renault Duster 110', 'Renault Duster 110PS', 'Renault Duster 85', 'Renault Duster 85PS', 'Renault Duster RxL', 'Renault Kwid', 'Renault Kwid 1.0', 'Renault Kwid RXT', 'Renault Lodgy 85', 'Renault Scala RxL', 'Skoda Fabia', 'Skoda Fabia 1.2L', 'Skoda Fabia Classic', 'Skoda Laura', 'Skoda Octavia Classic', 'Skoda Rapid Elegance', 'Skoda Superb 1.8', 'Skoda Yeti Ambition', 'Tata Aria Pleasure', 'Tata Bolt XM', 'Tata Indica', 'Tata Indica V2', 'Tata Indica eV2', 'Tata Indigo CS', 'Tata Indigo LS', 'Tata Indigo LX', 'Tata Indigo Marina', 'Tata Indigo eCS', 'Tata Manza',

'Tata Manza Aqua', 'Tata Manza Aura', 'Tata Manza ELAN',

```
'Mercedes Benz A', 'Mercedes Benz B', 'Mercedes Benz C',
                        'Mercedes Benz GLA', 'Mini Cooper S', 'Mitsubishi Lancer 1.8',
                        'Mitsubishi Pajero Sport', 'Nissan Micra XL', 'Nissan Micra XV',
                        'Nissan Sunny', 'Nissan Sunny XL', 'Nissan Terrano XL',
                        'Nissan X Trail', 'Renault Duster', 'Renault Duster 110',
                        'Renault Duster 110PS', 'Renault Duster 85', 'Renault Duster 85PS',
                        'Renault Duster RxL', 'Renault Kwid', 'Renault Kwid 1.0',
                        'Renault Kwid RXT', 'Renault Lodgy 85', 'Renault Scala RxL',
                        'Skoda Fabia', 'Skoda Fabia 1.2L', 'Skoda Fabia Classic',
                        'Skoda Laura', 'Skoda Octavia Classic', 'Skoda Rapid Elegance',
                        'Skoda Superb 1.8', 'Skoda Yeti Ambition', 'Tata Aria Pleasure',
                        'Tata Bolt XM', 'Tata Indica', 'Tata Indica V2', 'Tata Indica eV2',
                        'Tata Indigo CS', 'Tata Indigo LS', 'Tata Indigo LX',
                        'Tata Indigo Marina', 'Tata Indigo eCS', 'Tata Manza',
                        Tata Manza Aqua', 'Tata Manza Aura', 'Tata Manza ELAN',
                        Tata Nano', Tata Nano Cx', Tata Nano GenX', Tata Nano LX',
                        'Tata Nano Lx', 'Tata Sumo Gold', 'Tata Sumo Grande',
                        'Tata Sumo Victa', 'Tata Tiago Revotorq', 'Tata Tiago Revotron',
                        'Tata Tigor Revotron', 'Tata Venture EX', 'Tata Vista Quadrajet',
                        'Tata Zest Quadrajet', 'Tata Zest XE', 'Tata Zest XM',
                        'Toyota Corolla', 'Toyota Corolla Altis', 'Toyota Corolla H2',
                        'Toyota Etios', 'Toyota Etios G', 'Toyota Etios GD',
                        'Toyota Etios Liva', 'Toyota Fortuner', 'Toyota Fortuner 3.0',
                        Toyota Innova 2.0', Toyota Innova 2.5', Toyota Qualis',
                        'Volkswagen Jetta Comfortline', 'Volkswagen Jetta Highline',
                        'Volkswagen Passat Diesel', 'Volkswagen Polo',
                        'Volkswagen Polo Comfortline', 'Volkswagen Polo Highline',
                        'Volkswagen Polo Highline1.2L', 'Volkswagen Polo Trendline',
                        'Volkswagen Vento Comfortline', 'Volkswagen Vento Highline',
                        'Volkswagen Vento Konekt', 'Volvo S80 Summum'], dtype=object),
                    array(['Audi', 'BMW', 'Chevrolet', 'Datsun', 'Fiat', 'Force', 'Ford',
                         'Hindustan', 'Honda', 'Hyundai', 'Jaguar', 'Jeep', 'Land',
                        'Mahindra', 'Maruti', 'Mercedes', 'Mini', 'Mitsubishi', 'Nissan',
                        'Renault', 'Skoda', 'Tata', 'Toyota', 'Volkswagen', 'Volvo'],
                        dtype=object).
                    array(['Diesel', 'LPG', 'Petrol'], dtype=object)]
   In [357]:
                   column_trans=make_column_transformer((OneHotEncoder(categories=ohe.categories_)_fname;company:fuel_type]),re
Ir=LinearRegression()
   In [358]:
                   Ir=LinearRegression()
   In [359]:
                   pipe=make_pipeline(column_trans.lr)
   In [360]:
                   pipe.fit(x_train,y_train)
                   Pipeline(steps=[('columntransformer',
   Out[360]:
                             ColumnTransformer(remainder='passthrough',
                                        transformers=[('onehotencoder',
                                                 OneHotEncoder(categories=[array(['Audi A3 Cabriolet', 'Audi A4 1.8', 'Audi
                    A4 2.0', 'Audi A6 2.0',
                        'Audi A8', 'Audi Q3 2.0', 'Audi Q5 2.0', 'Audi Q7', 'BMW 3 Series',
                        'BMW 5 Series', 'BMW 7 Series', 'BMW X1', 'BMW X1 sDrive20d',
                        'BMW X1 xDrive20d', 'Chevrolet Beat', 'Chevrolet Beat...
                                                                array([Audi', 'BMW', 'Chevrolet', 'Datsun', 'Fiat', 'Force', 'For
                    ď,
                        'Hindustan', 'Honda', 'Hyundai', 'Jaguar', 'Jeep', 'Land',
                        'Mahindra', 'Maruti', 'Mercedes', 'Mini', 'Mitsubishi', 'Nissan',
                        'Renault', 'Skoda', 'Tata', 'Toyota', 'Volkswagen', 'Volvo'],
                       dtype=object),
                                                                array(['Diesel', 'LPG', 'Petrol'], dtype=object)]),
                                                 'name', 'company',
                                                  'fuel_type'])])),
                             ('linearregression', LinearRegression())])
   In [361]:
                   y_pred =pipe.predict(x_test)
   In [362]:
                   r2_score(y_test,y_pred)
   Out[362]:
                   0.4977698265719559
```

'Maruti Suzuki Vitara', 'Maruti Suzuki Wagon', 'Maruti Suzuki Zen',

```
In [363]: scores=[
                for i in range(1000):
                  x_train,x_test,y_train, y_test=train_test_split(x,y, test_size=0.2)
                  Ir=LinearRegression()
                  pipe=make_pipeline(column_trans,lr)
                  pipe.fit(x_train,y_train)
                  y_pred =pipe.predict(x_test)
                  scores.append(r2_score(y_test,y_pred))
In [364]:
                np.argmax(scores)
Out[364]:
In [365]:
                scores[np.argmax(scores)]
                 0.7973804780567894
Out[365]:
In [366]:
                x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2, random_state=np.argmax(scores))
                Ir=LinearRegression()
                pipe=make_pipeline(column_trans,lr)
                pipe.fit(x_train,y_train)
                 y_pred =pipe.predict(x_test)
                r2_score(y_test,y_pred)
                0.7066410845475624
Out[366]:
In [378]:
                pipe.steps[0][1].transformers[0][1].categories[0]
                 array(['Audi A3 Cabriolet', 'Audi A4 1.8', 'Audi A4 2.0', 'Audi A6 2.0',
Out[378]:
                     'Audi A8', 'Audi Q3 2.0', 'Audi Q5 2.0', 'Audi Q7', 'BMW 3 Series',
                     'BMW 5 Series', 'BMW 7 Series', 'BMW X1', 'BMW X1 sDrive20d',
                     'BMW X1 xDrive20d', 'Chevrolet Beat', 'Chevrolet Beat Diesel',
                     'Chevrolet Beat LS', 'Chevrolet Beat LT', 'Chevrolet Beat PS', 'Chevrolet Cruze LTZ', 'Chevrolet Enjoy', 'Chevrolet Enjoy 1.4',
                     'Chevrolet Sail 1.2', 'Chevrolet Sail UVA', 'Chevrolet Spark',
                     'Chevrolet Spark 1.0', 'Chevrolet Spark LS', 'Chevrolet Spark LT',
                     'Chevrolet Tavera LS', 'Chevrolet Tavera Neo', 'Datsun GO T',
                     'Datsun Go Plus', 'Datsun Redi GO', 'Fiat Linea Emotion',
                     'Fiat Petra ELX', 'Fiat Punto Emotion', 'Force Motors Force',
                     'Force Motors One', 'Ford EcoSport', 'Ford EcoSport Ambiente',
                     'Ford EcoSport Titanium', 'Ford EcoSport Trend',
                     'Ford Endeavor 4x4', 'Ford Fiesta', 'Ford Fiesta SXI', 'Ford Figo',
                     'Ford Figo Diesel', 'Ford Figo Duratorq', 'Ford Figo Petrol',
                     'Ford Fusion 1.4', 'Ford Ikon 1.3', 'Ford Ikon 1.6',
                     'Hindustan Motors Ambassador', 'Honda Accord', 'Honda Amaze',
                     'Honda Amaze 1.2', 'Honda Amaze 1.5', 'Honda Brio', 'Honda Brio V',
                     'Honda Brio VX', 'Honda City', 'Honda City 1.5', 'Honda City SV',
                     'Honda City VX', 'Honda City ZX', 'Honda Jazz S', 'Honda Jazz VX',
                     'Honda Mobilio', 'Honda Mobilio S', 'Honda WR V', 'Hyundai Accent',
                     'Hyundai Accent Executive', 'Hyundai Accent GLE',
                     'Hyundai Accent GLX', 'Hyundai Creta', 'Hyundai Creta 1.6',
                     'Hyundai Elantra 1.8', 'Hyundai Elantra SX', 'Hyundai Elite i20',
                     'Hyundai Eon', 'Hyundai Eon D', 'Hyundai Eon Era',
                     'Hyundai Eon Magna', 'Hyundai Eon Sportz', 'Hyundai Fluidic Verna',
                     'Hyundai Getz', 'Hyundai Getz GLE', 'Hyundai Getz Prime',
                     'Hyundai Grand i10', 'Hyundai Santro', 'Hyundai Santro AE',
                     'Hyundai Santro Xing', 'Hyundai Sonata Transform', 'Hyundai Verna',
                     'Hyundai Verna 1.4', 'Hyundai Verna 1.6', 'Hyundai Verna Fluidic',
                     'Hyundai Verna Transform', 'Hyundai Verna VGT',
                     'Hyundai Xcent Base', 'Hyundai Xcent SX', 'Hyundai i10',
                     'Hyundai i10 Era', 'Hyundai i10 Magna', 'Hyundai i10 Sportz',
                     'Hyundai i20', 'Hyundai i20 Active', 'Hyundai i20 Asta',
                     'Hyundai i20 Magna', 'Hyundai i20 Select', 'Hyundai i20 Sportz',
                     'Jaguar XE XE', 'Jaguar XF 2.2', 'Jeep Wrangler Unlimited',
                     'Land Rover Freelander', 'Mahindra Bolero DI',
                     'Mahindra Bolero Power', 'Mahindra Bolero SLE',
                     'Mahindra Jeep CL550', 'Mahindra Jeep MM', 'Mahindra KUV100',
                     'Mahindra KUV100 K8', 'Mahindra Logan', 'Mahindra Logan Diesel',
                     'Mahindra Quanto C4', 'Mahindra Quanto C8', 'Mahindra Scorpio',
                     'Mahindra Scorpio 2.6', 'Mahindra Scorpio LX',
                     'Mahindra Scorpio S10', 'Mahindra Scorpio S4',
                     'Mahindra Scorpio SLE', 'Mahindra Scorpio SLX',
                     'Mahindra Scorpio VLX', 'Mahindra Scorpio Vlx',
                     'Mahindra Scorpio W', 'Mahindra TUV300 T4', 'Mahindra TUV300 T8',
                     'Mahindra Thar CRDe', 'Mahindra XUV500', 'Mahindra XUV500 W10',
                     'Mahindra XUV500 W6', 'Mahindra XUV500 W8', 'Mahindra Xylo D2',
                     'Mahindra Xylo E4', 'Mahindra Xylo E8', 'Maruti Suzuki 800',
                     'Maruti Suzuki A', 'Maruti Suzuki Alto', 'Maruti Suzuki Baleno',
                     'Maruti Suzuki Celerio', 'Maruti Suzuki Ciaz',
                     'Maruti Suzuki Dzire', 'Maruti Suzuki Eeco',
```

```
Hyundai Grand ITU, Hyundai Santro, Hyundai Santro AE,
'Hyundai Santro Xing', 'Hyundai Sonata Transform', 'Hyundai Verna',
'Hyundai Verna 1.4', 'Hyundai Verna 1.6', 'Hyundai Verna Fluidic',
'Hyundai Verna Transform', 'Hyundai Verna VGT',
'Hyundai Xcent Base', 'Hyundai Xcent SX', 'Hyundai i10',
'Hyundai i10 Era', 'Hyundai i10 Magna', 'Hyundai i10 Sportz',
'Hyundai i20', 'Hyundai i20 Active', 'Hyundai i20 Asta',
'Hyundai i20 Magna', 'Hyundai i20 Select', 'Hyundai i20 Sportz',
'Jaguar XE XE', 'Jaguar XF 2.2', 'Jeep Wrangler Unlimited',
'Land Rover Freelander', 'Mahindra Bolero Di',
'Mahindra Bolero Power', 'Mahindra Bolero SLE',
'Mahindra Jeep CL550', 'Mahindra Jeep MM', 'Mahindra KUV100',
'Mahindra KUV100 K8', 'Mahindra Logan', 'Mahindra Logan Diesel',
'Mahindra Quanto C4', 'Mahindra Quanto C8', 'Mahindra Scorpio',
'Mahindra Scorpio 2.6', 'Mahindra Scorpio LX',
'Mahindra Scorpio S10', 'Mahindra Scorpio S4',
'Mahindra Scorpio SLE', 'Mahindra Scorpio SLX',
'Mahindra Scorpio VLX', 'Mahindra Scorpio Vlx',
'Mahindra Scorpio W', 'Mahindra TUV300 T4', 'Mahindra TUV300 T8',
'Mahindra Thar CRDe', 'Mahindra XUV500', 'Mahindra XUV500 W10',
'Mahindra XUV500 W6', 'Mahindra XUV500 W8', 'Mahindra Xylo D2',
'Mahindra Xylo E4', 'Mahindra Xylo E8', 'Maruti Suzuki 800',
'Maruti Suzuki A', 'Maruti Suzuki Alto', 'Maruti Suzuki Baleno',
'Maruti Suzuki Celerio', 'Maruti Suzuki Ciaz',
'Maruti Suzuki Dzire', 'Maruti Suzuki Eeco',
'Maruti Suzuki Ertiga', 'Maruti Suzuki Esteem',
'Maruti Suzuki Estilo', 'Maruti Suzuki Maruti',
'Maruti Suzuki Omni', 'Maruti Suzuki Ritz', 'Maruti Suzuki S',
'Maruti Suzuki SX4', 'Maruti Suzuki Stingray',
'Maruti Suzuki Swift', 'Maruti Suzuki Versa',
'Maruti Suzuki Vitara', 'Maruti Suzuki Wagon', 'Maruti Suzuki Zen',
'Mercedes Benz A', 'Mercedes Benz B', 'Mercedes Benz C',
'Mercedes Benz GLA', 'Mini Cooper S', 'Mitsubishi Lancer 1.8',
'Mitsubishi Pajero Sport', 'Nissan Micra XL', 'Nissan Micra XV',
'Nissan Sunny', 'Nissan Sunny XL', 'Nissan Terrano XL',
'Nissan X Trail', 'Renault Duster', 'Renault Duster 110',
'Renault Duster 110PS', 'Renault Duster 85', 'Renault Duster 85PS',
'Renault Duster RxL', 'Renault Kwid', 'Renault Kwid 1.0',
'Renault Kwid RXT', 'Renault Lodgy 85', 'Renault Scala RxL',
'Skoda Fabia', 'Skoda Fabia 1.2L', 'Skoda Fabia Classic',
'Skoda Laura', 'Skoda Octavia Classic', 'Skoda Rapid Elegance',
'Skoda Superb 1.8', 'Skoda Yeti Ambition', 'Tata Aria Pleasure',
'Tata Bolt XM', 'Tata Indica', 'Tata Indica V2', 'Tata Indica eV2',
'Tata Indigo CS', 'Tata Indigo LS', 'Tata Indigo LX',
'Tata Indigo Marina', 'Tata Indigo eCS', 'Tata Manza',
'Tata Manza Aqua', 'Tata Manza Aura', 'Tata Manza ELAN',
'Tata Nano', 'Tata Nano Cx', 'Tata Nano GenX', 'Tata Nano LX',
'Tata Nano Lx', 'Tata Sumo Gold', 'Tata Sumo Grande',
'Tata Sumo Victa', 'Tata Tiago Revotorq', 'Tata Tiago Revotron',
'Tata Tigor Revotron', 'Tata Venture EX', 'Tata Vista Quadrajet',
'Tata Zest Quadrajet', 'Tata Zest XE', 'Tata Zest XM',
'Toyota Corolla', 'Toyota Corolla Altis', 'Toyota Corolla H2',
'Toyota Etios', 'Toyota Etios G', 'Toyota Etios GD',
'Toyota Etios Liva', 'Toyota Fortuner', 'Toyota Fortuner 3.0',
'Toyota Innova 2.0', 'Toyota Innova 2.5', 'Toyota Qualis',
'Volkswagen Jetta Comfortline', 'Volkswagen Jetta Highline',
'Volkswagen Passat Diesel', 'Volkswagen Polo',
'Volkswagen Polo Comfortline', 'Volkswagen Polo Highline',
'Volkswagen Polo Highline1.2L', 'Volkswagen Polo Trendline',
'Volkswagen Vento Comfortline', 'Volkswagen Vento Highline',
'Volkswagen Vento Konekt', 'Volvo S80 Summum'], dtype=object)
```