

1 - Understand the Problem 😐

Content ✅

Car_ID -----> Unique id of each observation (Intreger)

Symboling -----> Its assigned insurance risk rating, A value of +3 indicates that the auto is risky, -3 that it is probably pretty safe. (Categorical)

carCompany -----> Name of car company (Categorical)

fueltype -----> Car fuel type i.e gas or diesel (Categorical)

aspiration -----> Aspiration used in a car (Categorical)

doornumber -----> Number of doors in a car (Categorical)

carbody -----> body of car (Categorical)

drivewheel -----> type of drive wheel (Categorical)

enginelocation -----> Location of car engine (Categorical)

wheelbase -----> Weelbase of car (Numeric)

carlength -----> Length of car (Numeric)

carwidth -----> Width of car (Numeric)

carheight -----> height of car (Numeric)

curbweight -----> The weight of a car without occupants or baggage. (Numeric)

enginetype -----> Type of engine. (Categorical)

cylindernumber -----> cylinder placed in the car (Categorical)

enginesize -----> Size of car (Numeric)

fuelsystem -----> Fuel system of car (Categorical)

boreratio -----> Boreratio of car (Numeric)

stroke -----> Stroke or volume inside the engine (Numeric)

compressionratio -----> compression ratio of car (Numeric)

horsepower -----> Horsepower (Numeric)

peakrpm -----> car peak rpm (Numeric)

citympg -----> Mileage in city (Numeric)

highwaympg -----> Mileage on highway (Numeric)

price(Dependent variable) -----> Price of car (Numeric)

target

The main objective is to train a predictive model that enables the company to accurately forecast car prices, helping optimize pricing strategies and maximize revenue in the automotive market.

import

In [843...]

```
import math
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.metrics import mean_squared_error, r2_score
import statsmodels.api as sm
from sklearn.preprocessing import PolynomialFeatures, StandardScaler, OneHotEncoder
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import GridSearchCV
import warnings
warnings.filterwarnings("ignore")
```

2 - Collect and Prepare the Data

In [844...]

```
path = "D:\\\\projects\\\\datasets\\\\CarPrice_Assignment.csv"
df = pd.read_csv(path)
pd.set_option('display.max_columns', len(df.columns))
df.rename(columns=lambda x: x.strip().lower().replace(' ', '_'), inplace=True)
df
```

	car_id	symboling	carname	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwid
0	1	3	alfa-romero giulia	gas	std	two	convertible	rwd	front	88.6	168.8	62
1	2	3	alfa-romero stelvio	gas	std	two	convertible	rwd	front	88.6	168.8	62
2	3	1	alfa-romero Quadrifoglio	gas	std	two	hatchback	rwd	front	94.5	171.2	65
3	4	2	audi 100 ls	gas	std	four	sedan	fwd	front	99.8	176.6	66
4	5	2	audi 100ls	gas	std	four	sedan	4wd	front	99.4	176.6	66
5	6	2	audi fox	gas	std	two	sedan	fwd	front	99.8	177.3	66
6	7	1	audi 100ls	gas	std	four	sedan	fwd	front	105.8	192.7	71
7	8	1	audi 5000	gas	std	four	wagon	fwd	front	105.8	192.7	71
8	9	1	audi 4000	gas	turbo	four	sedan	fwd	front	105.8	192.7	71
9	10	0	audi 5000s (diesel)	gas	turbo	two	hatchback	4wd	front	99.5	178.2	67
10	11	2	bmw 320i	gas	std	two	sedan	rwd	front	101.2	176.8	62
11	12	0	bmw 320i	gas	std	four	sedan	rwd	front	101.2	176.8	62
12	13	0	bmw x1	gas	std	two	sedan	rwd	front	101.2	176.8	62
13	14	0	bmw x3	gas	std	four	sedan	rwd	front	101.2	176.8	62
14	15	1	bmw z4	gas	std	four	sedan	rwd	front	103.5	189.0	66
15	16	0	bmw x4	gas	std	four	sedan	rwd	front	103.5	189.0	66
16	17	0	bmw x5	gas	std	two	sedan	rwd	front	103.5	193.8	67
17	18	0	bmw x3	gas	std	four	sedan	rwd	front	110.0	197.0	70
18	19	2	chevrolet impala	gas	std	two	hatchback	fwd	front	88.4	141.1	60
19	20	1	chevrolet monte carlo	gas	std	two	hatchback	fwd	front	94.5	155.9	63
20	21	0	chevrolet vega 2300	gas	std	four	sedan	fwd	front	94.5	158.8	63

car_id	symboling	carname	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwid
21	22	1 dodge rampage	gas	std	two	hatchback	fwd	front	93.7	157.3	63
22	23	1 dodge challenger se	gas	std	two	hatchback	fwd	front	93.7	157.3	63
23	24	1 dodge d200	gas	turbo	two	hatchback	fwd	front	93.7	157.3	63
24	25	1 dodge monaco (sw)	gas	std	four	hatchback	fwd	front	93.7	157.3	63
25	26	1 dodge colt hardtop	gas	std	four	sedan	fwd	front	93.7	157.3	63
26	27	1 dodge colt (sw)	gas	std	four	sedan	fwd	front	93.7	157.3	63
27	28	1 dodge coronet custom	gas	turbo	two	sedan	fwd	front	93.7	157.3	63
28	29	-1 dodge dart custom	gas	std	four	wagon	fwd	front	103.3	174.6	62
29	30	3 dodge coronet custom (sw)	gas	turbo	two	hatchback	fwd	front	95.9	173.2	66
30	31	2 honda civic	gas	std	two	hatchback	fwd	front	86.6	144.6	63
31	32	2 honda civic cvcc	gas	std	two	hatchback	fwd	front	86.6	144.6	63
32	33	1 honda civic	gas	std	two	hatchback	fwd	front	93.7	150.0	62
33	34	1 honda accord cvcc	gas	std	two	hatchback	fwd	front	93.7	150.0	62
34	35	1 honda civic cvcc	gas	std	two	hatchback	fwd	front	93.7	150.0	62
35	36	0 honda accord lx	gas	std	four	sedan	fwd	front	96.5	163.4	62

car_id	symboling	carnome	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwid
36	37	0	honda civic 1500 gl	gas	std	four	wagon	fwd	front	96.5	157.1
37	38	0	honda accord	gas	std	two	hatchback	fwd	front	96.5	167.5
38	39	0	honda civic 1300	gas	std	two	hatchback	fwd	front	96.5	167.5
39	40	0	honda prelude	gas	std	four	sedan	fwd	front	96.5	175.4
40	41	0	honda accord	gas	std	four	sedan	fwd	front	96.5	175.4
41	42	0	honda civic	gas	std	four	sedan	fwd	front	96.5	175.4
42	43	1	honda civic (auto)	gas	std	two	sedan	fwd	front	96.5	169.1
43	44	0	isuzu MU-X	gas	std	four	sedan	rwd	front	94.3	170.7
44	45	1	isuzu D-Max	gas	std	two	sedan	fwd	front	94.5	155.9
45	46	0	isuzu D-Max V-Cross	gas	std	four	sedan	fwd	front	94.5	155.9
46	47	2	isuzu D-Max	gas	std	two	hatchback	rwd	front	96.0	172.6
47	48	0	jaguar xj	gas	std	four	sedan	rwd	front	113.0	199.6
48	49	0	jaguar xf	gas	std	four	sedan	rwd	front	113.0	199.6
49	50	0	jaguar xk	gas	std	two	sedan	rwd	front	102.0	191.7
50	51	1	maxda rx3	gas	std	two	hatchback	fwd	front	93.1	159.1
51	52	1	maxda glc deluxe	gas	std	two	hatchback	fwd	front	93.1	159.1
52	53	1	mazda rx2 coupe	gas	std	two	hatchback	fwd	front	93.1	159.1
53	54	1	mazda rx-4	gas	std	four	sedan	fwd	front	93.1	166.8
54	55	1	mazda glc deluxe	gas	std	four	sedan	fwd	front	93.1	166.8

car_id	symboling	carname	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwid
55	56	3	mazda 626	gas	std	two	hatchback	rwd	front	95.3	169.0
56	57	3	mazda glc	gas	std	two	hatchback	rwd	front	95.3	169.0
57	58	3	mazda rx-7 gs	gas	std	two	hatchback	rwd	front	95.3	169.0
58	59	3	mazda glc 4	gas	std	two	hatchback	rwd	front	95.3	169.0
59	60	1	mazda 626	gas	std	two	hatchback	fwd	front	98.8	177.8
60	61	0	mazda glc custom l	gas	std	four	sedan	fwd	front	98.8	177.8
61	62	1	mazda glc custom	gas	std	two	hatchback	fwd	front	98.8	177.8
62	63	0	mazda rx-4	gas	std	four	sedan	fwd	front	98.8	177.8
63	64	0	mazda glc deluxe	diesel	std	four	sedan	fwd	front	98.8	177.8
64	65	0	mazda 626	gas	std	four	hatchback	fwd	front	98.8	177.8
65	66	0	mazda glc	gas	std	four	sedan	rwd	front	104.9	175.0
66	67	0	mazda rx-7 gs	diesel	std	four	sedan	rwd	front	104.9	175.0
67	68	-1	buick electra 225 custom	diesel	turbo	four	sedan	rwd	front	110.0	190.9
68	69	-1	buick century luxus (sw)	diesel	turbo	four	wagon	rwd	front	110.0	190.9
69	70	0	buick century	diesel	turbo	two	hardtop	rwd	front	106.7	187.5
70	71	-1	buick skyhawk	diesel	turbo	four	sedan	rwd	front	115.6	202.6
71	72	-1	buick opel isuzu deluxe	gas	std	four	sedan	rwd	front	115.6	202.6
72	73	3	buick skylark	gas	std	two	convertible	rwd	front	96.6	180.3

car_id	symboling	carname	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwid
73	74	0 buick century special	gas	std	four	sedan	rwd	front	120.9	208.1	71
74	75	1 buick regal sport coupe (turbo)	gas	std	two	hardtop	rwd	front	112.0	199.2	72
75	76	1 mercury cougar	gas	turbo	two	hatchback	rwd	front	102.7	178.4	68
76	77	2 mitsubishi mirage	gas	std	two	hatchback	fwd	front	93.7	157.3	62
77	78	2 mitsubishi lancer	gas	std	two	hatchback	fwd	front	93.7	157.3	62
78	79	2 mitsubishi outlander	gas	std	two	hatchback	fwd	front	93.7	157.3	62
79	80	1 mitsubishi g4	gas	turbo	two	hatchback	fwd	front	93.0	157.3	63
80	81	3 mitsubishi mirage g4	gas	turbo	two	hatchback	fwd	front	96.3	173.0	65
81	82	3 mitsubishi g4	gas	std	two	hatchback	fwd	front	96.3	173.0	65
82	83	3 mitsubishi outlander	gas	turbo	two	hatchback	fwd	front	95.9	173.2	66
83	84	3 mitsubishi g4	gas	turbo	two	hatchback	fwd	front	95.9	173.2	66
84	85	3 mitsubishi mirage g4	gas	turbo	two	hatchback	fwd	front	95.9	173.2	66
85	86	1 mitsubishi montero	gas	std	four	sedan	fwd	front	96.3	172.4	65
86	87	1 mitsubishi pajero	gas	std	four	sedan	fwd	front	96.3	172.4	65
87	88	1 mitsubishi outlander	gas	turbo	four	sedan	fwd	front	96.3	172.4	65

car_id	symboling	carname	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwid
88	89	-1	mitsubishi mirage g4	gas	std	four	sedan	fwd	front	96.3	172.4
89	90	1	Nissan versa	gas	std	two	sedan	fwd	front	94.5	165.3
90	91	1	nissan gt-r	diesel	std	two	sedan	fwd	front	94.5	165.3
91	92	1	nissan rogue	gas	std	two	sedan	fwd	front	94.5	165.3
92	93	1	nissan latio	gas	std	four	sedan	fwd	front	94.5	165.3
93	94	1	nissan titan	gas	std	four	wagon	fwd	front	94.5	170.2
94	95	1	nissan leaf	gas	std	two	sedan	fwd	front	94.5	165.3
95	96	1	nissan juke	gas	std	two	hatchback	fwd	front	94.5	165.6
96	97	1	nissan latio	gas	std	four	sedan	fwd	front	94.5	165.3
97	98	1	nissan note	gas	std	four	wagon	fwd	front	94.5	170.2
98	99	2	nissan clipper	gas	std	two	hardtop	fwd	front	95.1	162.4
99	100	0	nissan rogue	gas	std	four	hatchback	fwd	front	97.2	173.4
100	101	0	nissan nv200	gas	std	four	sedan	fwd	front	97.2	173.4
101	102	0	nissan dayz	gas	std	four	sedan	fwd	front	100.4	181.7
102	103	0	nissan fuga	gas	std	four	wagon	fwd	front	100.4	184.6
103	104	0	nissan otti	gas	std	four	sedan	fwd	front	100.4	184.6
104	105	3	nissan teana	gas	std	two	hatchback	rwd	front	91.3	170.7
105	106	3	nissan kicks	gas	turbo	two	hatchback	rwd	front	91.3	170.7
106	107	1	nissan clipper	gas	std	two	hatchback	rwd	front	99.2	178.5
107	108	0	peugeot 504	gas	std	four	sedan	rwd	front	107.9	186.7

car_id	symboling	carname	fuelytype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwid
108	109	0	peugeot 304	diesel	turbo	four	sedan	rwd	front	107.9	186.7
109	110	0	peugeot 504 (sw)	gas	std	four	wagon	rwd	front	114.2	198.9
110	111	0	peugeot 504	diesel	turbo	four	wagon	rwd	front	114.2	198.9
111	112	0	peugeot 504	gas	std	four	sedan	rwd	front	107.9	186.7
112	113	0	peugeot 604sl	diesel	turbo	four	sedan	rwd	front	107.9	186.7
113	114	0	peugeot 504	gas	std	four	wagon	rwd	front	114.2	198.9
114	115	0	peugeot 505s turbo diesel	diesel	turbo	four	wagon	rwd	front	114.2	198.9
115	116	0	peugeot 504	gas	std	four	sedan	rwd	front	107.9	186.7
116	117	0	peugeot 504	diesel	turbo	four	sedan	rwd	front	107.9	186.7
117	118	0	peugeot 604sl	gas	turbo	four	sedan	rwd	front	108.0	186.7
118	119	1	plymouth fury iii	gas	std	two	hatchback	fwd	front	93.7	157.3
119	120	1	plymouth cricket	gas	turbo	two	hatchback	fwd	front	93.7	157.3
120	121	1	plymouth fury iii	gas	std	four	hatchback	fwd	front	93.7	157.3
121	122	1	plymouth satellite custom (sw)	gas	std	four	sedan	fwd	front	93.7	167.3
122	123	1	plymouth fury gran sedan	gas	std	four	sedan	fwd	front	93.7	167.3

car_id	symboling	carnome	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwid
123	124	-1	plymouth valiant	gas	std	four	wagon	fwd	front	103.3	174.6
124	125	3	plymouth duster	gas	turbo	two	hatchback	rwd	front	95.9	173.2
125	126	3	porsche macan	gas	std	two	hatchback	rwd	front	94.5	168.9
126	127	3	porcshce panamera	gas	std	two	hardtop	rwd	rear	89.5	168.9
127	128	3	porsche cayenne	gas	std	two	hardtop	rwd	rear	89.5	168.9
128	129	3	porsche boxter	gas	std	two	convertible	rwd	rear	89.5	168.9
129	130	1	porsche cayenne	gas	std	two	hatchback	rwd	front	98.4	175.7
130	131	0	renault 12tl	gas	std	four	wagon	fwd	front	96.1	181.5
131	132	2	renault 5 gtl	gas	std	two	hatchback	fwd	front	96.1	176.8
132	133	3	saab 99e	gas	std	two	hatchback	fwd	front	99.1	186.6
133	134	2	saab 99le	gas	std	four	sedan	fwd	front	99.1	186.6
134	135	3	saab 99le	gas	std	two	hatchback	fwd	front	99.1	186.6
135	136	2	saab 99gle	gas	std	four	sedan	fwd	front	99.1	186.6
136	137	3	saab 99gle	gas	turbo	two	hatchback	fwd	front	99.1	186.6
137	138	2	saab 99e	gas	turbo	four	sedan	fwd	front	99.1	186.6
138	139	2	subaru	gas	std	two	hatchback	fwd	front	93.7	156.9
139	140	2	subaru dl	gas	std	two	hatchback	fwd	front	93.7	157.9
140	141	2	subaru dl	gas	std	two	hatchback	4wd	front	93.3	157.3
141	142	0	subaru	gas	std	four	sedan	fwd	front	97.2	172.0
142	143	0	subaru brz	gas	std	four	sedan	fwd	front	97.2	172.0
143	144	0	subaru baja	gas	std	four	sedan	fwd	front	97.2	172.0

car_id	symboling	carname	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwid
144	145	0	subaru r1	gas	std	four	sedan	4wd	front	97.0	172.0
145	146	0	subaru r2	gas	turbo	four	sedan	4wd	front	97.0	172.0
146	147	0	subaru trezia	gas	std	four	wagon	fwd	front	97.0	173.5
147	148	0	subaru tribeca	gas	std	four	wagon	fwd	front	97.0	173.5
148	149	0	subaru dl	gas	std	four	wagon	4wd	front	96.9	173.6
149	150	0	subaru dl	gas	turbo	four	wagon	4wd	front	96.9	173.6
150	151	1	toyota corona mark ii	gas	std	two	hatchback	fwd	front	95.7	158.7
151	152	1	toyota corona	gas	std	two	hatchback	fwd	front	95.7	158.7
152	153	1	toyota corolla 1200	gas	std	four	hatchback	fwd	front	95.7	158.7
153	154	0	toyota corona hardtop	gas	std	four	wagon	fwd	front	95.7	169.7
154	155	0	toyota corolla 1600 (sw)	gas	std	four	wagon	4wd	front	95.7	169.7
155	156	0	toyota carina	gas	std	four	wagon	4wd	front	95.7	169.7
156	157	0	toyota mark ii	gas	std	four	sedan	fwd	front	95.7	166.3
157	158	0	toyota corolla 1200	gas	std	four	hatchback	fwd	front	95.7	166.3
158	159	0	toyota corona	diesel	std	four	sedan	fwd	front	95.7	166.3
159	160	0	toyota corolla	diesel	std	four	hatchback	fwd	front	95.7	166.3

	car_id	symboling	carname	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwid
160	161	0	toyota corona	gas	std	four	sedan	fwd	front	95.7	166.3	62
161	162	0	toyota corolla	gas	std	four	hatchback	fwd	front	95.7	166.3	62
162	163	0	toyota mark ii	gas	std	four	sedan	fwd	front	95.7	166.3	62
163	164	1	toyota corolla liftback	gas	std	two	sedan	rwd	front	94.5	168.7	62
164	165	1	toyota corona	gas	std	two	hatchback	rwd	front	94.5	168.7	62
165	166	1	toyota celica gt liftback	gas	std	two	sedan	rwd	front	94.5	168.7	62
166	167	1	toyota corolla tercel	gas	std	two	hatchback	rwd	front	94.5	168.7	62
167	168	2	toyota corona liftback	gas	std	two	hardtop	rwd	front	98.4	176.2	65
168	169	2	toyota corolla	gas	std	two	hardtop	rwd	front	98.4	176.2	65
169	170	2	toyota starlet	gas	std	two	hatchback	rwd	front	98.4	176.2	65
170	171	2	toyota tercel	gas	std	two	hardtop	rwd	front	98.4	176.2	65
171	172	2	toyota corolla	gas	std	two	hatchback	rwd	front	98.4	176.2	65
172	173	2	toyota cressida	gas	std	two	convertible	rwd	front	98.4	176.2	65
173	174	-1	toyota corolla	gas	std	four	sedan	fwd	front	102.4	175.6	66
174	175	-1	toyota celica gt	diesel	turbo	four	sedan	fwd	front	102.4	175.6	66

car_id	symboling	carnome	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwid
175	176	-1	toyota corona	gas	std	four	hatchback	fwd	front	102.4	175.6
176	177	-1	toyota corolla	gas	std	four	sedan	fwd	front	102.4	175.6
177	178	-1	toyota mark ii	gas	std	four	hatchback	fwd	front	102.4	175.6
178	179	3	toyota corolla liftback	gas	std	two	hatchback	rwd	front	102.9	183.5
179	180	3	toyota corona	gas	std	two	hatchback	rwd	front	102.9	183.5
180	181	-1	toyota starlet	gas	std	four	sedan	rwd	front	104.5	187.8
181	182	-1	toyota tercel	gas	std	four	wagon	rwd	front	104.5	187.8
182	183	2	volkswagen rabbit	diesel	std	two	sedan	fwd	front	97.3	171.7
183	184	2	volkswagen 1131 deluxe sedan	gas	std	two	sedan	fwd	front	97.3	171.7
184	185	2	volkswagen model 111	diesel	std	four	sedan	fwd	front	97.3	171.7
185	186	2	volkswagen type 3	gas	std	four	sedan	fwd	front	97.3	171.7
186	187	2	volkswagen 411 (sw)	gas	std	four	sedan	fwd	front	97.3	171.7
187	188	2	volkswagen super beetle	diesel	turbo	four	sedan	fwd	front	97.3	171.7
188	189	2	volkswagen dasher	gas	std	four	sedan	fwd	front	97.3	171.7
189	190	3	vw dasher	gas	std	two	convertible	fwd	front	94.5	159.3
190	191	3	vw rabbit	gas	std	two	hatchback	fwd	front	94.5	165.7

car_id	symboling	carname	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwid
191	192	0	volkswagen rabbit	gas	std	four	sedan	fwd	front	100.4	180.2
192	193	0	volkswagen rabbit custom	diesel	turbo	four	sedan	fwd	front	100.4	180.2
193	194	0	volkswagen dasher	gas	std	four	wagon	fwd	front	100.4	183.1
194	195	-2	volvo 145e (sw)	gas	std	four	sedan	rwd	front	104.3	188.8
195	196	-1	volvo 144ea	gas	std	four	wagon	rwd	front	104.3	188.8
196	197	-2	volvo 244dl	gas	std	four	sedan	rwd	front	104.3	188.8
197	198	-1	volvo 245	gas	std	four	wagon	rwd	front	104.3	188.8
198	199	-2	volvo 264gl	gas	turbo	four	sedan	rwd	front	104.3	188.8
199	200	-1	volvo diesel	gas	turbo	four	wagon	rwd	front	104.3	188.8
200	201	-1	volvo 145e (sw)	gas	std	four	sedan	rwd	front	109.1	188.8
201	202	-1	volvo 144ea	gas	turbo	four	sedan	rwd	front	109.1	188.8
202	203	-1	volvo 244dl	gas	std	four	sedan	rwd	front	109.1	188.8
203	204	-1	volvo 246	diesel	turbo	four	sedan	rwd	front	109.1	188.8
204	205	-1	volvo 264gl	gas	turbo	four	sedan	rwd	front	109.1	188.8

Exploratory Data Analysis (EDA)

In [845...]

[df.info\(\)](#)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
 #   Column      Non-Null Count Dtype  
 --- 
 0   car_id      205 non-null   int64  
 1   symboling   205 non-null   int64  
 2   carname     205 non-null   object  
 3   fuelytype   205 non-null   object  
 4   aspiration   205 non-null   object  
 5   doornumber   205 non-null   object  
 6   carbody     205 non-null   object  
 7   drivewheel   205 non-null   object  
 8   enginelocation 205 non-null   object  
 9   wheelbase    205 non-null   float64 
 10  carlength    205 non-null   float64 
 11  carwidth     205 non-null   float64 
 12  carheight    205 non-null   float64 
 13  curbweight   205 non-null   int64  
 14  enginetype   205 non-null   object  
 15  cylindernumber 205 non-null   object  
 16  enginesize    205 non-null   int64  
 17  fuelsystem   205 non-null   object  
 18  borerratio   205 non-null   float64 
 19  stroke       205 non-null   float64 
 20  compressionratio 205 non-null   float64 
 21  horsepower    205 non-null   int64  
 22  peakrmp      205 non-null   int64  
 23  citympg      205 non-null   int64  
 24  highwaympg    205 non-null   int64  
 25  price        205 non-null   float64 
dtypes: float64(8), int64(8), object(10)
memory usage: 41.8+ KB
```

In [846...]

```
df.describe()
```

Out[846]:

	car_id	symboling	wheelbase	carlength	carwidth	carheight	curbweight	enginesize	boreratio	stroke	compressioni
count	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000
mean	103.000000	0.834146	98.756585	174.049268	65.907805	53.724878	2555.565854	126.907317	3.329756	3.255415	10.141
std	59.322565	1.245307	6.021776	12.337289	2.145204	2.443522	520.680204	41.642693	0.270844	0.313597	3.971
min	1.000000	-2.000000	86.600000	141.100000	60.300000	47.800000	1488.000000	61.000000	2.540000	2.070000	7.000
25%	52.000000	0.000000	94.500000	166.300000	64.100000	52.000000	2145.000000	97.000000	3.150000	3.110000	8.600
50%	103.000000	1.000000	97.000000	173.200000	65.500000	54.100000	2414.000000	120.000000	3.310000	3.290000	9.000
75%	154.000000	2.000000	102.400000	183.100000	66.900000	55.500000	2935.000000	141.000000	3.580000	3.410000	9.400
max	205.000000	3.000000	120.900000	208.100000	72.300000	59.800000	4066.000000	326.000000	3.940000	4.170000	23.000

In [847...]

```
df.isna().sum().to_frame()
```

Out[847] :

	0
car_id	0
symboling	0
carname	0
fueltype	0
aspiration	0
doornumber	0
carbody	0
drivewheel	0
enginelocation	0
wheelbase	0
carlength	0
carwidth	0
carheight	0
curbweight	0
enginetype	0
cylindernumber	0
enginesize	0
fuelsystem	0
boreratio	0
stroke	0
compressionratio	0
horsepower	0
peakrpm	0
citympg	0
highwaympg	0

0

price 0

In [848... df.duplicated().sum()

Out[848]: 0

In [849... df['carname'].value_counts().to_frame()

Out[849] :

carname	count
toyota corona	6
toyota corolla	6
peugeot 504	6
subaru dl	4
mitsubishi mirage g4	3
mazda 626	3
toyota mark ii	3
mitsubishi outlander	3
mitsubishi g4	3
honda civic	3
volvo 264gl	2
bmw 320i	2
isuzu D-Max	2
audi 100ls	2
volvo 244dl	2
porsche cayenne	2
toyota corolla liftback	2
honda accord	2
bmw x3	2
volvo 144ea	2
volvo 145e (sw)	2
honda civic cvcc	2
saab 99le	2
toyota starlet	2

	count
carname	
saab 99gle	2
toyota corolla 1200	2
mazda rx-4	2
mazda glc deluxe	2
nissan rogue	2
nissan latio	2
mazda glc	2
mazda rx-7 gs	2
plymouth fury iii	2
volkswagen dasher	2
subaru	2
nissan clipper	2
peugeot 604sl	2
saab 99e	2
peugeot 304	1
peugeot 504 (sw)	1
renault 5 gtl	1
renault 12tl	1
porsche boxter	1
subaru brz	1
porsche macan	1
plymouth duster	1
plymouth valiant	1
plymouth fury gran sedan	1

	count
carname	
plymouth satellite custom (sw)	1
plymouth cricket	1
peugeot 505s turbo diesel	1
porcshce panamera	1
toyota corona mark ii	1
subaru baja	1
toyota celica gt	1
volvo 245	1
volkswagen rabbit custom	1
volkswagen rabbit	1
vw rabbit	1
vw dasher	1
volkswagen super beetle	1
volkswagen 411 (sw)	1
volkswagen type 3	1
volkswagen model 111	1
volkswagen 1131 deluxe sedan	1
vokswagen rabbit	1
toyouta tercel	1
toyota cressida	1
subaru r1	1
toyota tercel	1
toyota corona liftback	1
toyota corolla tercel	1

	count
carname	
toyota celica gt liftback	1
volvo diesel	1
toyota carina	1
toyota corolla 1600 (sw)	1
toyota corona hardtop	1
nissan kicks	1
subaru tribeca	1
subaru trezia	1
subaru r2	1
alfa-romero giulia	1
nissan titan	1
nissan teana	1
nissan otti	1
isuzu MU-X	1
honda civic (auto)	1
honda prelude	1
honda civic 1300	1
honda civic 1500 gl	1
honda accord lx	1
honda accord cvcc	1
dodge coronet custom (sw)	1
dodge dart custom	1
dodge coronet custom	1
dodge colt (sw)	1

	count
carname	
dodge colt hardtop	1
dodge monaco (sw)	1
dodge d200	1
dodge challenger se	1
dodge rampage	1
chevrolet vega 2300	1
chevrolet monte carlo	1
chevrolet impala	1
bmw x5	1
bmw x4	1
bmw z4	1
bmw x1	1
audi 5000s (diesel)	1
audi 4000	1
audi 5000	1
audi fox	1
audi 100 ls	1
alfa-romero Quadrifoglio	1
isuzu D-Max V-Cross	1
jaguar xj	1
jaguar xf	1
mercury cougar	1
nissan fuga	1
nissan dayz	1

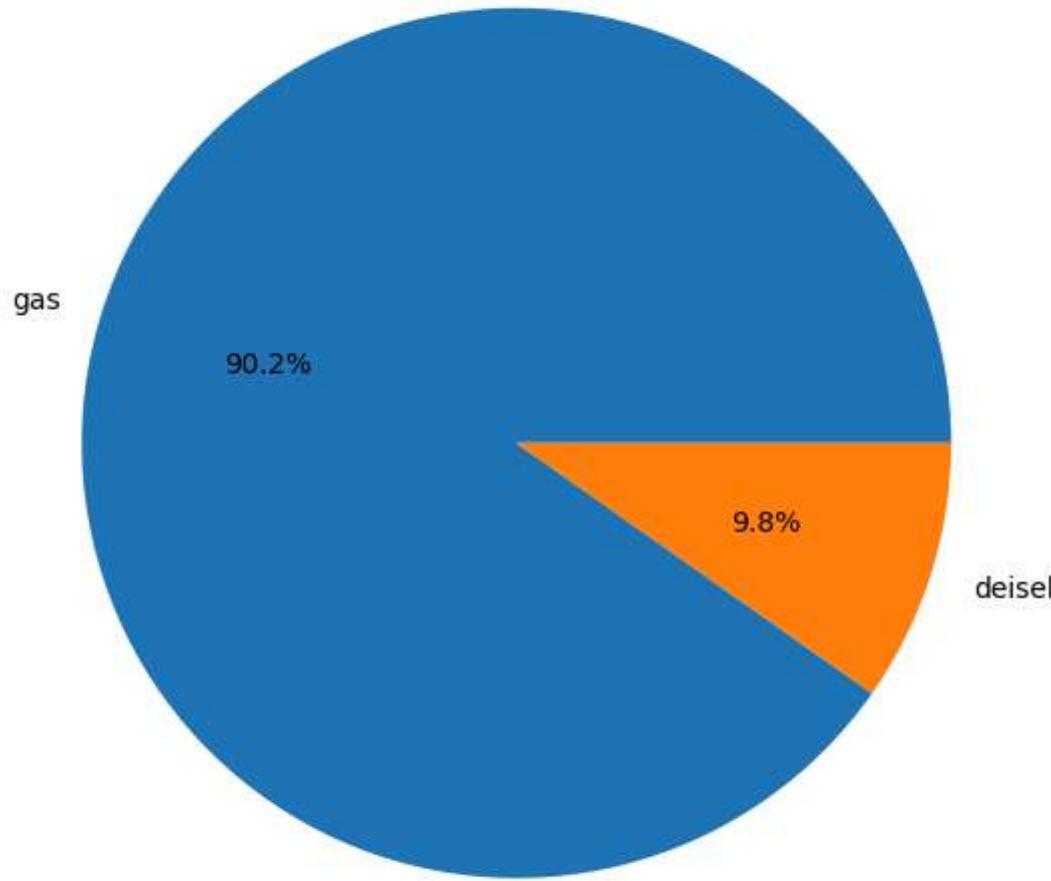
	count
carname	
nissan nv200	1
nissan note	1
nissan juke	1
nissan leaf	1
alfa-romero stelvio	1
nissan gt-r	1
Nissan versa	1
mitsubishi pajero	1
mitsubishi montero	1
mitsubishi lancer	1
mitsubishi mirage	1
buick regal sport coupe (turbo)	1
jaguar xk	1
buick century special	1
buick skylark	1
buick opel isuzu deluxe	1
buick skyhawk	1
buick century	1
buick century luxus (sw)	1
buick electra 225 custom	1
mazda glc custom	1
mazda glc custom l	1
mazda glc 4	1
mazda rx2 coupe	1

	count
carname	
maxda glc deluxe	1
maxda rx3	1
volvo 246	1

In [850...]

```
fueltype = df['fueltype'].value_counts()
plt.figure(figsize=(7,10))
plt.pie(fueltype,labels=['gas','deisel'] , autopct='%1.1f%%')
plt.title('fuel types')
plt.show()
```

fuel types

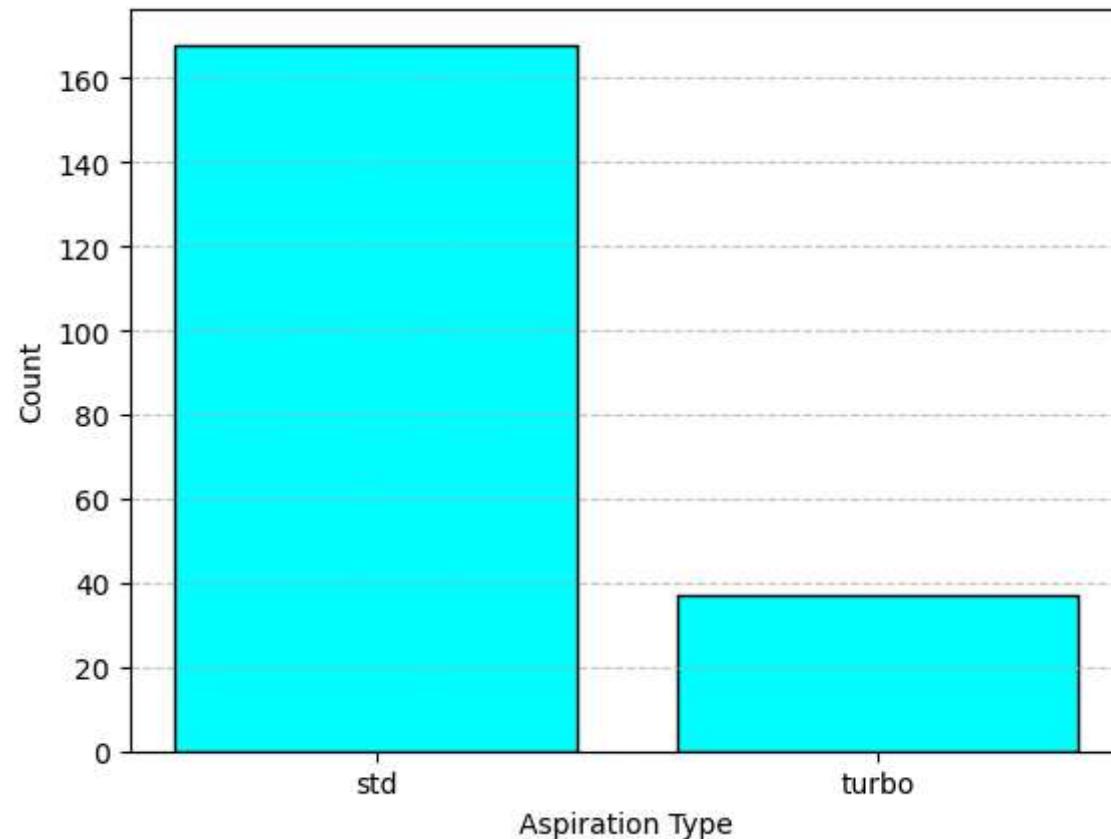


gas cars dominates

In [851]:

```
aspiration = df['aspiration'].value_counts()
plt.bar(['std','turbo'],aspiration, color='cyan', edgecolor='black')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.xlabel('Aspiration Type')
```

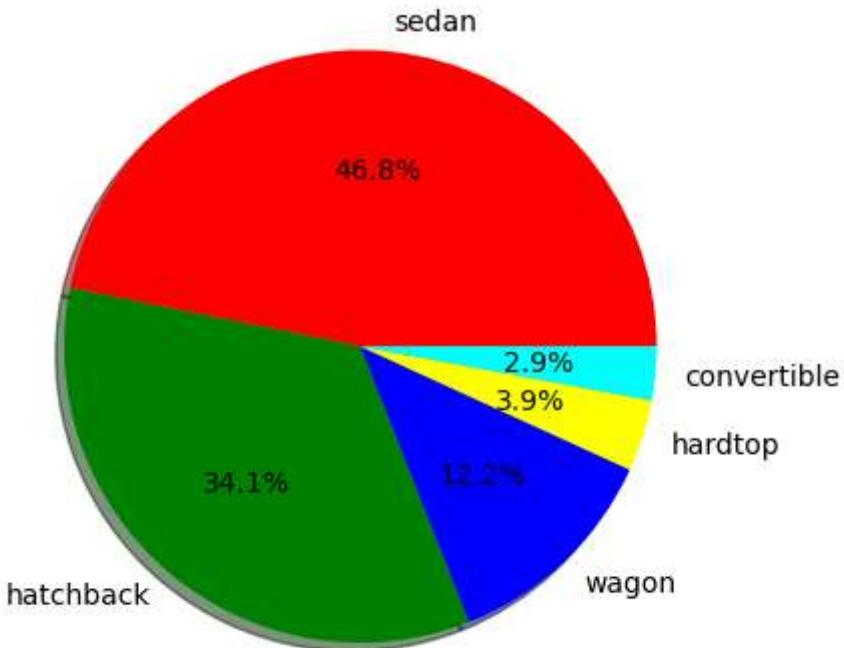
```
plt.ylabel('Count')
plt.show()
```



the majority of the cars are air aspiration (std)

```
In [852...]: carbody = df['carbody'].value_counts().sort_values(ascending=False)
print(carbody)
labels = ['sedan','hatchback','wagon','hardtop','convertible']
plt.pie(carbody,colors=['red','green','blue','yellow','cyan'],labels=labels,shadow=True, autopct='%1.1f%%')
plt.show()
```

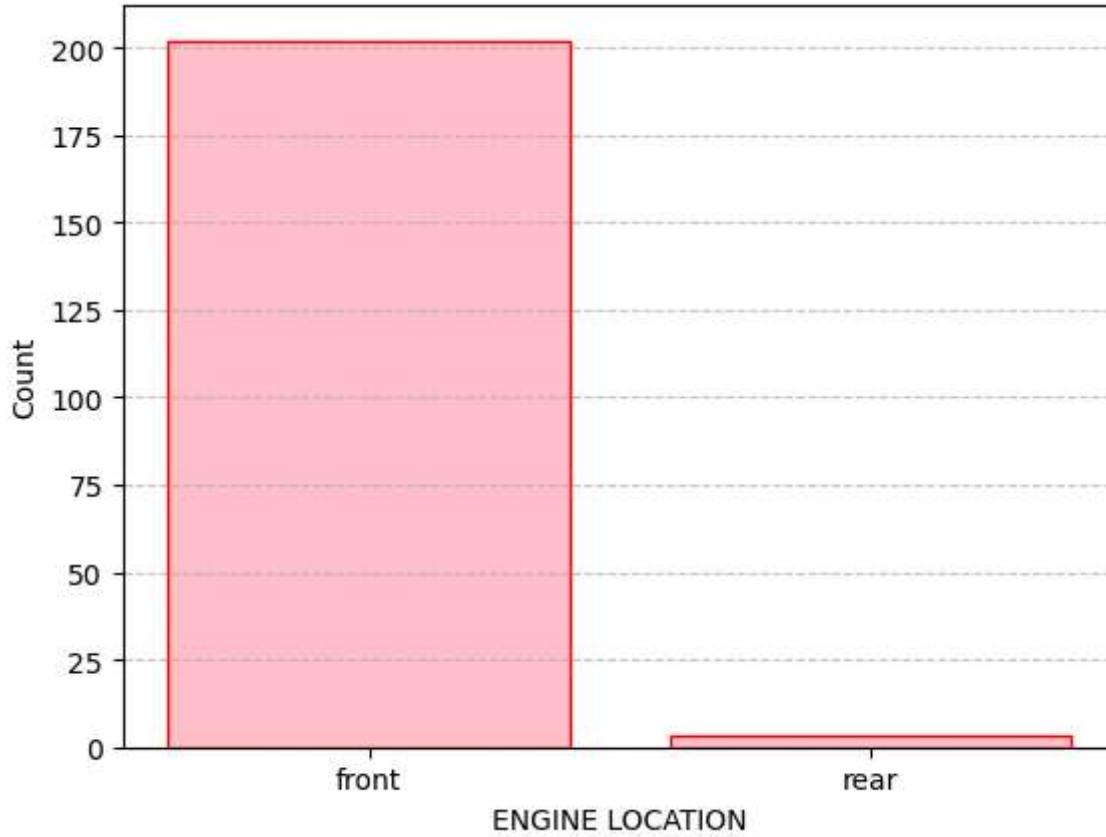
```
carbody
sedan    96
hatchback   70
wagon     25
hardtop      8
convertible    6
Name: count, dtype: int64
```



In [853]:

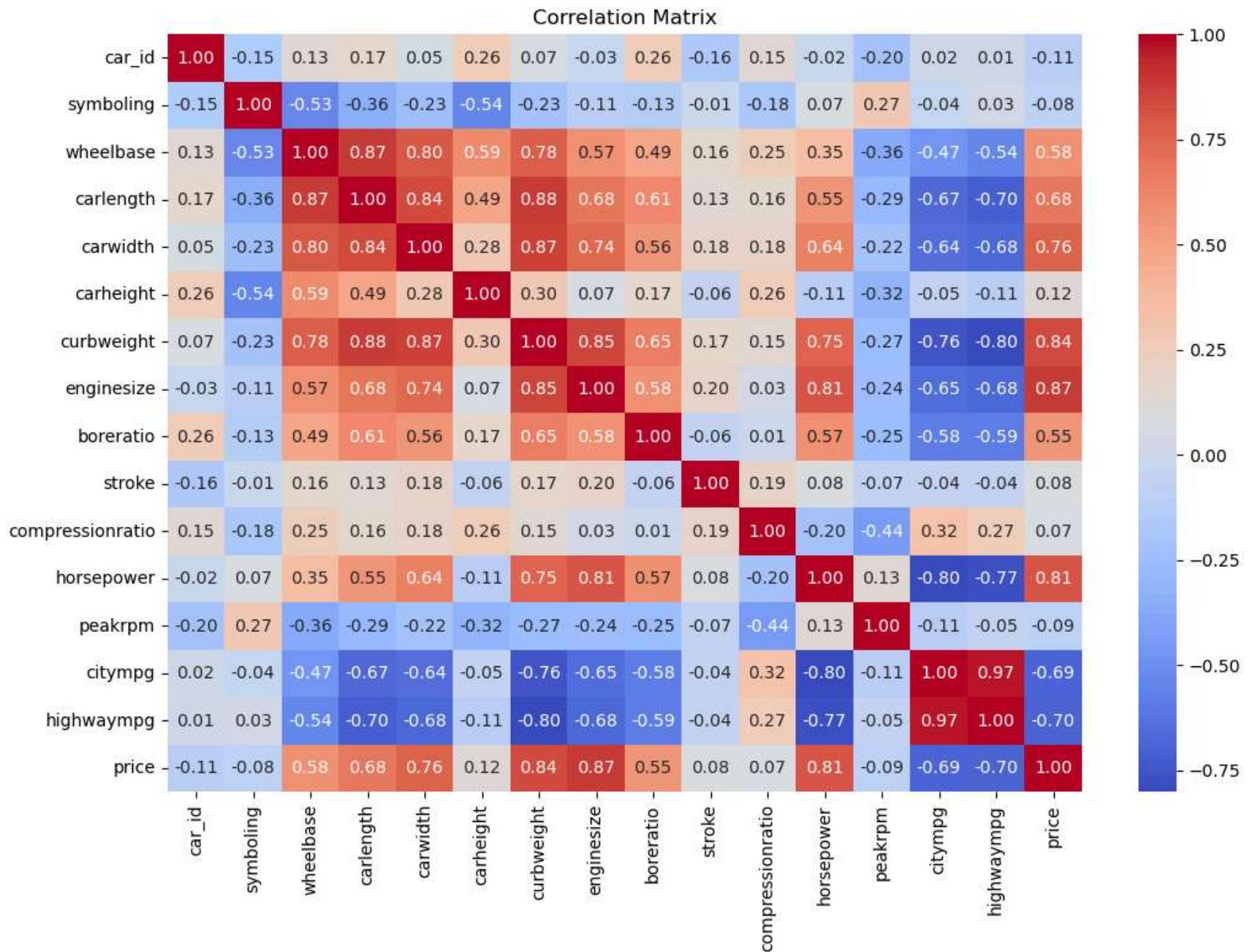
```
englocation = df['enginolocation'].value_counts()
print(englocation)
plt.bar(['front','rear'],englocation, color='pink', edgecolor='red')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.xlabel('ENGINE LOCATION')
plt.ylabel('Count')
plt.show()
```

```
enginolocation
front    202
rear      3
Name: count, dtype: int64
```

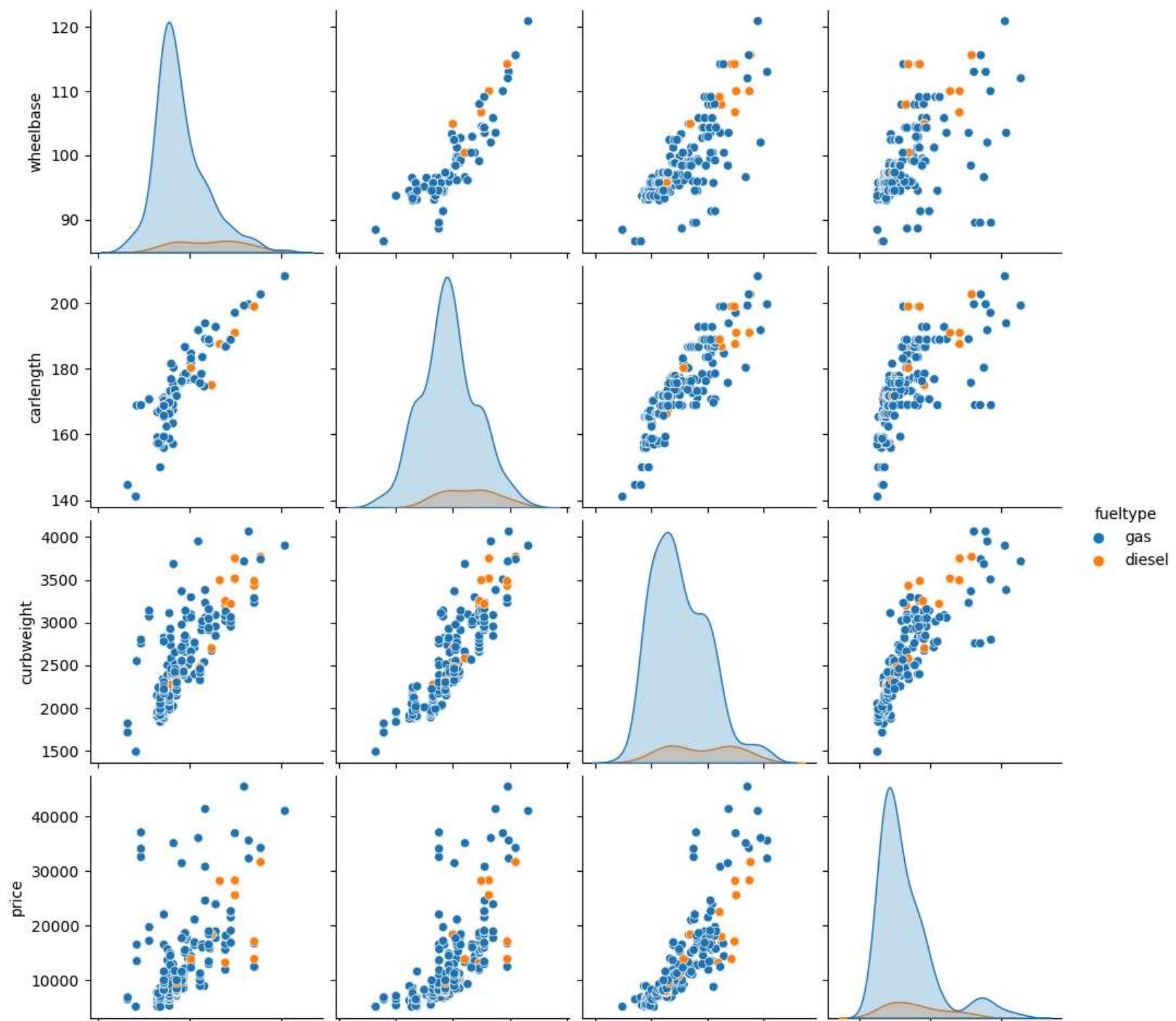


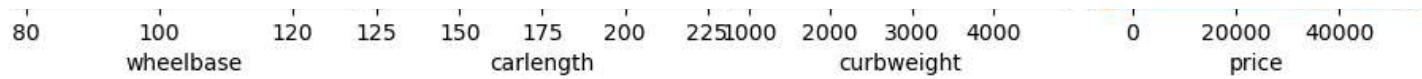
```
In [854...]:  
correlation_matrix = df.select_dtypes(include=['float64', 'int64']).corr()  
print(correlation_matrix['price'].sort_values(ascending=False))  
plt.figure(figsize=(12, 8))  
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm')  
plt.title('Correlation Matrix')  
plt.show()
```

```
price      1.000000
enginesize   0.874145
curbweight    0.835305
horsepower    0.808139
carwidth     0.759325
carlength     0.682920
wheelbase     0.577816
boreratio     0.553173
carheight     0.119336
stroke       0.079443
compressionratio 0.067984
symboling     -0.079978
peakrpm      -0.085267
car_id        -0.109093
citympg       -0.685751
highwaympg    -0.697599
Name: price, dtype: float64
```



```
In [855]: sns.pairplot(df, vars=['wheelbase', 'carlength', 'curbweight', 'price'], hue='fueltype')
plt.show()
```





Feature Engineering

In [856]:

```
df['hp_weight_ratio'] = df['horsepower'] / df['curbweight']
print(df['hp_weight_ratio'].to_frame())
```

	hp_weight_ratio
0	0.043564
1	0.043564
2	0.054552
3	0.043646
4	0.040722
5	0.043877
6	0.038678
7	0.037238
8	0.045366
9	0.052407
10	0.042171
11	0.042171
12	0.044649
13	0.043761
14	0.039607
15	0.056347
16	0.053846
17	0.051926
18	0.032258
19	0.037353
20	0.036668
21	0.036247
22	0.036247
23	0.047932
24	0.034570
25	0.034188
26	0.034188
27	0.046554
28	0.034714
29	0.051583
30	0.033859
31	0.041781
32	0.032662
33	0.039175
34	0.038855
35	0.037811
36	0.037549
37	0.038462
38	0.037571
39	0.037326
40	0.036256
41	0.040974
42	0.043611
43	0.033376

44	0.037353
45	0.036668
46	0.032919
47	0.043286
48	0.043286
49	0.066329
50	0.035979
51	0.035789
52	0.035696
53	0.034961
54	0.034872
55	0.042437
56	0.042437
57	0.042348
58	0.054000
59	0.035220
60	0.034855
61	0.035220
62	0.034855
63	0.026197
64	0.034639
65	0.044944
66	0.026667
67	0.034993
68	0.032800
69	0.035193
70	0.032626
71	0.041444
72	0.042062
73	0.047179
74	0.049529
75	0.060137
76	0.035454
77	0.034979
78	0.033932
79	0.047552
80	0.048945
81	0.037801
82	0.051182
83	0.049641
84	0.049556
85	0.037209
86	0.036590
87	0.048273
88	0.048273

89	0.036527
90	0.027268
91	0.035975
92	0.035604
93	0.034091
94	0.035366
95	0.034024
96	0.035008
97	0.033873
98	0.034363
99	0.041738
100	0.042137
101	0.049111
102	0.046117
103	0.049673
104	0.052100
105	0.063715
106	0.050972
107	0.032119
108	0.029715
109	0.030031
110	0.027697
111	0.030894
112	0.029213
113	0.028919
114	0.027260
115	0.031545
116	0.029213
117	0.045367
118	0.035454
119	0.047932
120	0.034570
121	0.034188
122	0.031036
123	0.034714
124	0.051455
125	0.051476
126	0.075109
127	0.075109
128	0.073929
129	0.085561
130	0.034897
131	0.036585
132	0.041384
133	0.040816

134	0.040635
135	0.039884
136	0.056980
137	0.056200
138	0.033659
139	0.034434
140	0.032589
141	0.038228
142	0.037443
143	0.040171
144	0.034382
145	0.044223
146	0.035808
147	0.038289
148	0.033884
149	0.041887
150	0.031234
151	0.030392
152	0.030769
153	0.027193
154	0.027074
155	0.019936
156	0.033638
157	0.033191
158	0.024615
159	0.024615
160	0.033429
161	0.032988
162	0.032710
163	0.032273
164	0.031760
165	0.049448
166	0.048696
167	0.045669
168	0.045741
169	0.045472
170	0.043300
171	0.042741
172	0.038992
173	0.039553
174	0.029435
175	0.038111
176	0.038111
177	0.037429
178	0.054099

```
179    0.053382
180    0.049824
181    0.049508
182    0.022999
183    0.038479
184    0.022968
185    0.038427
186    0.037363
187    0.029323
188    0.043478
189    0.039929
190    0.040522
191    0.041338
192    0.026367
193    0.034335
194    0.039148
195    0.037574
196    0.038842
197    0.037475
198    0.053202
199    0.051315
200    0.038618
201    0.052476
202    0.044489
203    0.032950
204    0.037231
```

Preprocessing

In [857...]

```
print("null value sum : ",df.isna().sum().sum())
print("duplicates value sum : ",df.duplicated().sum())
#no need for cleaning or fill_nullvalues
```

```
null value sum : 0
duplicates value sum : 0
```

In [858...]

```
copy = df.copy()
copy.drop(columns=['car_id','carname'],inplace=True)
copy
```

Out[858]:

	symboling	fueltype	aspiration	doomnumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbw
0	3	gas	std	two	convertible	rwd	front	88.6	168.8	64.1	48.8	
1	3	gas	std	two	convertible	rwd	front	88.6	168.8	64.1	48.8	
2	1	gas	std	two	hatchback	rwd	front	94.5	171.2	65.5	52.4	
3	2	gas	std	four	sedan	fwd	front	99.8	176.6	66.2	54.3	
4	2	gas	std	four	sedan	4wd	front	99.4	176.6	66.4	54.3	
5	2	gas	std	two	sedan	fwd	front	99.8	177.3	66.3	53.1	
6	1	gas	std	four	sedan	fwd	front	105.8	192.7	71.4	55.7	
7	1	gas	std	four	wagon	fwd	front	105.8	192.7	71.4	55.7	
8	1	gas	turbo	four	sedan	fwd	front	105.8	192.7	71.4	55.9	
9	0	gas	turbo	two	hatchback	4wd	front	99.5	178.2	67.9	52.0	
10	2	gas	std	two	sedan	rwd	front	101.2	176.8	64.8	54.3	
11	0	gas	std	four	sedan	rwd	front	101.2	176.8	64.8	54.3	
12	0	gas	std	two	sedan	rwd	front	101.2	176.8	64.8	54.3	
13	0	gas	std	four	sedan	rwd	front	101.2	176.8	64.8	54.3	
14	1	gas	std	four	sedan	rwd	front	103.5	189.0	66.9	55.7	
15	0	gas	std	four	sedan	rwd	front	103.5	189.0	66.9	55.7	
16	0	gas	std	two	sedan	rwd	front	103.5	193.8	67.9	53.7	
17	0	gas	std	four	sedan	rwd	front	110.0	197.0	70.9	56.3	
18	2	gas	std	two	hatchback	fwd	front	88.4	141.1	60.3	53.2	
19	1	gas	std	two	hatchback	fwd	front	94.5	155.9	63.6	52.0	
20	0	gas	std	four	sedan	fwd	front	94.5	158.8	63.6	52.0	
21	1	gas	std	two	hatchback	fwd	front	93.7	157.3	63.8	50.8	
22	1	gas	std	two	hatchback	fwd	front	93.7	157.3	63.8	50.8	
23	1	gas	turbo	two	hatchback	fwd	front	93.7	157.3	63.8	50.8	
24	1	gas	std	four	hatchback	fwd	front	93.7	157.3	63.8	50.6	

symboling	fueltype	aspiration	doomnumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbw
25	1	gas	std	four	sedan	fwd	front	93.7	157.3	63.8	50.6
26	1	gas	std	four	sedan	fwd	front	93.7	157.3	63.8	50.6
27	1	gas	turbo	two	sedan	fwd	front	93.7	157.3	63.8	50.6
28	-1	gas	std	four	wagon	fwd	front	103.3	174.6	64.6	59.8
29	3	gas	turbo	two	hatchback	fwd	front	95.9	173.2	66.3	50.2
30	2	gas	std	two	hatchback	fwd	front	86.6	144.6	63.9	50.8
31	2	gas	std	two	hatchback	fwd	front	86.6	144.6	63.9	50.8
32	1	gas	std	two	hatchback	fwd	front	93.7	150.0	64.0	52.6
33	1	gas	std	two	hatchback	fwd	front	93.7	150.0	64.0	52.6
34	1	gas	std	two	hatchback	fwd	front	93.7	150.0	64.0	52.6
35	0	gas	std	four	sedan	fwd	front	96.5	163.4	64.0	54.5
36	0	gas	std	four	wagon	fwd	front	96.5	157.1	63.9	58.3
37	0	gas	std	two	hatchback	fwd	front	96.5	167.5	65.2	53.3
38	0	gas	std	two	hatchback	fwd	front	96.5	167.5	65.2	53.3
39	0	gas	std	four	sedan	fwd	front	96.5	175.4	65.2	54.1
40	0	gas	std	four	sedan	fwd	front	96.5	175.4	62.5	54.1
41	0	gas	std	four	sedan	fwd	front	96.5	175.4	65.2	54.1
42	1	gas	std	two	sedan	fwd	front	96.5	169.1	66.0	51.0
43	0	gas	std	four	sedan	rwd	front	94.3	170.7	61.8	53.5
44	1	gas	std	two	sedan	fwd	front	94.5	155.9	63.6	52.0
45	0	gas	std	four	sedan	fwd	front	94.5	155.9	63.6	52.0
46	2	gas	std	two	hatchback	rwd	front	96.0	172.6	65.2	51.4
47	0	gas	std	four	sedan	rwd	front	113.0	199.6	69.6	52.8
48	0	gas	std	four	sedan	rwd	front	113.0	199.6	69.6	52.8
49	0	gas	std	two	sedan	rwd	front	102.0	191.7	70.6	47.8

symboling	fueltype	aspiration	doomnumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbw
50	1	gas	std	two	hatchback	fwd	front	93.1	159.1	64.2	54.1
51	1	gas	std	two	hatchback	fwd	front	93.1	159.1	64.2	54.1
52	1	gas	std	two	hatchback	fwd	front	93.1	159.1	64.2	54.1
53	1	gas	std	four	sedan	fwd	front	93.1	166.8	64.2	54.1
54	1	gas	std	four	sedan	fwd	front	93.1	166.8	64.2	54.1
55	3	gas	std	two	hatchback	rwd	front	95.3	169.0	65.7	49.6
56	3	gas	std	two	hatchback	rwd	front	95.3	169.0	65.7	49.6
57	3	gas	std	two	hatchback	rwd	front	95.3	169.0	65.7	49.6
58	3	gas	std	two	hatchback	rwd	front	95.3	169.0	65.7	49.6
59	1	gas	std	two	hatchback	fwd	front	98.8	177.8	66.5	53.7
60	0	gas	std	four	sedan	fwd	front	98.8	177.8	66.5	55.5
61	1	gas	std	two	hatchback	fwd	front	98.8	177.8	66.5	53.7
62	0	gas	std	four	sedan	fwd	front	98.8	177.8	66.5	55.5
63	0	diesel	std	four	sedan	fwd	front	98.8	177.8	66.5	55.5
64	0	gas	std	four	hatchback	fwd	front	98.8	177.8	66.5	55.5
65	0	gas	std	four	sedan	rwd	front	104.9	175.0	66.1	54.4
66	0	diesel	std	four	sedan	rwd	front	104.9	175.0	66.1	54.4
67	-1	diesel	turbo	four	sedan	rwd	front	110.0	190.9	70.3	56.5
68	-1	diesel	turbo	four	wagon	rwd	front	110.0	190.9	70.3	58.7
69	0	diesel	turbo	two	hardtop	rwd	front	106.7	187.5	70.3	54.9
70	-1	diesel	turbo	four	sedan	rwd	front	115.6	202.6	71.7	56.3
71	-1	gas	std	four	sedan	rwd	front	115.6	202.6	71.7	56.5
72	3	gas	std	two	convertible	rwd	front	96.6	180.3	70.5	50.8
73	0	gas	std	four	sedan	rwd	front	120.9	208.1	71.7	56.7
74	1	gas	std	two	hardtop	rwd	front	112.0	199.2	72.0	55.4

symboling	fueltype	aspiration	doomnumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbw
75	1	gas	turbo	two	hatchback	rwd	front	102.7	178.4	68.0	54.8
76	2	gas	std	two	hatchback	fwd	front	93.7	157.3	64.4	50.8
77	2	gas	std	two	hatchback	fwd	front	93.7	157.3	64.4	50.8
78	2	gas	std	two	hatchback	fwd	front	93.7	157.3	64.4	50.8
79	1	gas	turbo	two	hatchback	fwd	front	93.0	157.3	63.8	50.8
80	3	gas	turbo	two	hatchback	fwd	front	96.3	173.0	65.4	49.4
81	3	gas	std	two	hatchback	fwd	front	96.3	173.0	65.4	49.4
82	3	gas	turbo	two	hatchback	fwd	front	95.9	173.2	66.3	50.2
83	3	gas	turbo	two	hatchback	fwd	front	95.9	173.2	66.3	50.2
84	3	gas	turbo	two	hatchback	fwd	front	95.9	173.2	66.3	50.2
85	1	gas	std	four	sedan	fwd	front	96.3	172.4	65.4	51.6
86	1	gas	std	four	sedan	fwd	front	96.3	172.4	65.4	51.6
87	1	gas	turbo	four	sedan	fwd	front	96.3	172.4	65.4	51.6
88	-1	gas	std	four	sedan	fwd	front	96.3	172.4	65.4	51.6
89	1	gas	std	two	sedan	fwd	front	94.5	165.3	63.8	54.5
90	1	diesel	std	two	sedan	fwd	front	94.5	165.3	63.8	54.5
91	1	gas	std	two	sedan	fwd	front	94.5	165.3	63.8	54.5
92	1	gas	std	four	sedan	fwd	front	94.5	165.3	63.8	54.5
93	1	gas	std	four	wagon	fwd	front	94.5	170.2	63.8	53.5
94	1	gas	std	two	sedan	fwd	front	94.5	165.3	63.8	54.5
95	1	gas	std	two	hatchback	fwd	front	94.5	165.6	63.8	53.3
96	1	gas	std	four	sedan	fwd	front	94.5	165.3	63.8	54.5
97	1	gas	std	four	wagon	fwd	front	94.5	170.2	63.8	53.5
98	2	gas	std	two	hardtop	fwd	front	95.1	162.4	63.8	53.3
99	0	gas	std	four	hatchback	fwd	front	97.2	173.4	65.2	54.7

symboling	fueltype	aspiration	doomnumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbw
100	0	gas	std	four	sedan	fwd	front	97.2	173.4	65.2	54.7
101	0	gas	std	four	sedan	fwd	front	100.4	181.7	66.5	55.1
102	0	gas	std	four	wagon	fwd	front	100.4	184.6	66.5	56.1
103	0	gas	std	four	sedan	fwd	front	100.4	184.6	66.5	55.1
104	3	gas	std	two	hatchback	rwd	front	91.3	170.7	67.9	49.7
105	3	gas	turbo	two	hatchback	rwd	front	91.3	170.7	67.9	49.7
106	1	gas	std	two	hatchback	rwd	front	99.2	178.5	67.9	49.7
107	0	gas	std	four	sedan	rwd	front	107.9	186.7	68.4	56.7
108	0	diesel	turbo	four	sedan	rwd	front	107.9	186.7	68.4	56.7
109	0	gas	std	four	wagon	rwd	front	114.2	198.9	68.4	58.7
110	0	diesel	turbo	four	wagon	rwd	front	114.2	198.9	68.4	58.7
111	0	gas	std	four	sedan	rwd	front	107.9	186.7	68.4	56.7
112	0	diesel	turbo	four	sedan	rwd	front	107.9	186.7	68.4	56.7
113	0	gas	std	four	wagon	rwd	front	114.2	198.9	68.4	56.7
114	0	diesel	turbo	four	wagon	rwd	front	114.2	198.9	68.4	58.7
115	0	gas	std	four	sedan	rwd	front	107.9	186.7	68.4	56.7
116	0	diesel	turbo	four	sedan	rwd	front	107.9	186.7	68.4	56.7
117	0	gas	turbo	four	sedan	rwd	front	108.0	186.7	68.3	56.0
118	1	gas	std	two	hatchback	fwd	front	93.7	157.3	63.8	50.8
119	1	gas	turbo	two	hatchback	fwd	front	93.7	157.3	63.8	50.8
120	1	gas	std	four	hatchback	fwd	front	93.7	157.3	63.8	50.6
121	1	gas	std	four	sedan	fwd	front	93.7	167.3	63.8	50.8
122	1	gas	std	four	sedan	fwd	front	93.7	167.3	63.8	50.8
123	-1	gas	std	four	wagon	fwd	front	103.3	174.6	64.6	59.8
124	3	gas	turbo	two	hatchback	rwd	front	95.9	173.2	66.3	50.2

symboling	fueltype	aspiration	doomnumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbw
125	3	gas	std	two	hatchback	rwd	front	94.5	168.9	68.3	50.2
126	3	gas	std	two	hardtop	rwd	rear	89.5	168.9	65.0	51.6
127	3	gas	std	two	hardtop	rwd	rear	89.5	168.9	65.0	51.6
128	3	gas	std	two	convertible	rwd	rear	89.5	168.9	65.0	51.6
129	1	gas	std	two	hatchback	rwd	front	98.4	175.7	72.3	50.5
130	0	gas	std	four	wagon	fwd	front	96.1	181.5	66.5	55.2
131	2	gas	std	two	hatchback	fwd	front	96.1	176.8	66.6	50.5
132	3	gas	std	two	hatchback	fwd	front	99.1	186.6	66.5	56.1
133	2	gas	std	four	sedan	fwd	front	99.1	186.6	66.5	56.1
134	3	gas	std	two	hatchback	fwd	front	99.1	186.6	66.5	56.1
135	2	gas	std	four	sedan	fwd	front	99.1	186.6	66.5	56.1
136	3	gas	turbo	two	hatchback	fwd	front	99.1	186.6	66.5	56.1
137	2	gas	turbo	four	sedan	fwd	front	99.1	186.6	66.5	56.1
138	2	gas	std	two	hatchback	fwd	front	93.7	156.9	63.4	53.7
139	2	gas	std	two	hatchback	fwd	front	93.7	157.9	63.6	53.7
140	2	gas	std	two	hatchback	4wd	front	93.3	157.3	63.8	55.7
141	0	gas	std	four	sedan	fwd	front	97.2	172.0	65.4	52.5
142	0	gas	std	four	sedan	fwd	front	97.2	172.0	65.4	52.5
143	0	gas	std	four	sedan	fwd	front	97.2	172.0	65.4	52.5
144	0	gas	std	four	sedan	4wd	front	97.0	172.0	65.4	54.3
145	0	gas	turbo	four	sedan	4wd	front	97.0	172.0	65.4	54.3
146	0	gas	std	four	wagon	fwd	front	97.0	173.5	65.4	53.0
147	0	gas	std	four	wagon	fwd	front	97.0	173.5	65.4	53.0
148	0	gas	std	four	wagon	4wd	front	96.9	173.6	65.4	54.9
149	0	gas	turbo	four	wagon	4wd	front	96.9	173.6	65.4	54.9

symboling	fueltype	aspiration	doomnumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbw
150	1	gas	std	two	hatchback	fwd	front	95.7	158.7	63.6	54.5
151	1	gas	std	two	hatchback	fwd	front	95.7	158.7	63.6	54.5
152	1	gas	std	four	hatchback	fwd	front	95.7	158.7	63.6	54.5
153	0	gas	std	four	wagon	fwd	front	95.7	169.7	63.6	59.1
154	0	gas	std	four	wagon	4wd	front	95.7	169.7	63.6	59.1
155	0	gas	std	four	wagon	4wd	front	95.7	169.7	63.6	59.1
156	0	gas	std	four	sedan	fwd	front	95.7	166.3	64.4	53.0
157	0	gas	std	four	hatchback	fwd	front	95.7	166.3	64.4	52.8
158	0	diesel	std	four	sedan	fwd	front	95.7	166.3	64.4	53.0
159	0	diesel	std	four	hatchback	fwd	front	95.7	166.3	64.4	52.8
160	0	gas	std	four	sedan	fwd	front	95.7	166.3	64.4	53.0
161	0	gas	std	four	hatchback	fwd	front	95.7	166.3	64.4	52.8
162	0	gas	std	four	sedan	fwd	front	95.7	166.3	64.4	52.8
163	1	gas	std	two	sedan	rwd	front	94.5	168.7	64.0	52.6
164	1	gas	std	two	hatchback	rwd	front	94.5	168.7	64.0	52.6
165	1	gas	std	two	sedan	rwd	front	94.5	168.7	64.0	52.6
166	1	gas	std	two	hatchback	rwd	front	94.5	168.7	64.0	52.6
167	2	gas	std	two	hardtop	rwd	front	98.4	176.2	65.6	52.0
168	2	gas	std	two	hardtop	rwd	front	98.4	176.2	65.6	52.0
169	2	gas	std	two	hatchback	rwd	front	98.4	176.2	65.6	52.0
170	2	gas	std	two	hardtop	rwd	front	98.4	176.2	65.6	52.0
171	2	gas	std	two	hatchback	rwd	front	98.4	176.2	65.6	52.0
172	2	gas	std	two	convertible	rwd	front	98.4	176.2	65.6	53.0
173	-1	gas	std	four	sedan	fwd	front	102.4	175.6	66.5	54.9
174	-1	diesel	turbo	four	sedan	fwd	front	102.4	175.6	66.5	54.9

symboling	fueltype	aspiration	doomnumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbw
175	-1	gas	std	four	hatchback	fwd	front	102.4	175.6	66.5	53.9
176	-1	gas	std	four	sedan	fwd	front	102.4	175.6	66.5	54.9
177	-1	gas	std	four	hatchback	fwd	front	102.4	175.6	66.5	53.9
178	3	gas	std	two	hatchback	rwd	front	102.9	183.5	67.7	52.0
179	3	gas	std	two	hatchback	rwd	front	102.9	183.5	67.7	52.0
180	-1	gas	std	four	sedan	rwd	front	104.5	187.8	66.5	54.1
181	-1	gas	std	four	wagon	rwd	front	104.5	187.8	66.5	54.1
182	2	diesel	std	two	sedan	fwd	front	97.3	171.7	65.5	55.7
183	2	gas	std	two	sedan	fwd	front	97.3	171.7	65.5	55.7
184	2	diesel	std	four	sedan	fwd	front	97.3	171.7	65.5	55.7
185	2	gas	std	four	sedan	fwd	front	97.3	171.7	65.5	55.7
186	2	gas	std	four	sedan	fwd	front	97.3	171.7	65.5	55.7
187	2	diesel	turbo	four	sedan	fwd	front	97.3	171.7	65.5	55.7
188	2	gas	std	four	sedan	fwd	front	97.3	171.7	65.5	55.7
189	3	gas	std	two	convertible	fwd	front	94.5	159.3	64.2	55.6
190	3	gas	std	two	hatchback	fwd	front	94.5	165.7	64.0	51.4
191	0	gas	std	four	sedan	fwd	front	100.4	180.2	66.9	55.1
192	0	diesel	turbo	four	sedan	fwd	front	100.4	180.2	66.9	55.1
193	0	gas	std	four	wagon	fwd	front	100.4	183.1	66.9	55.1
194	-2	gas	std	four	sedan	rwd	front	104.3	188.8	67.2	56.2
195	-1	gas	std	four	wagon	rwd	front	104.3	188.8	67.2	57.5
196	-2	gas	std	four	sedan	rwd	front	104.3	188.8	67.2	56.2
197	-1	gas	std	four	wagon	rwd	front	104.3	188.8	67.2	57.5
198	-2	gas	turbo	four	sedan	rwd	front	104.3	188.8	67.2	56.2
199	-1	gas	turbo	four	wagon	rwd	front	104.3	188.8	67.2	57.5

symboling	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbw
200	-1	gas	std	four	sedan	rwd	front	109.1	188.8	68.9	55.5
201	-1	gas	turbo	four	sedan	rwd	front	109.1	188.8	68.8	55.5
202	-1	gas	std	four	sedan	rwd	front	109.1	188.8	68.9	55.5
203	-1	diesel	turbo	four	sedan	rwd	front	109.1	188.8	68.9	55.5
204	-1	gas	turbo	four	sedan	rwd	front	109.1	188.8	68.9	55.5

Categorical encoding

In [859...]

```
doornumber = copy['doornumber'].value_counts().sort_values(ascending=False)
drivewheel = copy['drivewheel'].value_counts().sort_values(ascending=False)
enginetype = copy['enginetype'].value_counts().sort_values(ascending=False)
cylindernumber = copy['cylindernumber'].value_counts().sort_values(ascending=False)
fuelsystem = copy['fuelsystem'].value_counts().sort_values(ascending=False)
drivewheel = copy['aspiration'].value_counts().sort_values(ascending=False)
drivewheel = copy['fueltype'].value_counts().sort_values(ascending=False)
drivewheel = copy['carbody'].value_counts().sort_values(ascending=False)
enginelocation = copy['enginelocation'].value_counts().sort_values(ascending=False)

print(doornumber)
print(drivewheel)
print(enginetype)
print(cylindernumber)
print(fuelsystem)
print(aspiration)
print(fueltype)
print(carbody)
print(enginelocation)
```

```
doornumber
four    115
two     90
Name: count, dtype: int64
carbody
sedan    96
hatchback   70
wagon     25
hardtop     8
convertible   6
Name: count, dtype: int64
enginetype
ohc     148
ohcf    15
ohcv    13
dohc    12
l      12
rotor    4
dohcv   1
Name: count, dtype: int64
cylindernumber
four    159
six     24
five    11
eight    5
two     4
three    1
twelve   1
Name: count, dtype: int64
fuelsystem
mpfi   94
2bbl    66
idi     20
1bbl    11
spdi    9
4bbl    3
mfi     1
spfi    1
Name: count, dtype: int64
aspiration
std     168
turbo   37
Name: count, dtype: int64
fueltype
gas    185
```

```
diesel    20
Name: count, dtype: int64
carbody
sedan     96
hatchback   70
wagon      25
hardtop      8
convertible    6
Name: count, dtype: int64
enginelocation
front    202
rear      3
Name: count, dtype: int64
```

```
In [860...]: columns = ['doornumber', 'drivewheel', 'enginetype', 'cylindernumber', 'fuelsystem', 'aspiration', 'fueltype', 'carbody', 'enginelocation']
```

```
for col in columns:
    if col == 'doornumber':
        conditions = [
            (copy[col] == 'four'), # If doornumber is 'four'
            (copy[col] == 'two')  # If doornumber is 'two'
        ]
        values = [1, 2]

    elif col == 'drivewheel':
        conditions = [
            (copy[col] == 'fwd'), # front-wheel drive
            (copy[col] == 'rwd'), # rear-wheel drive
            (copy[col] == '4wd') # four-wheel drive
        ]
        values = [1, 2, 3]

    elif col == 'enginetype':
        conditions = [
            (copy[col] == 'ohc'), # overhead cam
            (copy[col] == 'ohcf'), # overhead cam front
            (copy[col] == 'ohcv'), # overhead cam vertical
            (copy[col] == 'dohc'), # dual overhead cam
            (copy[col] == 'I'), # inline engine
            (copy[col] == 'rotor'), # rotor engine
            (copy[col] == 'dohcv') # dual overhead cam vertical
        ]
        values = [1, 2, 3, 4, 5, 6, 7]

    elif col == 'cylindernumber':
```

```
conditions = [
    (copy[col] == 'four'),
    (copy[col] == 'six'),
    (copy[col] == 'five'),
    (copy[col] == 'eight'),
    (copy[col] == 'two'),
    (copy[col] == 'three'),
    (copy[col] == 'twelve')
]
values = [1, 2, 3, 4, 5, 6, 7]

elif col == 'fuelsystem':
    conditions = [
        (copy[col] == 'mpfi'), # Multi-point fuel injection
        (copy[col] == '2bbl'), # 2-barrel carburetor
        (copy[col] == 'idi'), # Idi fuel system
        (copy[col] == '1bbl'), # 1-barrel carburetor
        (copy[col] == 'spdi'), # Single-point direct injection
        (copy[col] == '4bbl'), # 4-barrel carburetor
        (copy[col] == 'mfi'), # Mechanical fuel injection
        (copy[col] == 'spfi') # Single-point fuel injection
    ]
    values = [1, 2, 3, 4, 5, 6, 7, 8]

elif col == 'aspiration':
    conditions = [
        (copy[col] == 'std'), # Standard aspiration
        (copy[col] == 'turbo') # Turbocharged
    ]
    values = [1, 2]

elif col == 'fueltype':
    conditions = [
        (copy[col] == 'gas'), # Gas fuel
        (copy[col] == 'diesel') # Diesel fuel
    ]
    values = [1, 2]

elif col == 'carbody':
    conditions = [
        (copy[col] == 'sedan'),
        (copy[col] == 'hatchback'),
        (copy[col] == 'wagon'),
        (copy[col] == 'hardtop'),
        (copy[col] == 'convertible')
    ]
```

```
]
values = [1, 2, 3, 4, 5]

elif col =='enginelocation':
    conditions = [
        (copy[col]=='front'),
        (copy[col]=='rear')
    ]
    values = [1,2]

copy[col] = np.select(conditions, values, default=0)

copy
```

Out[860]:

	symboling	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbwei
0	3	1	1	2	5	2	1	88.6	168.8	64.1	48.8	25
1	3	1	1	2	5	2	1	88.6	168.8	64.1	48.8	25
2	1	1	1	2	2	2	1	94.5	171.2	65.5	52.4	28
3	2	1	1	1	1	1	1	99.8	176.6	66.2	54.3	25
4	2	1	1	1	1	3	1	99.4	176.6	66.4	54.3	28
5	2	1	1	2	1	1	1	99.8	177.3	66.3	53.1	25
6	1	1	1	1	1	1	1	105.8	192.7	71.4	55.7	28
7	1	1	1	1	3	1	1	105.8	192.7	71.4	55.7	29
8	1	1	2	1	1	1	1	105.8	192.7	71.4	55.9	30
9	0	1	2	2	2	3	1	99.5	178.2	67.9	52.0	30
10	2	1	1	2	1	2	1	101.2	176.8	64.8	54.3	25
11	0	1	1	1	1	2	1	101.2	176.8	64.8	54.3	25
12	0	1	1	2	1	2	1	101.2	176.8	64.8	54.3	27
13	0	1	1	1	1	2	1	101.2	176.8	64.8	54.3	27
14	1	1	1	1	1	2	1	103.5	189.0	66.9	55.7	30
15	0	1	1	1	1	2	1	103.5	189.0	66.9	55.7	32
16	0	1	1	2	1	2	1	103.5	193.8	67.9	53.7	33
17	0	1	1	1	1	2	1	110.0	197.0	70.9	56.3	35
18	2	1	1	2	2	1	1	88.4	141.1	60.3	53.2	14
19	1	1	1	2	2	1	1	94.5	155.9	63.6	52.0	18
20	0	1	1	1	1	1	1	94.5	158.8	63.6	52.0	19
21	1	1	1	2	2	1	1	93.7	157.3	63.8	50.8	18
22	1	1	1	2	2	1	1	93.7	157.3	63.8	50.8	18
23	1	1	2	2	2	1	1	93.7	157.3	63.8	50.8	21
24	1	1	1	1	2	1	1	93.7	157.3	63.8	50.6	19

symboling	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbwei	
25	1	1	1	1	1	1	93.7	157.3	63.8	50.6	19	
26	1	1	1	1	1	1	93.7	157.3	63.8	50.6	19	
27	1	1	2	2	1	1	93.7	157.3	63.8	50.6	21	
28	-1	1	1	1	3	1	103.3	174.6	64.6	59.8	21	
29	3	1	2	2	2	1	95.9	173.2	66.3	50.2	28	
30	2	1	1	2	2	1	86.6	144.6	63.9	50.8	17	
31	2	1	1	2	2	1	86.6	144.6	63.9	50.8	18	
32	1	1	1	2	2	1	93.7	150.0	64.0	52.6	18	
33	1	1	1	2	2	1	93.7	150.0	64.0	52.6	19	
34	1	1	1	2	2	1	93.7	150.0	64.0	52.6	19	
35	0	1	1	1	1	1	96.5	163.4	64.0	54.5	20	
36	0	1	1	1	3	1	96.5	157.1	63.9	58.3	20	
37	0	1	1	2	2	1	96.5	167.5	65.2	53.3	22	
38	0	1	1	2	2	1	96.5	167.5	65.2	53.3	22	
39	0	1	1	1	1	1	96.5	175.4	65.2	54.1	23	
40	0	1	1	1	1	1	96.5	175.4	62.5	54.1	23	
41	0	1	1	1	1	1	96.5	175.4	65.2	54.1	24	
42	1	1	1	2	1	1	96.5	169.1	66.0	51.0	22	
43	0	1	1	1	1	2	1	94.3	170.7	61.8	53.5	23
44	1	1	1	2	1	1	94.5	155.9	63.6	52.0	18	
45	0	1	1	1	1	1	94.5	155.9	63.6	52.0	19	
46	2	1	1	2	2	2	1	96.0	172.6	65.2	51.4	27
47	0	1	1	1	1	2	1	113.0	199.6	69.6	52.8	40
48	0	1	1	1	1	2	1	113.0	199.6	69.6	52.8	40
49	0	1	1	2	1	2	1	102.0	191.7	70.6	47.8	39

symboling	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbwei
50	1	1	1	2	2	1	1	93.1	159.1	64.2	54.1
51	1	1	1	2	2	1	1	93.1	159.1	64.2	54.1
52	1	1	1	2	2	1	1	93.1	159.1	64.2	54.1
53	1	1	1	1	1	1	1	93.1	166.8	64.2	54.1
54	1	1	1	1	1	1	1	93.1	166.8	64.2	54.1
55	3	1	1	2	2	2	1	95.3	169.0	65.7	49.6
56	3	1	1	2	2	2	1	95.3	169.0	65.7	49.6
57	3	1	1	2	2	2	1	95.3	169.0	65.7	49.6
58	3	1	1	2	2	2	1	95.3	169.0	65.7	49.6
59	1	1	1	2	2	1	1	98.8	177.8	66.5	53.7
60	0	1	1	1	1	1	1	98.8	177.8	66.5	55.5
61	1	1	1	2	2	1	1	98.8	177.8	66.5	53.7
62	0	1	1	1	1	1	1	98.8	177.8	66.5	55.5
63	0	2	1	1	1	1	1	98.8	177.8	66.5	55.5
64	0	1	1	1	2	1	1	98.8	177.8	66.5	55.5
65	0	1	1	1	1	2	1	104.9	175.0	66.1	54.4
66	0	2	1	1	1	2	1	104.9	175.0	66.1	54.4
67	-1	2	2	1	1	2	1	110.0	190.9	70.3	56.5
68	-1	2	2	1	3	2	1	110.0	190.9	70.3	58.7
69	0	2	2	2	4	2	1	106.7	187.5	70.3	54.9
70	-1	2	2	1	1	2	1	115.6	202.6	71.7	56.3
71	-1	1	1	1	1	2	1	115.6	202.6	71.7	56.5
72	3	1	1	2	5	2	1	96.6	180.3	70.5	50.8
73	0	1	1	1	1	2	1	120.9	208.1	71.7	56.7
74	1	1	1	2	4	2	1	112.0	199.2	72.0	55.4

symboling	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbwei
75	1	1	2	2	2	2	102.7	178.4	68.0	54.8	29
76	2	1	1	2	2	1	93.7	157.3	64.4	50.8	19
77	2	1	1	2	2	1	93.7	157.3	64.4	50.8	19
78	2	1	1	2	2	1	93.7	157.3	64.4	50.8	20
79	1	1	2	2	2	1	93.0	157.3	63.8	50.8	21
80	3	1	2	2	2	1	96.3	173.0	65.4	49.4	23
81	3	1	1	2	2	1	96.3	173.0	65.4	49.4	23
82	3	1	2	2	2	1	95.9	173.2	66.3	50.2	28
83	3	1	2	2	2	1	95.9	173.2	66.3	50.2	29
84	3	1	2	2	2	1	95.9	173.2	66.3	50.2	29
85	1	1	1	1	1	1	96.3	172.4	65.4	51.6	23
86	1	1	1	1	1	1	96.3	172.4	65.4	51.6	24
87	1	1	2	1	1	1	96.3	172.4	65.4	51.6	24
88	-1	1	1	1	1	1	96.3	172.4	65.4	51.6	24
89	1	1	1	2	1	1	94.5	165.3	63.8	54.5	18
90	1	2	1	2	1	1	94.5	165.3	63.8	54.5	20
91	1	1	1	2	1	1	94.5	165.3	63.8	54.5	19
92	1	1	1	1	1	1	94.5	165.3	63.8	54.5	19
93	1	1	1	1	3	1	94.5	170.2	63.8	53.5	20
94	1	1	1	2	1	1	94.5	165.3	63.8	54.5	19
95	1	1	1	2	2	1	94.5	165.6	63.8	53.3	20
96	1	1	1	1	1	1	94.5	165.3	63.8	54.5	19
97	1	1	1	1	3	1	94.5	170.2	63.8	53.5	20
98	2	1	1	2	4	1	95.1	162.4	63.8	53.3	20
99	0	1	1	1	2	1	97.2	173.4	65.2	54.7	23

symboling	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbwei
100	0	1	1	1	1	1	97.2	173.4	65.2	54.7	23
101	0	1	1	1	1	1	100.4	181.7	66.5	55.1	30
102	0	1	1	1	3	1	100.4	184.6	66.5	56.1	32
103	0	1	1	1	1	1	100.4	184.6	66.5	55.1	30
104	3	1	1	2	2	2	91.3	170.7	67.9	49.7	30
105	3	1	2	2	2	2	1	91.3	170.7	67.9	49.7
106	1	1	1	2	2	2	1	99.2	178.5	67.9	49.7
107	0	1	1	1	1	2	1	107.9	186.7	68.4	56.7
108	0	2	2	1	1	2	1	107.9	186.7	68.4	56.7
109	0	1	1	1	3	2	1	114.2	198.9	68.4	58.7
110	0	2	2	1	3	2	1	114.2	198.9	68.4	58.7
111	0	1	1	1	1	2	1	107.9	186.7	68.4	56.7
112	0	2	2	1	1	2	1	107.9	186.7	68.4	56.7
113	0	1	1	1	3	2	1	114.2	198.9	68.4	56.7
114	0	2	2	1	3	2	1	114.2	198.9	68.4	58.7
115	0	1	1	1	1	2	1	107.9	186.7	68.4	56.7
116	0	2	2	1	1	2	1	107.9	186.7	68.4	56.7
117	0	1	2	1	1	2	1	108.0	186.7	68.3	56.0
118	1	1	1	2	2	1	1	93.7	157.3	63.8	50.8
119	1	1	2	2	2	1	1	93.7	157.3	63.8	50.8
120	1	1	1	1	2	1	1	93.7	157.3	63.8	50.6
121	1	1	1	1	1	1	1	93.7	167.3	63.8	50.8
122	1	1	1	1	1	1	1	93.7	167.3	63.8	50.8
123	-1	1	1	1	3	1	1	103.3	174.6	64.6	59.8
124	3	1	2	2	2	2	1	95.9	173.2	66.3	50.2

symboling	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbwei
125	3	1	1	2	2	2	1	94.5	168.9	68.3	50.2
126	3	1	1	2	4	2	2	89.5	168.9	65.0	51.6
127	3	1	1	2	4	2	2	89.5	168.9	65.0	51.6
128	3	1	1	2	5	2	2	89.5	168.9	65.0	51.6
129	1	1	1	2	2	2	1	98.4	175.7	72.3	50.5
130	0	1	1	1	3	1	1	96.1	181.5	66.5	55.2
131	2	1	1	2	2	1	1	96.1	176.8	66.6	50.5
132	3	1	1	2	2	1	1	99.1	186.6	66.5	56.1
133	2	1	1	1	1	1	1	99.1	186.6	66.5	56.1
134	3	1	1	2	2	1	1	99.1	186.6	66.5	56.1
135	2	1	1	1	1	1	1	99.1	186.6	66.5	56.1
136	3	1	2	2	2	1	1	99.1	186.6	66.5	56.1
137	2	1	2	1	1	1	1	99.1	186.6	66.5	56.1
138	2	1	1	2	2	1	1	93.7	156.9	63.4	53.7
139	2	1	1	2	2	1	1	93.7	157.9	63.6	53.7
140	2	1	1	2	2	3	1	93.3	157.3	63.8	55.7
141	0	1	1	1	1	1	1	97.2	172.0	65.4	52.5
142	0	1	1	1	1	1	1	97.2	172.0	65.4	52.5
143	0	1	1	1	1	1	1	97.2	172.0	65.4	52.5
144	0	1	1	1	1	3	1	97.0	172.0	65.4	54.3
145	0	1	2	1	1	3	1	97.0	172.0	65.4	54.3
146	0	1	1	1	3	1	1	97.0	173.5	65.4	53.0
147	0	1	1	1	3	1	1	97.0	173.5	65.4	53.0
148	0	1	1	1	3	3	1	96.9	173.6	65.4	54.9
149	0	1	2	1	3	3	1	96.9	173.6	65.4	54.9

symboling	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbwei
150	1	1	1	2	2	1	1	95.7	158.7	63.6	54.5
151	1	1	1	2	2	1	1	95.7	158.7	63.6	54.5
152	1	1	1	1	2	1	1	95.7	158.7	63.6	54.5
153	0	1	1	1	3	1	1	95.7	169.7	63.6	59.1
154	0	1	1	1	3	3	1	95.7	169.7	63.6	59.1
155	0	1	1	1	3	3	1	95.7	169.7	63.6	59.1
156	0	1	1	1	1	1	1	95.7	166.3	64.4	53.0
157	0	1	1	1	2	1	1	95.7	166.3	64.4	52.8
158	0	2	1	1	1	1	1	95.7	166.3	64.4	53.0
159	0	2	1	1	2	1	1	95.7	166.3	64.4	52.8
160	0	1	1	1	1	1	1	95.7	166.3	64.4	53.0
161	0	1	1	1	2	1	1	95.7	166.3	64.4	52.8
162	0	1	1	1	1	1	1	95.7	166.3	64.4	52.8
163	1	1	1	2	1	2	1	94.5	168.7	64.0	52.6
164	1	1	1	2	2	2	1	94.5	168.7	64.0	52.6
165	1	1	1	2	1	2	1	94.5	168.7	64.0	52.6
166	1	1	1	2	2	2	1	94.5	168.7	64.0	52.6
167	2	1	1	2	4	2	1	98.4	176.2	65.6	52.0
168	2	1	1	2	4	2	1	98.4	176.2	65.6	52.0
169	2	1	1	2	2	2	1	98.4	176.2	65.6	52.0
170	2	1	1	2	4	2	1	98.4	176.2	65.6	52.0
171	2	1	1	2	2	2	1	98.4	176.2	65.6	52.0
172	2	1	1	2	5	2	1	98.4	176.2	65.6	53.0
173	-1	1	1	1	1	1	1	102.4	175.6	66.5	54.9
174	-1	2	2	1	1	1	1	102.4	175.6	66.5	54.9

symboling	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbwei
175	-1	1	1	1	2	1	1	102.4	175.6	66.5	53.9
176	-1	1	1	1	1	1	1	102.4	175.6	66.5	54.9
177	-1	1	1	1	2	1	1	102.4	175.6	66.5	53.9
178	3	1	1	2	2	2	1	102.9	183.5	67.7	52.0
179	3	1	1	2	2	2	1	102.9	183.5	67.7	52.0
180	-1	1	1	1	1	2	1	104.5	187.8	66.5	54.1
181	-1	1	1	1	3	2	1	104.5	187.8	66.5	54.1
182	2	2	1	2	1	1	1	97.3	171.7	65.5	55.7
183	2	1	1	2	1	1	1	97.3	171.7	65.5	55.7
184	2	2	1	1	1	1	1	97.3	171.7	65.5	55.7
185	2	1	1	1	1	1	1	97.3	171.7	65.5	55.7
186	2	1	1	1	1	1	1	97.3	171.7	65.5	55.7
187	2	2	2	1	1	1	1	97.3	171.7	65.5	55.7
188	2	1	1	1	1	1	1	97.3	171.7	65.5	55.7
189	3	1	1	2	5	1	1	94.5	159.3	64.2	55.6
190	3	1	1	2	2	1	1	94.5	165.7	64.0	51.4
191	0	1	1	1	1	1	1	100.4	180.2	66.9	55.1
192	0	2	2	1	1	1	1	100.4	180.2	66.9	55.1
193	0	1	1	1	3	1	1	100.4	183.1	66.9	55.1
194	-2	1	1	1	1	2	1	104.3	188.8	67.2	56.2
195	-1	1	1	1	3	2	1	104.3	188.8	67.2	57.5
196	-2	1	1	1	1	2	1	104.3	188.8	67.2	56.2
197	-1	1	1	1	3	2	1	104.3	188.8	67.2	57.5
198	-2	1	2	1	1	2	1	104.3	188.8	67.2	56.2
199	-1	1	2	1	3	2	1	104.3	188.8	67.2	57.5

symboling	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	carwidth	carheight	curbwei
200	-1	1	1	1	1	2	1	109.1	188.8	68.9	55.5
201	-1	1	2	1	1	2	1	109.1	188.8	68.8	55.5
202	-1	1	1	1	1	2	1	109.1	188.8	68.9	55.5
203	-1	2	2	1	1	2	1	109.1	188.8	68.9	55.5
204	-1	1	2	1	1	2	1	109.1	188.8	68.9	55.5

detecting outliers

In [861...]

```
prices = df['price']
plt.boxplot(prices)
plt.title('prices outliers')
plt.show()
```



In [862]:

```
df_filt = copy.copy()

for column in df_filt.columns:
    Q1 = df_filt[column].quantile(0.25)
    Q3 = df_filt[column].quantile(0.75)
    IQR = Q3 - Q1

    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    df_filt = df_filt[(df_filt[column] >= lower_bound) & (df_filt[column] <= upper_bound)]
```

df_filt.shape

Out[862]: (65, 25)

model building

1.2 - linear model (original data)

In [863...]

```
# Assuming df_filt is your DataFrame
columns = df_filt.columns
x = copy[['enginesize','curbweight','horsepower','highwaympg','citympg']]
y = copy['price']

# Step 1: Split the data into training, validation, and test sets
x_temp, x_test, y_temp, y_test = train_test_split(x, y, test_size=0.2, random_state=0) # 20% for testing
x_train, x_val, y_train, y_val = train_test_split(x_temp, y_temp, test_size=0.25, random_state=0) # 25% of the remaining 80%

# Step 2: Train the model on the training set
model = LinearRegression()
model.fit(x_train, y_train)

# Step 3: Validate the model using the validation set
y_val_pred = model.predict(x_val)
mse_val = mean_squared_error(y_val, y_val_pred)
R_val = model.score(x_val, y_val) # R^2 for validation set

# Calculate Adjusted R^2 for validation set
n_val = len(y_val)
p = x_val.shape[1]
adjusted_r_squared_val = 1 - ((1 - R_val**2) * (n_val - 1)) / (n_val - p - 1)

# Print results for validation
print("Validation Results:")
print(f"Mean Squared Error: {mse_val}")
print(f"R^2: {R_val}")
print(f"Adjusted R^2: {adjusted_r_squared_val}")
#####
#####
```



```
# Step 4: Test the model using the test set
y_test_pred = model.predict(x_test)
mse_test = mean_squared_error(y_test, y_test_pred)
R_test = model.score(x_test, y_test) # R^2 for test set
```

```

# Calculate Adjusted R^2 for test set
n_test = len(y_test)
adjusted_r_squared_test = 1 - ((1 - R_test**2) * (n_test - 1)) / (n_test - p - 1)

# Print results for test
print("\nTest Results:")
print(f"Mean Squared Error: {mse_test}")
print(f"R^2: {R_test}")
print(f"Adjusted R^2: {adjusted_r_squared_test}")

#####
# Print coefficients
print("\nModel Coefficients:")
for i, feature in enumerate(x.columns):
    print(f"Feature: {feature} : {model.coef_[i]}")

print(f"Intercept: {model.intercept_}")

```

Validation Results:

Mean Squared Error: 10302324.649325434
 R^2: 0.7416415132972272
 Adjusted R^2: 0.48575101056662995

Test Results:

Mean Squared Error: 15214876.35402552
 R^2: 0.8034669066922782
 Adjusted R^2: 0.5949246515996092

Model Coefficients:

Feature: enginesize : 67.58380223548275
 Feature: curbweight : 5.326863850732126
 Feature: horsepower : 60.303599767195294
 Feature: highwaympg : 49.321179065064214
 Feature: citympg : -34.477680492973015
 Intercept: -15740.789129925106

In [864...]

```

data = {
    'enginesize': [150], # Example engine sizes
    'curbweight': [3000], # Example curb weights
    'horsepower': [150], # Example horsepower values
    'highwaympg': [31], # Example highway MPG
    'citympg': [22] # Example city MPG
}

```

```
}
```

```
df_new = pd.DataFrame(data)
```

```
predicted_prices = model.predict(df_new)
```

```
# Print the predicted prices
```

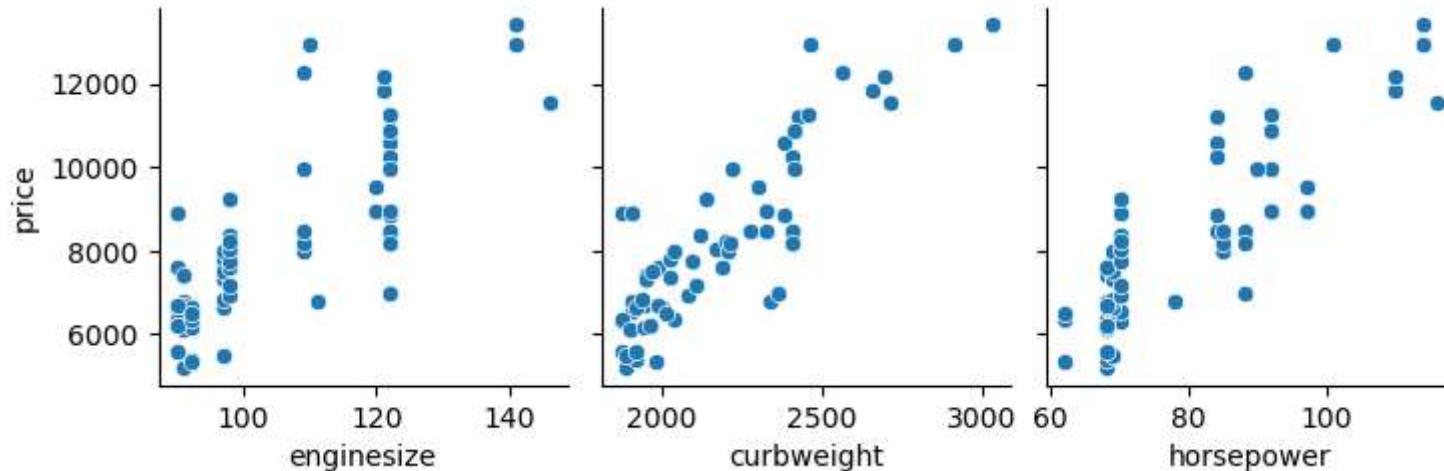
```
print("Predicted Prices:",predicted_prices[0],"$$$$")
```

Predicted Prices: 20193.360302844565 \$\$\$\$\pmb{}

```
In [865...]
```

```
sns.pairplot(df_filt, x_vars=['enginesize', 'curbweight', 'horsepower'], y_vars='price', kind='scatter')
```

```
plt.show()
```



1.3 - linear model (filtered data)

```
In [866...]
```

```
# Assuming df_filt is your DataFrame
```

```
columns = df_filt.columns
```

```
x = df_filt[['enginesize','curbweight','horsepower','highwaympg','citympg']]
```

```
y = df_filt['price']
```

```
# Step 1: Split the data into training, validation, and test sets
```

```
x_temp,x_test,y_temp,y_test = train_test_split(x,y,test_size=0.2,random_state=0) # 20% for testing
```

```
x_train,x_val,y_train,y_val = train_test_split(x_temp,y_temp,test_size=0.25,random_state=0) # 25% of the remaining 80%
```

```

# Step 2: Train the model on the training set
model = LinearRegression()
model.fit(x_train, y_train)

# Step 3: Validate the model using the validation set
y_val_pred = model.predict(x_val)
mse_val = mean_squared_error(y_val, y_val_pred)
R_val = model.score(x_val, y_val) # R^2 for validation set

# Calculate Adjusted R^2 for validation set
n_val = len(y_val)
p = x_val.shape[1]
adjusted_r_squared_val = 1 - ((1 - R_val**2) * (n_val - 1)) / (n_val - p - 1)

# Print results for validation
print("Validation Results:")
print(f"Mean Squared Error: {mse_val}")
print(f"R^2: {R_val}")
print(f"Adjusted R^2: {adjusted_r_squared_val}")
#####
# Step 4: Test the model using the test set
y_test_pred = model.predict(x_test)
mse_test = mean_squared_error(y_test, y_test_pred)
R_test = model.score(x_test, y_test) # R^2 for test set

# Calculate Adjusted R^2 for test set
n_test = len(y_test)
adjusted_r_squared_test = 1 - ((1 - R_test**2) * (n_test - 1)) / (n_test - p - 1)

# Print results for test
print("\nTest Results:")
print(f"Mean Squared Error: {mse_test}")
print(f"R^2: {R_test}")
print(f"Adjusted R^2: {adjusted_r_squared_test}")
#####
# Print coefficients
print("\nModel Coefficients:")
for i, feature in enumerate(x.columns):
    print(f"Feature: {feature} : {model.coef_[i]}")

print(f"Intercept: {model.intercept_}")

```

Validation Results:

Mean Squared Error: 1230178.0625439766

R^2: 0.7658167509233005

Adjusted R^2: 0.29110050741952087

Test Results:

Mean Squared Error: 1070899.0327989294

R^2: 0.800675579706682

Adjusted R^2: 0.38471094389479654

Model Coefficients:

Feature: enginesize : -42.2885696232706

Feature: curbweight : 5.0343545325627845

Feature: horsepower : 84.85688648101201

Feature: highwaympg : -127.06069148082915

Feature: citympg : 157.38859307147953

Intercept: -5034.357824559029

Tune the Model

2 - polynomial model (filtered data)

In [867...]

```
# Assuming df_filt is your DataFrame
columns = df_filt.columns
x = df_filt.drop(columns=['price'])
y = df_filt['price']

# Step 1: Split the data into training, validation, and test sets
x_temp, x_test, y_temp, y_test = train_test_split(x, y, test_size=0.2, random_state=0) # 20% for testing
x_train, x_val, y_train, y_val = train_test_split(x_temp, y_temp, test_size=0.25, random_state=0) # 25% of remaining

# Step 2: Polynomial Feature Transformation
poly = PolynomialFeatures(degree=2) # Adjust degree as needed
x_train_poly = poly.fit_transform(x_train)
x_val_poly = poly.transform(x_val)
```

```

x_test_poly = poly.transform(x_test)

# Function to calculate adjusted R^2
def adjusted_r_squared(r_squared, n, p):
    return 1 - (1 - r_squared) * (n - 1) / (n - p - 1)

#####
# Step 3: Apply Ridge Regression

# Ridge on Validation Set

scaler = StandardScaler()
x_train_poly_scaled = scaler.fit_transform(x_train_poly)
x_val_poly_scaled = scaler.transform(x_val_poly)
x_test_poly_scaled = scaler.transform(x_test_poly)

ridge_model = Ridge(alpha=0.10) # You can adjust alpha for regularization strength
ridge_model.fit(x_train_poly_scaled, y_train)

# Validation Evaluation

y_val_pred = ridge_model.predict(x_val_poly_scaled)
mse_val = mean_squared_error(y_val, y_val_pred)
r_squared_val = ridge_model.score(x_val_poly_scaled, y_val)
n_val = len(y_val)
p_val = x_val_poly_scaled.shape[1]
adjusted_r2_val = adjusted_r_squared(r_squared_val, n_val, p_val)

print("\nRidge Regression - Validation Results:")
print(f"Mean Squared Error: {mse_val}")
print(f"R^2: {r_squared_val}")
print(f"Adjusted R^2: {adjusted_r2_val}")

# Test Set Evaluation

y_test_pred = ridge_model.predict(x_test_poly_scaled)
mse_test = mean_squared_error(y_test, y_test_pred)
r_squared_test = ridge_model.score(x_test_poly_scaled, y_test)
n_test = len(y_test)
p_test = x_test_poly_scaled.shape[1]

```

```

adjusted_r2_test = adjusted_r_squared(r_squared_test, n_test, p_test)

print("\nRidge Regression - Test Results:")
print(f"Mean Squared Error: {mse_test}")
print(f"R^2: {r_squared_test}")
print(f"Adjusted R^2: {adjusted_r2_test}")

#####
# Step 4: Apply Lasso Regression

# Lasso on Validation Set
lasso_model = Lasso(alpha=100) # You can adjust alpha for regularization strength
lasso_model.fit(x_train_poly, y_train)

# Validation Evaluation
y_val_pred_lasso = lasso_model.predict(x_val_poly)
mse_val_lasso = mean_squared_error(y_val, y_val_pred_lasso)
r_squared_val_lasso = lasso_model.score(x_val_poly, y_val)
adjusted_r2_val_lasso = adjusted_r_squared(r_squared_val_lasso, n_val, p_val)

print("\nLasso Regression - Validation Results:")
print(f"Mean Squared Error: {mse_val_lasso}")
print(f"R^2: {r_squared_val_lasso}")
print(f"Adjusted R^2: {adjusted_r2_val_lasso}")

# Test Set Evaluation
y_test_pred_lasso = lasso_model.predict(x_test_poly)
mse_test_lasso = mean_squared_error(y_test, y_test_pred_lasso)
r_squared_test_lasso = lasso_model.score(x_test_poly, y_test)
adjusted_r2_test_lasso = adjusted_r_squared(r_squared_test_lasso, n_test, p_test)

print("\nLasso Regression - Test Results:")
print(f"Mean Squared Error: {mse_test_lasso}")
print(f"R^2: {r_squared_test_lasso}")
print(f"Adjusted R^2: {adjusted_r2_test_lasso}")

```

Ridge Regression - Validation Results:
Mean Squared Error: 1478614.025686725
 R^2 : 0.7185231575747456
Adjusted R^2 : 1.0107914444380288

Ridge Regression - Test Results:
Mean Squared Error: 1969000.9563929364
 R^2 : 0.6335135599438901
Adjusted R^2 : 1.0140505983408092

Lasso Regression - Validation Results:
Mean Squared Error: 2866982.267499875
 R^2 : 0.4542259833019884
Adjusted R^2 : 1.020924243451681

Lasso Regression - Test Results:
Mean Squared Error: 2813510.8495312077
 R^2 : 0.47632652388708985
Adjusted R^2 : 1.0200769383813257

In [868...]

```
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures, StandardScaler
from sklearn.linear_model import Ridge, Lasso

# Assuming `poly` and `scaler` are already fitted, and ridge_model and lasso_model are trained

# Example new input data (replace this with your actual data)
new_data = pd.DataFrame({
    'symboling': [3],
    'fueltype': [1],
    'aspiration': [2],
    'doornumber': [2],
    'carbody': [2],
    'drivewheel': [2],
    'enginelocation': [1],
    'wheelbase': [98.4],
    'carlength': [168.8],
    'carwidth': [64.1],
    'carheight': [54.3],
    'curbweight': [3000],
    'enginetype': [4],
    'cylindernumber': [4],
    'enginesize': [150],
    'fuelsystem': [3],
    'boreratio': [3.19],
```

```

'stroke': [3.4],
'compressionratio': [9.0],
'horsepower': [150],
'peakrpm': [5000],
'citympg': [22],
'highwaympg': [31],
'hp_weight_ratio': [0.0433]
})

# Step 1: Preprocess the new data
new_data_poly = poly.transform(new_data)
new_data_poly_scaled = scaler.transform(new_data_poly)

# Step 2: Make predictions
ridge_prediction = ridge_model.predict(new_data_poly_scaled)
lasso_prediction = lasso_model.predict(new_data_poly)

# Output the predictions
print(f"Predicted price using Ridge Regression: {ridge_prediction[0]}")
print(f"Predicted price using Lasso Regression: {lasso_prediction[0]}")

```

Predicted price using Ridge Regression: 76304.59416261516

Predicted price using Lasso Regression: 16976.892222935472

2.1 - polynomial model (original data)

In [869...]

```

# Assuming df_filt is your DataFrame
columns = df_filt.columns
x = copy.drop(columns=['price'])
y = copy['price']

# Step 1: Split the data into training, validation, and test sets
x_temp, x_test, y_temp, y_test = train_test_split(x, y, test_size=0.2, random_state=0) # 20% for testing
x_train, x_val, y_train, y_val = train_test_split(x_temp, y_temp, test_size=0.25, random_state=0) # 25% of remaining

# Step 2: Polynomial Feature Transformation
poly = PolynomialFeatures(degree=2) # Adjust degree as needed

```

```

x_train_poly = poly.fit_transform(x_train)
x_val_poly = poly.transform(x_val)
x_test_poly = poly.transform(x_test)

# Function to calculate adjusted R^2
def adjusted_r_squared(r_squared, n, p):
    return 1 - (1 - r_squared) * (n - 1) / (n - p - 1)

#####
# Step 3: Apply Ridge Regression

# Ridge on Validation Set

scaler = StandardScaler()
x_train_poly_scaled = scaler.fit_transform(x_train_poly)
x_val_poly_scaled = scaler.transform(x_val_poly)
x_test_poly_scaled = scaler.transform(x_test_poly)

ridge_model = Ridge(alpha=0.10) # You can adjust alpha for regularization strength
ridge_model.fit(x_train_poly_scaled, y_train)

# Validation Evaluation
y_val_pred = ridge_model.predict(x_val_poly_scaled)
mse_val = mean_squared_error(y_val, y_val_pred)
r_squared_val = ridge_model.score(x_val_poly_scaled, y_val)
n_val = len(y_val)
p_val = x_val_poly_scaled.shape[1]
adjusted_r2_val = adjusted_r_squared(r_squared_val, n_val, p_val)

print("\nRidge Regression - Validation Results:")
print(f"Mean Squared Error: {mse_val}")
print(f"R^2: {r_squared_val}")
print(f"Adjusted R^2: {adjusted_r2_val}")

# Test Set Evaluation
y_test_pred = ridge_model.predict(x_test_poly_scaled)
mse_test = mean_squared_error(y_test, y_test_pred)
r_squared_test = ridge_model.score(x_test_poly_scaled, y_test)
n_test = len(y_test)
p_test = x_test_poly_scaled.shape[1]

```

```

adjusted_r2_test = adjusted_r_squared(r_squared_test, n_test, p_test)

print("\nRidge Regression - Test Results:")
print(f"Mean Squared Error: {mse_test}")
print(f"R^2: {r_squared_test}")
print(f"Adjusted R^2: {adjusted_r2_test}")

#####
# Step 4: Apply Lasso Regression

# Lasso on Validation Set
lasso_model = Lasso(alpha=100) # You can adjust alpha for regularization strength
lasso_model.fit(x_train_poly, y_train)

# Validation Evaluation
y_val_pred_lasso = lasso_model.predict(x_val_poly)
mse_val_lasso = mean_squared_error(y_val, y_val_pred_lasso)
r_squared_val_lasso = lasso_model.score(x_val_poly, y_val)
adjusted_r2_val_lasso = adjusted_r_squared(r_squared_val_lasso, n_val, p_val)

print("\nLasso Regression - Validation Results:")
print(f"Mean Squared Error: {mse_val_lasso}")
print(f"R^2: {r_squared_val_lasso}")
print(f"Adjusted R^2: {adjusted_r2_val_lasso}")



# Test Set Evaluation
y_test_pred_lasso = lasso_model.predict(x_test_poly)
mse_test_lasso = mean_squared_error(y_test, y_test_pred_lasso)
r_squared_test_lasso = lasso_model.score(x_test_poly, y_test)
adjusted_r2_test_lasso = adjusted_r_squared(r_squared_test_lasso, n_test, p_test)

print("\nLasso Regression - Test Results:")
print(f"Mean Squared Error: {mse_test_lasso}")
print(f"R^2: {r_squared_test_lasso}")
print(f"Adjusted R^2: {adjusted_r2_test_lasso}")

```

Ridge Regression - Validation Results:
Mean Squared Error: 7348360.94900899
 R^2 : 0.8157200943326879
Adjusted R^2 : 1.0258638464094474

Ridge Regression - Test Results:
Mean Squared Error: 22664467.94766678
 R^2 : 0.7072392906597529
Adjusted R^2 : 1.0410892223635435

Lasso Regression - Validation Results:
Mean Squared Error: 6228650.551532028
 R^2 : 0.8437998427083567
Adjusted R^2 : 1.021922829093564

Lasso Regression - Test Results:
Mean Squared Error: 11753193.648127817
 R^2 : 0.8481820390673053
Adjusted R^2 : 1.0213077839905536

In [870...]

```
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures, StandardScaler
from sklearn.linear_model import Ridge, Lasso

# Assuming `poly` and `scaler` are already fitted, and ridge_model and lasso_model are trained

# Example new input data (replace this with your actual data)
new_data = pd.DataFrame({
    'symboling': [3],
    'fueltype': [1],
    'aspiration': [2],
    'doornumber': [2],
    'carbody': [2],
    'drivewheel': [2],
    'enginelocation': [1],
    'wheelbase': [98.4],
    'carlength': [168.8],
    'carwidth': [64.1],
    'carheight': [54.3],
    'curbweight': [3000],
    'enginetype': [4],
    'cylindernumber': [4],
    'enginesize': [150],
    'fuelsystem': [3],
    'boreratio': [3.19],
```

```
'stroke': [3.4],  
'compressionratio': [9.0],  
'horsepower': [150],  
'peakrpm': [5000],  
'citympg': [22],  
'highwaympg': [31],  
'hp_weight_ratio': [0.0433]  
})  
  
# Step 1: Preprocess the new data  
new_data_poly = poly.transform(new_data)  
new_data_poly_scaled = scaler.transform(new_data_poly)  
  
# Step 2: Make predictions  
ridge_prediction = ridge_model.predict(new_data_poly_scaled)  
lasso_prediction = lasso_model.predict(new_data_poly)  
  
# Output the predictions  
print(f"Predicted price using Ridge Regression: {ridge_prediction[0]}")  
print(f"Predicted price using Lasso Regression: {lasso_prediction[0]}")
```

Predicted price using Ridge Regression: 14432.752189240991

Predicted price using Lasso Regression: 8237.739556120381