In [1]:
```
#import basic library libraries

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
```

In [3]:
```
cars=pd.read_csv(r"C:\Users\ASUS\Downloads\cars_clus (1).csv")
```

In [4]:
```
cars.head()
```

Out[4]:

	manufact	model	sales	resale	type	price	engine_s	horsepow	wheelbas	width	length	curb_wgt	fuel_cap	mpg	lnsales	partition
0	Acura	Integra	16.919	16.360	0	21.50	1.8	140	101.2	67.3	172.4	2.639	13.2	28.0	2.828	0
1	Acura	TL	39.384	19.875	0	28.40	3.2	225	108.1	70.3	192.9	3.517	17.2	25.0	3.673	0
2	Acura	RL	8.588	29.725	0	42.00	3.5	210	114.6	71.4	196.6	3.850	18.0	22.0	2.150	0
3	Audi	A4	20.397	22.255	0	23.99	1.8	150	102.6	68.2	178.0	2.998	16.4	27.0	3.015	0
4	Audi	A6	18.780	23.555	0	33.95	2.8	200	108.7	76.1	192.0	3.561	18.5	22.0	2.933	0

In [5]:
```
cars_cleaned=cars.drop(columns=['manufact','model'])
```

In [6]:
```
cars_cleaned.head()
```

Out[6]:

	sales	resale	type	price	engine_s	horsepow	wheelbas	width	length	curb_wgt	fuel_cap	mpg	lnsales	partition
0	16.919	16.360	0	21.50	1.8	140	101.2	67.3	172.4	2.639	13.2	28.0	2.828	0
1	39.384	19.875	0	28.40	3.2	225	108.1	70.3	192.9	3.517	17.2	25.0	3.673	0
2	8.588	29.725	0	42.00	3.5	210	114.6	71.4	196.6	3.850	18.0	22.0	2.150	0
3	20.397	22.255	0	23.99	1.8	150	102.6	68.2	178.0	2.998	16.4	27.0	3.015	0
4	18.780	23.555	0	33.95	2.8	200	108.7	76.1	192.0	3.561	18.5	22.0	2.933	0

In [7]:
```
cars_cleaned.info()
```

Out[7]:
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 117 entries, 0 to 116
Data columns (total 14 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   sales      117 non-null    float64
 1   resale     117 non-null    float64
 2   type       117 non-null    int64
 3   price      117 non-null    float64
 4   engine_s   117 non-null    float64
 5   horsepow   117 non-null    int64
 6   wheelbas   117 non-null    float64
 7   width      117 non-null    float64
 8   length     117 non-null    float64
 9   curb_wgt   117 non-null    float64
 10  fuel_cap   117 non-null    float64
 11  mpg        117 non-null    float64
 12  lnsales    117 non-null    float64
 13  partition  117 non-null    int64
dtypes: float64(11), int64(3)
memory usage: 12.9 KB
```

In [8]:
```
cars_cleaned.isnull().sum()
```

Out[8]:
```
sales        0
resale       0
type         0
price        0
engine_s     0
horsepow     0
wheelbas     0
width        0
length       0
curb_wgt     0
fuel_cap     0
mpg          0
lnsales      0
partition    0
dtype: int64
```

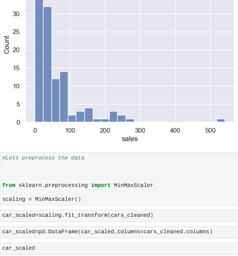
In [9]:
```
cars_cleaned.describe()
```

Out[9]:

	sales	resale	type	price	engine_s	horsepow	wheelbas	width	length	curb_wgt	fuel_cap	mpg	lnsales	partition
count	117.000000	117.000000	117.000000	117.000000	117.000000	117.000000	117.000000	117.000000	117.000000	117.000000	117.000000	117.000000	117.000000	117.0
mean	58.112316	18.053158	0.247863	25.969487	3.040718	181.282051	107.325496	71.189744	187.717949	3.324051	17.812821	24.117949	3.401179	0.0
std	75.058933	11.605632	0.433629	14.149699	1.055169	58.591786	8.050588	3.520151	13.849926	0.597177	3.794609	4.404164	1.337862	0.0
min	0.110000	5.160000	0.000000	9.235000	1.000000	55.000000	92.600000	62.600000	149.400000	1.895000	10.300000	15.000000	-2.207000	0.0
25%	16.767000	11.240000	0.000000	16.980000	2.200000	140.000000	102.400000	68.500000	177.500000	2.911000	15.300000	22.000000	2.819000	0.0
50%	32.299000	14.010000	0.000000	21.660000	3.000000	175.000000	107.000000	70.400000	187.800000	3.340000	17.200000	24.000000	3.475000	0.0
75%	76.029000	19.875000	0.000000	29.465000	3.800000	210.000000	111.600000	73.600000	196.500000	3.823000	19.800000	26.000000	4.331000	0.0
max	540.561000	67.550000	1.000000	82.600000	8.000000	450.000000	138.700000	79.300000	224.500000	5.115000	32.000000	45.000000	6.293000	0.0

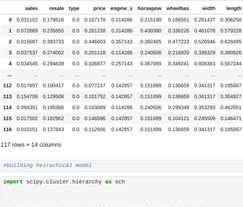
In [12]:
```
#Plotting the data


sns.scatterplot(data=cars_cleaned,x="price",y="mpg")
```

Out[12]:
```
<AxesSubplot:xlabel='price', ylabel='mpg'>
```



In [13]:
```
#visualising the distribution of sales


sns.histplot(cars_cleaned["sales"])
```

Out[13]:
```
<AxesSubplot:xlabel='sales', ylabel='Count'>
```



In [16]:
```
#Lets preprocess the data


from sklearn.preprocessing import MinMaxScaler

scaling = MinMaxScaler()
```

In [17]:
```
car_scaled=scaling.fit_transform(cars_cleaned)
```

In [18]:
```
car_scaled=pd.DataFrame(car_scaled,columns=cars_cleaned.columns)
```

In [19]:
```
car_scaled
```

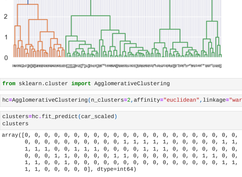
Out[19]:

	sales	resale	type	price	engine_s	horsepow	wheelbas	width	length	curb_wgt	fuel_cap	mpg	lnsales	partition
0	0.031102	0.179516	0.0	0.167178	0.114286	0.215190	0.186551	0.281437	0.306258	0.231056	0.113641	0.433333	0.592353	0.0
1	0.072669	0.230645	0.0	0.261228	0.314286	0.430380	0.336226	0.461078	0.579228	0.502727	0.317972	0.333333	0.691765	0.0
2	0.015687	0.393733	0.0	0.446603	0.357143	0.392405	0.477223	0.526946	0.628495	0.607143	0.354839	0.233333	0.512588	0.0
3	0.037537	0.274002	0.0	0.201118	0.114286	0.240506	0.216920	0.335329	0.380826	0.342547	0.281106	0.400000	0.614353	0.0
4	0.034545	0.294839	0.0	0.336877	0.257143	0.367089	0.349241	0.808383	0.567244	0.517391	0.377880	0.233333	0.604706	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
112	0.017857	0.100417	0.0	0.077217	0.142857	0.151899	0.136659	0.341317	0.185087	0.270807	0.193548	0.366667	0.527647	0.0
113	0.154706	0.129508	0.0	0.101752	0.142857	0.151899	0.136659	0.341317	0.304927	0.297516	0.193548	0.366667	0.780471	0.0
114	0.094351	0.185366	0.0	0.163089	0.114286	0.240506	0.299349	0.353293	0.462051	0.356522	0.281106	0.400000	0.722471	0.0
115	0.017502	0.182962	0.0	0.146596	0.142857	0.151899	0.104121	0.245509	0.146471	0.367702	0.156682	0.366667	0.525412	0.0
116	0.010151	0.137043	0.0	0.112656	0.142857	0.151899	0.136659	0.341317	0.185087	0.269255	0.190157	0.366667	0.462235	0.0

117 rows × 14 columns

In [20]:
```
#building heirachical model
```

In [21]:
```
import scipy.cluster.hierarchy as sch


dendogram=sch.dendrogram(sch.linkage(car_scaled,method="ward"))
```



In [22]:
```
from sklearn.cluster import AgglomerativeClustering
```

In [23]:
```
hc=AgglomerativeClustering(n_clusters=2,affinity="euclidean",linkage="ward")
```

In [24]:
```
clusters=hc.fit_predict(car_scaled)
clusters
```

Out[24]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1,
       1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
       1, 1, 0, 0, 0, 0], dtype=int64)
```

In [25]:
```
cars['cluster']=clusters
```

In [26]:
```
cars
```

Out[26]:

	manufact	model	sales	resale	type	price	engine_s	horsepow	wheelbas	width	length	curb_wgt	fuel_cap	mpg	lnsales	partition	cluster
0	Acura	Integra	16.919	16.360	0	21.50	1.8	140	101.2	67.3	172.4	2.639	13.2	28.0	2.828	0	0
1	Acura	TL	39.384	19.875	0	28.40	3.2	225	108.1	70.3	192.9	3.517	17.2	25.0	3.673	0	0
2	Acura	RL	8.588	29.725	0	42.00	3.5	210	114.6	71.4	196.6	3.850	18.0	22.0	2.150	0	0
3	Audi	A4	20.397	22.255	0	23.99	1.8	150	102.6	68.2	178.0	2.998	16.4	27.0	3.015	0	0
4	Audi	A6	18.780	23.555	0	33.95	2.8	200	108.7	76.1	192.0	3.561	18.5	22.0	2.933	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
112	Volkswagen	Golf	9.761	11.425	0	14.90	2.0	115	98.9	68.3	163.3	2.767	14.5	26.0	2.278	0	0
113	Volkswagen	Jetta	83.721	13.240	0	16.70	2.0	115	98.9	68.3	172.3	2.853	14.5	26.0	4.427	0	0
114	Volkswagen	Passat	51.022	16.725	0	21.20	1.8	150	106.4	68.5	184.1	3.043	16.4	27.0	3.934	0	0
115	Volkswagen	Cabrio	9.569	16.575	0	19.99	2.0	115	97.4	66.7	160.4	3.079	13.7	26.0	2.259	0	0
116	Volkswagen	GTI	5.596	13.760	0	17.50	2.0	115	98.9	68.3	163.3	2.762	14.6	26.0	1.722	0	0

117 rows × 17 columns

In [27]:
```
#cluster analysis
cars.groupby('cluster').mean()
```

Out[27]:

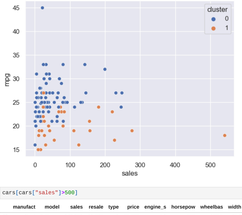
	sales	resale	type	price	engine_s	horsepow	wheelbas	width	length	curb_wgt	fuel_cap	mpg	lnsales	partition
cluster														
0	47.151841	18.639716	0.0	26.586025	2.939636	182.818182	105.554545	70.744318	186.848864	3.163875	16.539773	25.475	3.196625	0.0
1	95.406172	16.186034	1.0	24.096793	3.382759	176.625690	112.703448	72.541379	190.355172	3.810103	21.675862	20.000	4.015828	0.0

In [29]:
```
#visualize cluster on base of type & mileage

sns.scatterplot(x=cars['type'],y=cars['mpg'],hue=cars['cluster'])
```

Out[29]:
```
<AxesSubplot:xlabel='type', ylabel='mpg'>
```

In [31]:
```
#lets visualise the cluster based on sales and mileage


sns.scatterplot(x=cars["sales"],y=cars["mpg"],hue=cars["cluster"])
```

Out[31]:
```
<AxesSubplot:xlabel='sales', ylabel='mpg'>
```



In [32]:
```
cars[cars["sales"]>500]
```

Out[32]:

	manufact	model	sales	resale	type	price	engine_s	horsepow	wheelbas	width	length	curb_wgt	fuel_cap	mpg	lnsales	partition	cluster
46	Ford	F-Series	540.561	15.075	1	26.935	4.6	220	138.5	79.1	224.5	4.241	25.1	18.0	6.293	0	1

In [ ]: