In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_excel("used_cars_set1.xlsx")
df
```

Out[2]:

	brand	model	model_year	milage	fuel_type	engine	transmission	ext_col	int_col	accident	clean_title	price
0	Ford	Utility Police Interceptor Base	2013.0	51,000 mi.	E85 Flex Fuel	300.0HP 3.7L V6 Cylinder Engine Flex Fuel Capa...	6-Speed A/T	Black	Black	At least 1 accident or damage reported	Yes	$10,300
1	Hyundai	Palisade SEL	2021.0	34,742 mi.	Gasoline	3.8L V6 24V GDI DOHC	8-Speed Automatic	Moonlight Cloud	Gray	At least 1 accident or damage reported	Yes	$38,005
2	Lexus	RX 350 RX 350	2022.0	22,372 mi.	Gasoline	3.5 Liter DOHC	Automatic	Blue	Black	None reported	NaN	$54,598
3	INFINITI	Q50 Hybrid Sport	2015.0	88,900 mi.	Hybrid	354.0HP 3.5L V6 Cylinder Engine Gas/Electric H...	7-Speed A/T	Black	Black	None reported	Yes	$15,500
4	Audi	Q3 45 S line Premium Plus	2021.0	9,835 mi.	Gasoline	2.0L I4 16V GDI DOHC Turbo	8-Speed Automatic	Glacier White Metallic	Black	None reported	NaN	$34,999
...	...	...	...	...	...	...	...	...	...	...	...	...
4004	Bentley	Continental GT Speed	2023.0	714 mi.	Gasoline	6.0L W12 48V PDI DOHC Twin Turbo	8-Speed Automatic with Auto-Shift	C / C	Hotspur	None reported	Yes	$349,950
4005	Audi	S4 3.0T Premium Plus	2022.0	10,900 mi.	Gasoline	349.0HP 3.0L V6 Cylinder Engine Gasoline Fuel	Transmission w/Dual Shift Mode	Black	Black	None reported	Yes	$53,900
4006	Porsche	Taycan	2022.0	2,116 mi.	NaN	Electric	Automatic	Black	Black	None reported	NaN	$90,998
4007	Ford	F-150 Raptor	2020.0	33,000 mi.	Gasoline	450.0HP 3.5L V6 Cylinder Engine Gasoline Fuel	A/T	Blue	Black	None reported	Yes	$62,999
4008	BMW	X3 xDrive30i	2020.0	43,000 mi.	Gasoline	248.0HP 2.0L 4 Cylinder Engine Gasoline Fuel	A/T	Gray	Brown	At least 1 accident or damage reported	Yes	$40,000

4009 rows × 12 columns

In [3]:
```
df.shape
```

Out[3]:
```
(4009, 12)
```

In [4]:
```
df.describe()
```

Out[4]:

	model_year
count	4003.000000
mean	2015.519360
std	6.105954
min	1974.000000
25%	2012.000000
50%	2017.000000
75%	2020.000000
max	2024.000000

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
brand             7
model            14
model_year        6
milage            6
fuel_type       181
engine            6
transmission      6
ext_col           6
int_col           6
accident        119
clean_title     602
price             6
dtype: int64
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4009 entries, 0 to 4008
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   brand         4002 non-null   object 
 1   model         3995 non-null   object 
 2   model_year    4003 non-null   float64
 3   milage        4003 non-null   object 
 4   fuel_type     3828 non-null   object 
 5   engine        4003 non-null   object 
 6   transmission  4003 non-null   object 
 7   ext_col       4003 non-null   object 
 8   int_col       4003 non-null   object 
 9   accident      3890 non-null   object 
 10  clean_title   3407 non-null   object 
 11  price         4003 non-null   object 
dtypes: float64(1), object(11)
memory usage: 376.0+ KB
```

In [7]:
```
df.dropna(thresh=11,axis=0,inplace=True)
```

In [8]:
```
df['price'] = df['price'].replace('[\$,]', '', regex=True).astype(float)
```

In [9]:
```
df['milage'] = df['milage'].replace('[\s,mi.]', '', regex=True).astype(float)
```

In [10]:
```
# df.columns = df.columns.str.strip().str.lower().str.replace(" ", "_")
# df['model'] = df.groupby('brand')['model'].transform(lambda x: x.fillna(x.mode()[0]))
# print(df['model'].isnull().sum())
```

In [11]:
```
categorical_cols = ['brand', 'model', 'fuel_type', 'transmission','engine','int_col','ext_col','accident','clean_title']
for col in categorical_cols:
    df[col].fillna(df[col].mode()[0], inplace=True)

numerical_cols = ['model_year', 'milage', 'price']
for col in numerical_cols:
    df[col].fillna(df[col].mean(), inplace=True)
```

In [12]:
```
for col in ['fuel_type', 'engine', 'transmission', 'ext_col', 'int_col']:
    df[col].replace('–', df[col].mode()[0], inplace=True)

df['transmission'].replace('2', df['transmission'].mode()[0], inplace=True)
```

In [13]:
```
df.dtypes
```

Out[13]:
```
brand            object
model            object
model_year      float64
milage          float64
fuel_type        object
engine           object
transmission     object
ext_col          object
int_col          object
accident         object
clean_title      object
price           float64
dtype: object
```

In [14]:
```
df.shape
```

Out[14]:
```
(3867, 12)
```

In [15]:
```
df.isnull().sum()
```

Out[15]:
```
brand           0
model           0
model_year      0
milage          0
fuel_type       0
engine          0
transmission    0
ext_col         0
int_col         0
accident        0
clean_title     0
price           0
dtype: int64
```

In [16]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3867 entries, 0 to 4008
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   brand         3867 non-null   object 
 1   model         3867 non-null   object 
 2   model_year    3867 non-null   float64
 3   milage        3867 non-null   float64
 4   fuel_type     3867 non-null   object 
 5   engine        3867 non-null   object 
 6   transmission  3867 non-null   object 
 7   ext_col       3867 non-null   object 
 8   int_col       3867 non-null   object 
 9   accident      3867 non-null   object 
 10  clean_title   3867 non-null   object 
 11  price         3867 non-null   float64
dtypes: float64(3), object(9)
memory usage: 392.7+ KB
```

In [17]:
```
df.head()
```

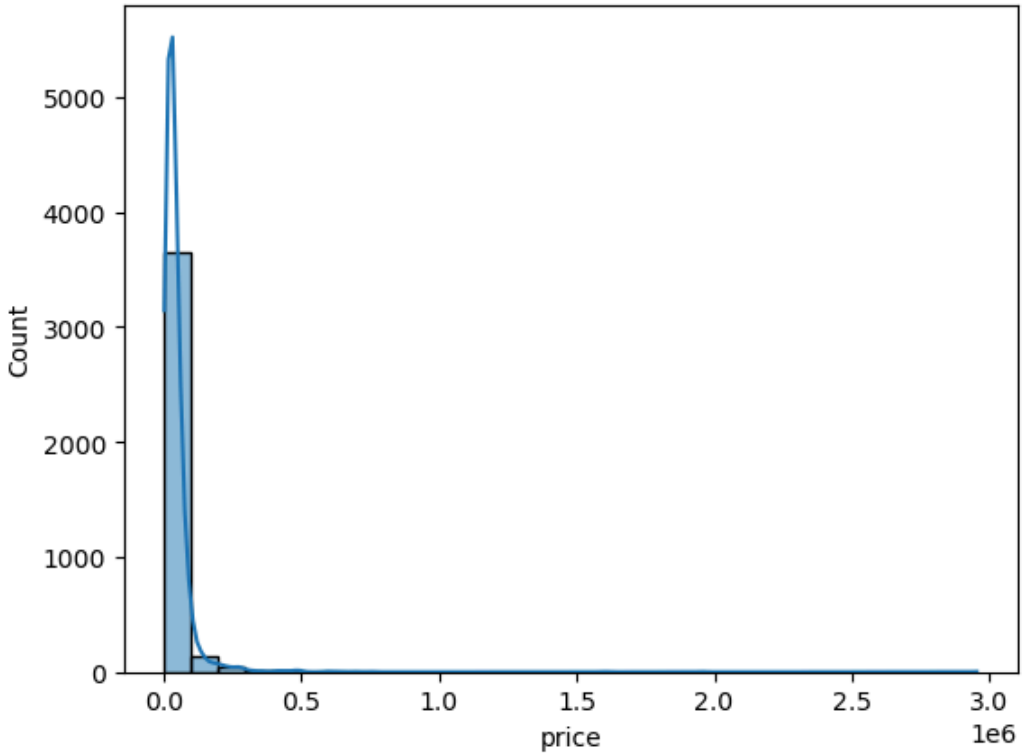
Out[17]:

	brand	model	model_year	milage	fuel_type	engine	transmission	ext_col	int_col	accident	clean_title	price
0	Ford	Utility Police Interceptor Base	2013.0	51000.0	E85 Flex Fuel	300.0HP 3.7L V6 Cylinder Engine Flex Fuel Capa...	6-Speed A/T	Black	Black	At least 1 accident or damage reported	Yes	10300.0
1	Hyundai	Palisade SEL	2021.0	34742.0	Gasoline	3.8L V6 24V GDI DOHC	8-Speed Automatic	Moonlight Cloud	Gray	At least 1 accident or damage reported	Yes	38005.0
2	Lexus	RX 350 RX 350	2022.0	22372.0	Gasoline	3.5 Liter DOHC	Automatic	Blue	Black	None reported	Yes	54598.0
3	INFINITI	Q50 Hybrid Sport	2015.0	88900.0	Hybrid	354.0HP 3.5L V6 Cylinder Engine Gas/Electric H...	7-Speed A/T	Black	Black	None reported	Yes	15500.0
4	Audi	Q3 45 S line Premium Plus	2021.0	9835.0	Gasoline	2.0L I4 16V GDI DOHC Turbo	8-Speed Automatic	Glacier White Metallic	Black	None reported	Yes	34999.0

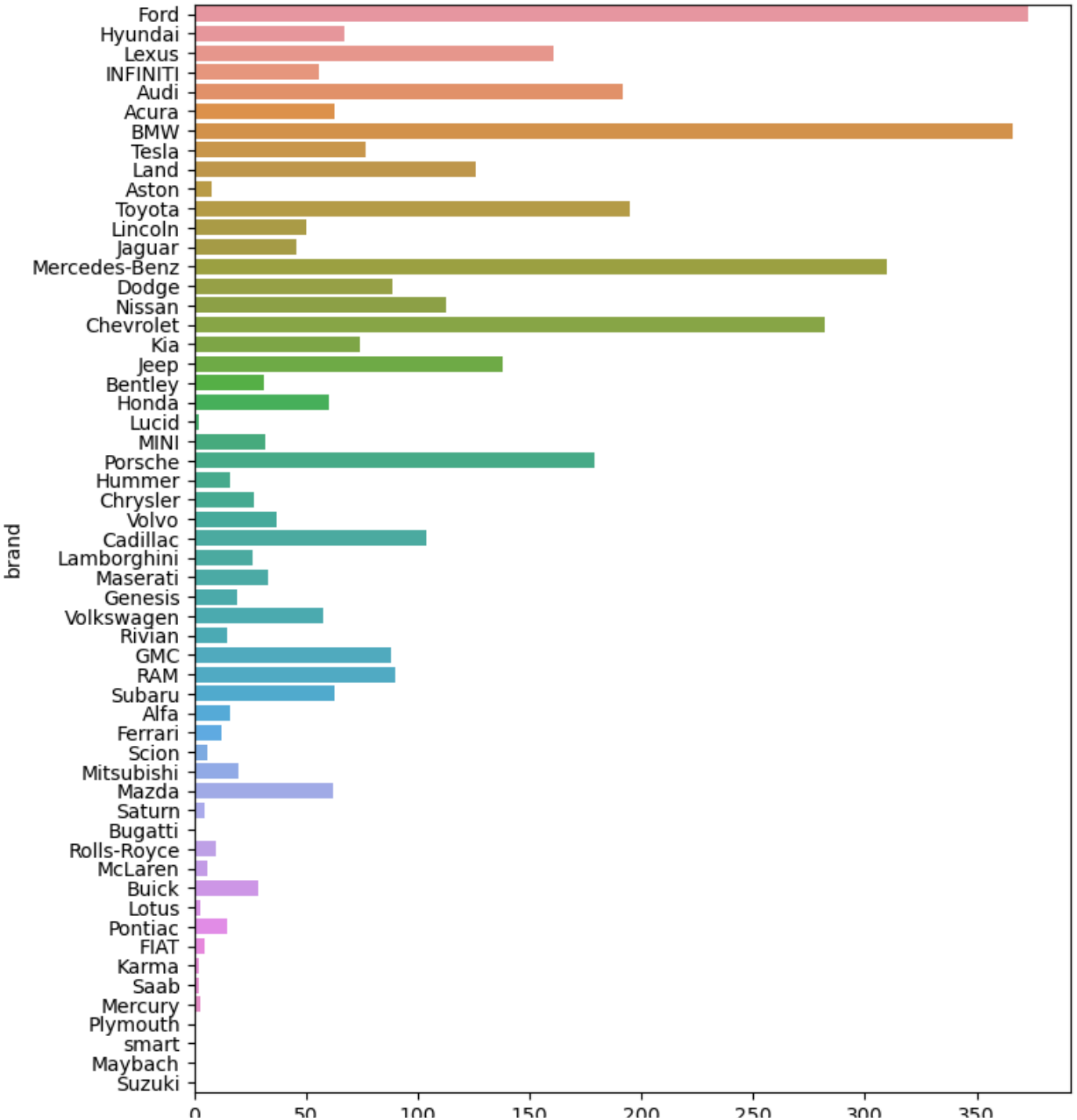
In [18]:
```
print(df.loc[130])
```

```
brand                            Chrysler
model                     Pacifica Touring
model_year                          2017.0
milage                             87305.0
fuel_type                          Gasoline
engine          2.0L I4 16V GDI DOHC Turbo
transmission                    9-Speed A/T
ext_col                              Silver
int_col                               Black
accident                   None reported
clean_title                            Yes
price                               9000.0
Name: 130, dtype: object
```

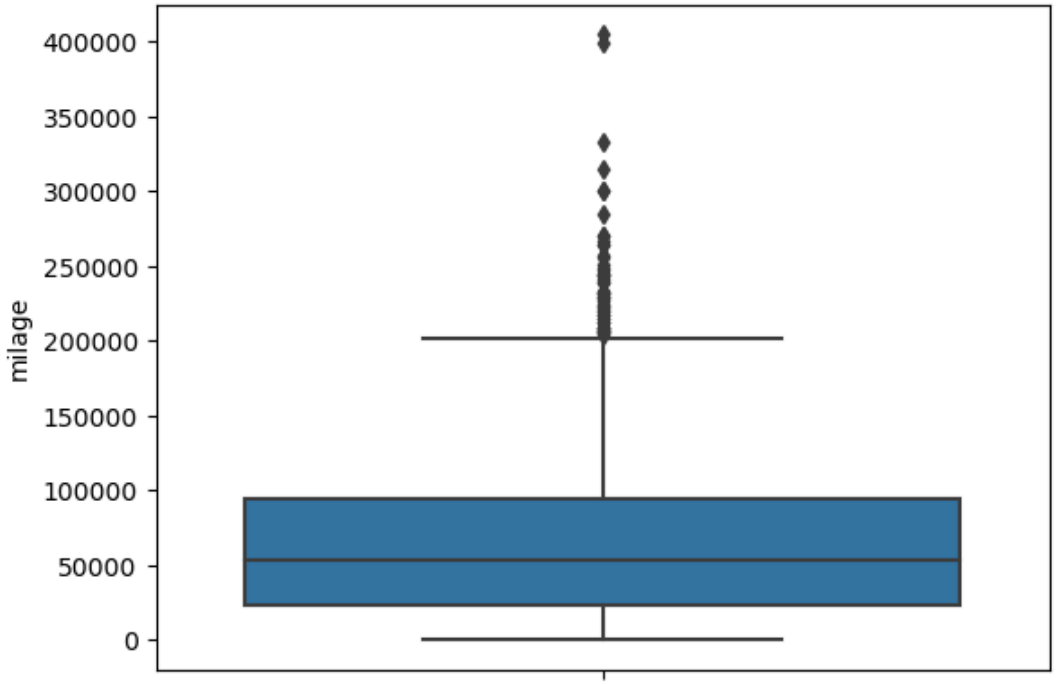
In [19]:
```
sns.histplot(df.price, bins=30, kde=True)
plt.show()
```



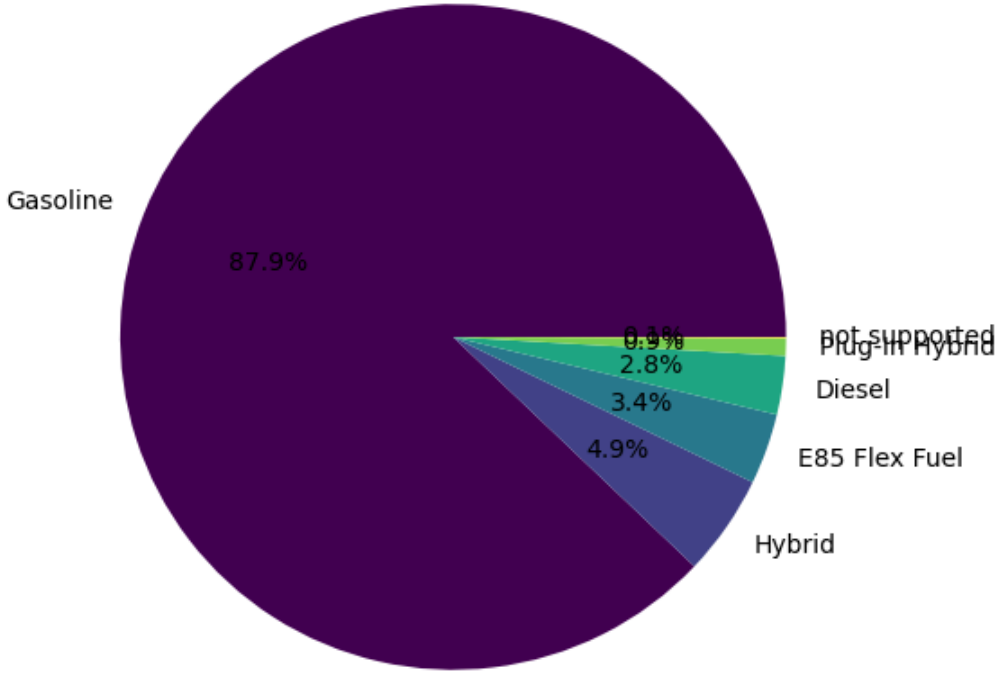
In [20]:
```
plt.figure(figsize=(8,10))
sns.countplot(y=df['brand'])
plt.show()
```


brand
Ford
Hyundai
Lexus
INFINITI
Audi
Acura
BMW
Tesla
Land
Aston
Toyota
Lincoln
Jaguar
Mercedes-Benz
Dodge
Nissan
Chevrolet
Kia
Jeep
Bentley
Honda
Lucid
MINI
Porsche
Hummer
Chrysler
Volvo
Cadillac
Lamborghini
Maserati
Genesis
Volkswagen
Rivian
GMC
RAM
Subaru
Alfa
Ferrari
Scion
Mitsubishi
Mazda
Saturn
Bugatti
Rolls-Royce
McLaren
Buick
Lotus
Pontiac
FIAT
Karma
Saab
Mercury
Plymouth
smart
Maybach
Suzuki
0
50
100
150
200
250
300
350

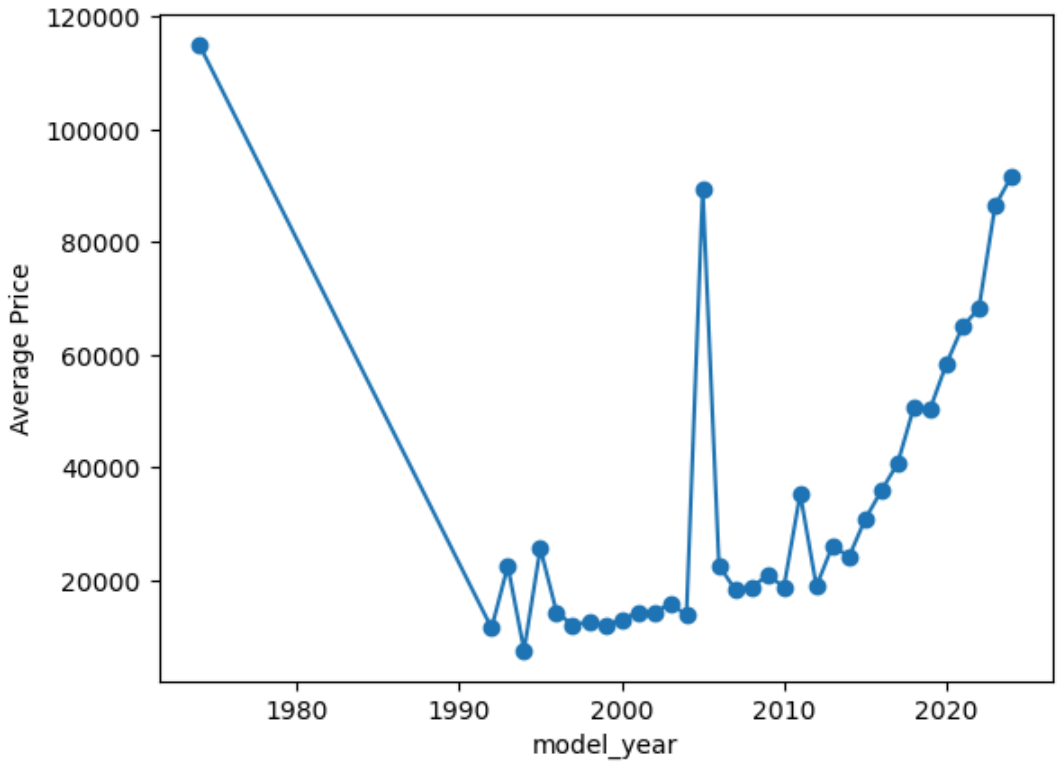
In [21]:
```
sns.boxplot(y=df['milage'])
plt.show()
```



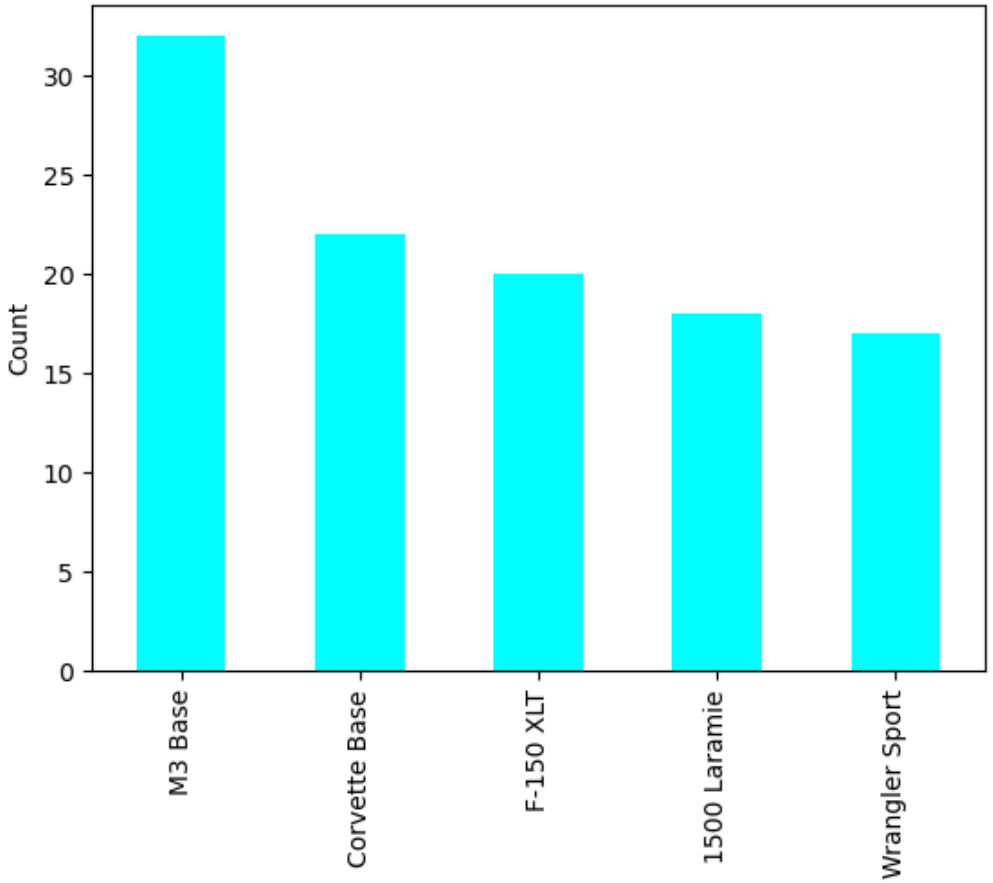
In [22]:
```
df['fuel_type'].value_counts().plot.pie(autopct='%1.1f%%', cmap='viridis', figsize=(6, 6))
plt.ylabel('')
plt.show()
```



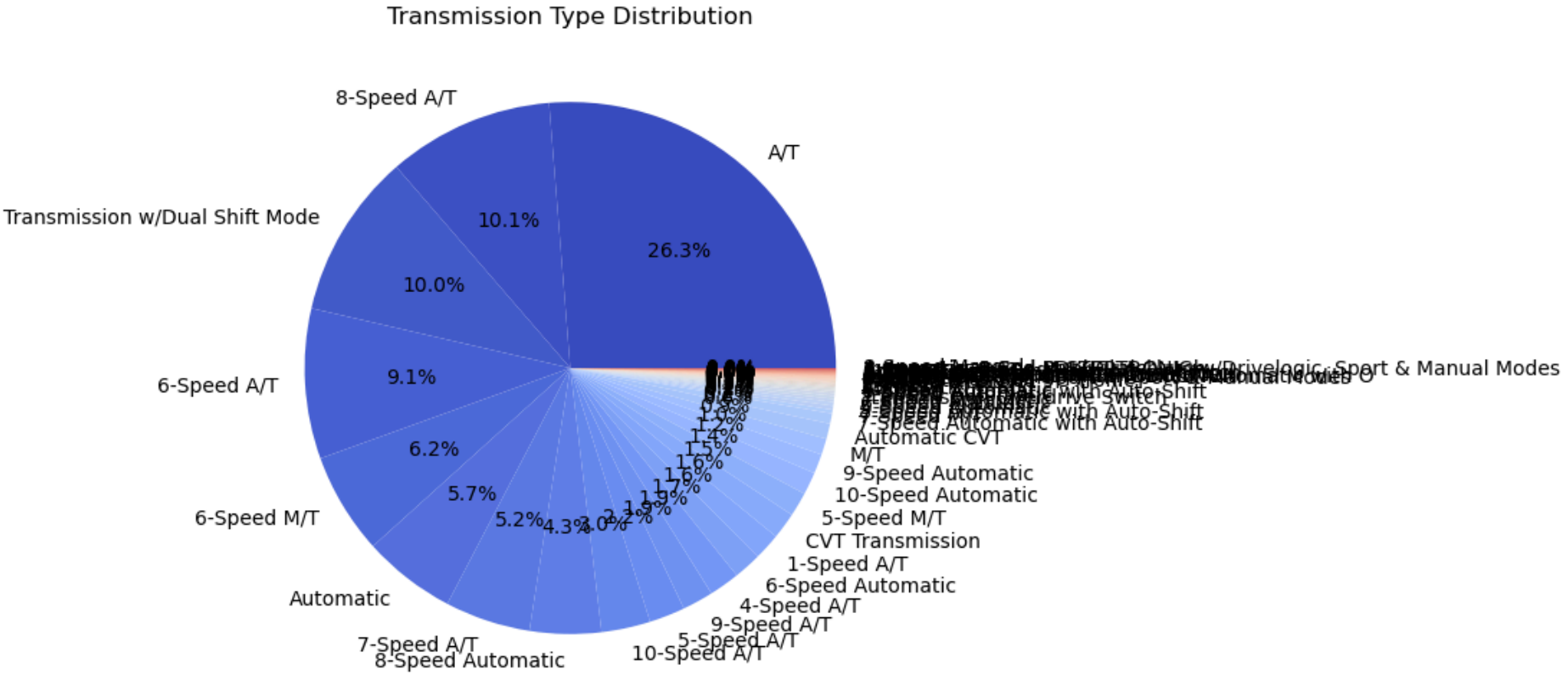
In [23]:
```
df.groupby('model_year')['price'].mean().plot(kind='line', marker='o')
plt.ylabel('Average Price')
plt.show()
```



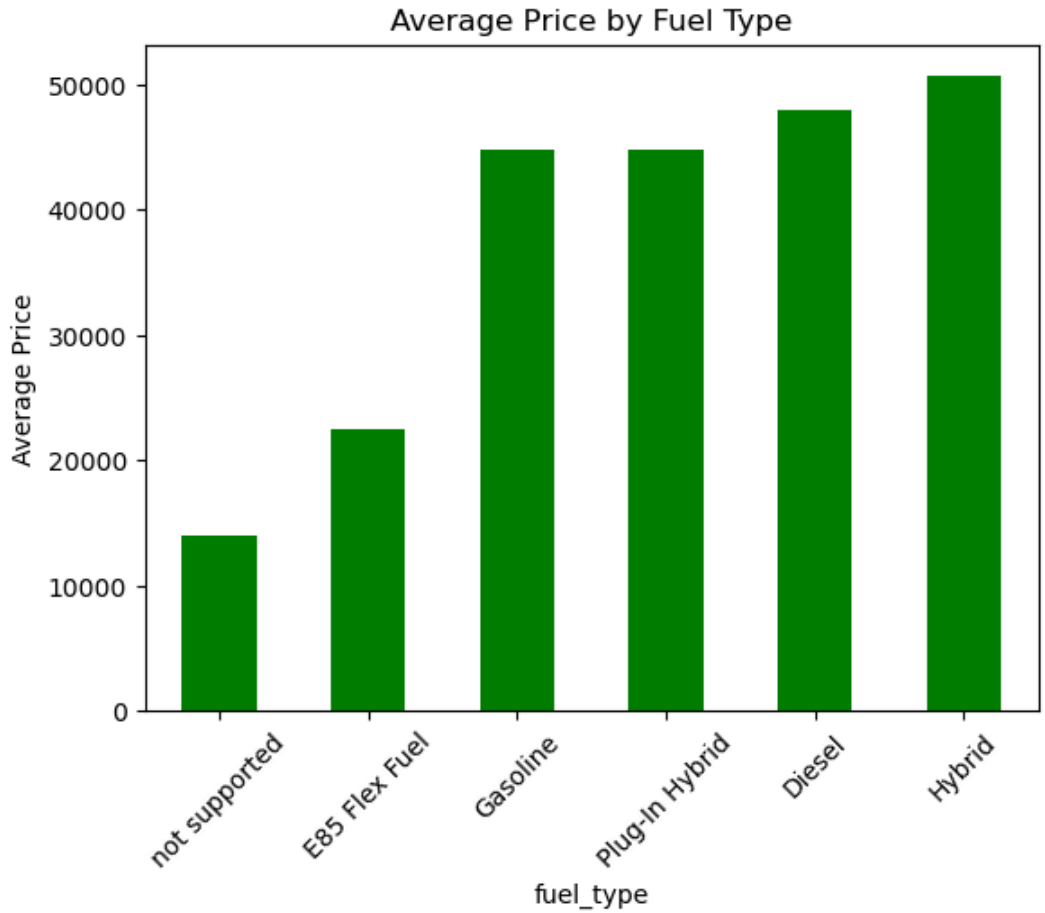
In [24]:
```
df['model'].value_counts()[:5].plot(kind='bar', color='cyan')
plt.ylabel('Count')
plt.show()
```



In [25]:
```
plt.figure(figsize=(15,15))
df['transmission'].value_counts().plot.pie(autopct='%1.1f%%', cmap='coolwarm', figsize=(6, 6))
plt.ylabel('')
plt.title("Transmission Type Distribution")
plt.show()
```

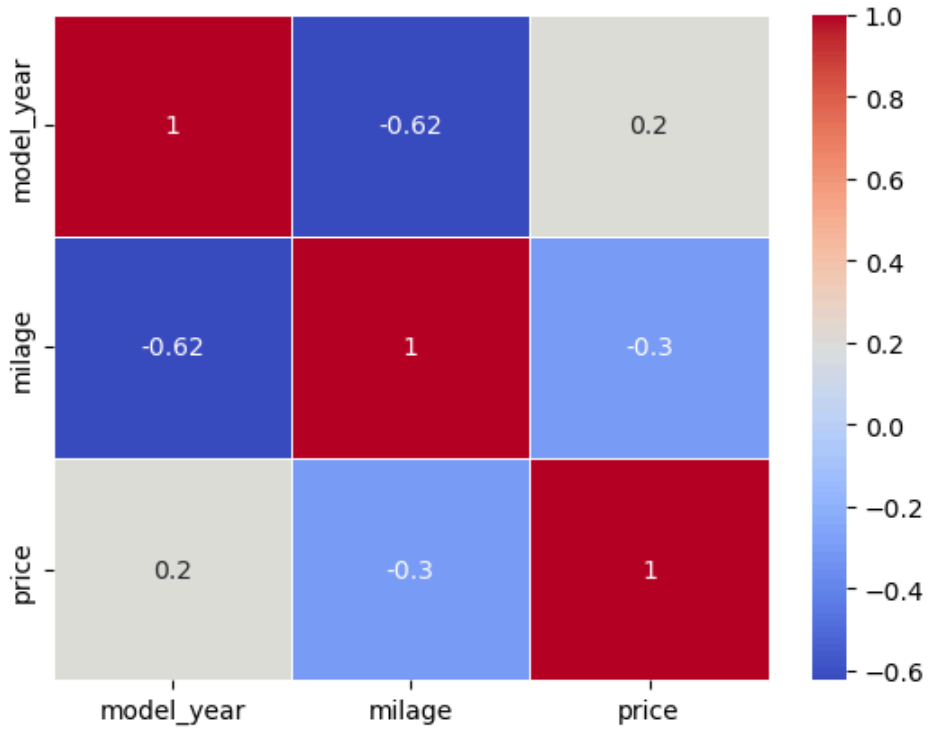


In [26]:
```
df.groupby('fuel_type')['price'].mean().sort_values().plot(kind='bar', color='green')
plt.ylabel("Average Price")
plt.title("Average Price by Fuel Type")
plt.xticks(rotation=45)
plt.show()
```



In [27]:
```
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
plt.show()
```

```
C:\Users\subha\AppData\Local\Temp\ipykernel_22636\1760633978.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr
is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silenc
e this warning.
  sns.heatmap(df.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
```



In [56]:
```
brand = pd.get_dummies(df['brand'], prefix="Brand")
model = pd.get_dummies(df['model'], prefix="Model")
fuel_type = pd.get_dummies(df['fuel_type'], prefix="Fuel")
transmission = pd.get_dummies(df['transmission'], prefix="Trans")
engine = pd.get_dummies(df['engine'], prefix="Engine")
int_col = pd.get_dummies(df['int_col'], prefix="IntColor")
ext_col = pd.get_dummies(df['ext_col'], prefix="ExtColor")
accident = pd.get_dummies(df['accident'], prefix="Accident")
clean_title = pd.get_dummies(df['clean_title'], prefix="Title")
```

In [58]:
```
print (brand)
```

```
      Brand_Acura  Brand_Alfa  Brand_Aston  Brand_Audi  Brand_BMW  \
0               0           0            0           0          0
1               0           0            0           0          0
2               0           0            0           0          0
3               0           0            0           0          0
4               0           0            0           1          0
...           ...         ...          ...         ...        ...
4003            0           0            0           0          0
4004            0           0            0           0          0
4005            0           0            0           1          0
4007            0           0            0           0          0
4008            0           0            0           0          1

      Brand_Bentley  Brand_Bugatti  Brand_Buick  Brand_Cadillac  \
0                 0              0            0               0
1                 0              0            0               0
2                 0              0            0               0
3                 0              0            0               0
4                 0              0            0               0
...             ...            ...          ...             ...
4003              0              0            0               0
4004              1              0            0               0
4005              0              0            0               0
4007              0              0            0               0
4008              0              0            0               0

      Brand_Chevrolet  ...  Brand_Saab  Brand_Saturn  Brand_Scion  \
0                   0  ...           0             0            0
1                   0  ...           0             0            0
2                   0  ...           0             0            0
3                   0  ...           0             0            0
4                   0  ...           0             0            0
...               ...  ...         ...           ...          ...
4003                0  ...           0             0            0
4004                0  ...           0             0            0
4005                0  ...           0             0            0
4007                0  ...           0             0            0
4008                0  ...           0             0            0

      Brand_Subaru  Brand_Suzuki  Brand_Tesla  Brand_Toyota  Brand_Volkswagen  \
0                0             0            0             0                 0
1                0             0            0             0                 0
2                0             0            0             0                 0
3                0             0            0             0                 0
4                0             0            0             0                 0
...            ...           ...          ...           ...               ...
```

```
4003             0             0             0             0                0
4004             0             0             0             0                0
4005             0             0             0             0                0
4007             0             0             0             0                0
4008             0             0             0             0                0

      Brand_Volvo  Brand_smart
0               0            0
1               0            0
2               0            0
3               0            0
4               0            0
...           ...          ...
4003            0            0
4004            0            0
4005            0            0
4007            0            0
4008            0            0

[3867 rows x 56 columns]
```

In [60]:
```
print(model)
```

```
      Model_124 Spider Abarth  Model_128 i  Model_135 i  Model_135 is  \
0                           0            0            0             0
1                           0            0            0             0
2                           0            0            0             0
3                           0            0            0             0
4                           0            0            0             0
...                       ...          ...          ...           ...
4003                        0            0            0             0
4004                        0            0            0             0
4005                        0            0            0             0
4007                        0            0            0             0
4008                        0            0            0             0

      Model_1500 Big Horn  Model_1500 Cheyenne  \
0                       0                    0
1                       0                    0
2                       0                    0
3                       0                    0
4                       0                    0
...                   ...                  ...
4003                    0                    0
4004                    0                    0
4005                    0                    0
4007                    0                    0
4008                    0                    0

      Model_1500 Cheyenne Extended Cab  Model_1500 Classic SLT  \
0                                    0                       0
1                                    0                       0
2                                    0                       0
3                                    0                       0
4                                    0                       0
...                                ...                     ...
4003                                 0                       0
4004                                 0                       0
4005                                 0                       0
4007                                 0                       0
4008                                 0                       0

      Model_1500 Classic Tradesman  Model_1500 Classic Warlock  ...  \
0                                0                           0  ...
1                                0                           0  ...
2                                0                           0  ...
3                                0                           0  ...
4                                0                           0  ...
...                            ...                         ...  ...
```

```
4003                            0                           0  ...
4004                            0                           0  ...
4005                            0                           0  ...
4007                            0                           0  ...
4008                            0                           0  ...

      Model_e-tron Prestige  Model_i3 120Ah w/Range Extender  Model_i3 94 Ah  \
0                         0                                 0               0
1                         0                                 0               0
2                         0                                 0               0
3                         0                                 0               0
4                         0                                 0               0
...                     ...                               ...             ...
4003                      0                                 0               0
4004                      0                                 0               0
4005                      0                                 0               0
4007                      0                                 0               0
4008                      0                                 0               0

      Model_i3 Base  Model_i3 Base w/Range Extender  Model_i8 Base  \
0                 0                               0              0
1                 0                               0              0
2                 0                               0              0
3                 0                               0              0
4                 0                               0              0
...             ...                             ...            ...
4003              0                               0              0
4004              0                               0              0
4005              0                               0              0
4007              0                               0              0
4008              0                               0              0

      Model_tC Anniversary Edition  Model_tC Base  \
0                                0              0
1                                0              0
2                                0              0
3                                0              0
4                                0              0
...                            ...            ...
4003                             0              0
4004                             0              0
4005                             0              0
4007                             0              0
4008                             0              0

      Model_tC Release Series 6.0  Model_xB Base
```

```
0                              0             0
1                              0             0
2                              0             0
3                              0             0
4                              0             0
...                          ...           ...
4003                           0             0
4004                           0             0
4005                           0             0
4007                           0             0
4008                           0             0

[3867 rows x 1855 columns]
```

In [62]:
```
df.drop(['brand', 'model', 'fuel_type', 'transmission','engine','int_col','ext_col','accident','clean_title'],axis=1,inplace=True)
df = pd.concat([df, brand, model, fuel_type, transmission, engine, int_col, ext_col, accident, clean_title], axis=1)
df
```

Out[62]:

	model_year	milage	price	Brand_Acura	Brand_Alfa	Brand_Aston	Brand_Audi	Brand_BMW	Brand_Bentley	Brand_Bugatti	...	ExtColor_Wolf Gray	ExtColor_Yell
0	2013.0	51000.0	10300.0	0	0	0	0	0	0	0	...	0	
1	2021.0	34742.0	38005.0	0	0	0	0	0	0	0	...	0	
2	2022.0	22372.0	54598.0	0	0	0	0	0	0	0	...	0	
3	2015.0	88900.0	15500.0	0	0	0	0	0	0	0	...	0	
4	2021.0	9835.0	34999.0	0	0	0	1	0	0	0	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4003	2018.0	53705.0	25900.0	0	0	0	0	0	0	0	...	0	
4004	2023.0	714.0	349950.0	0	0	0	0	0	1	0	...	0	
4005	2022.0	10900.0	53900.0	0	0	0	1	0	0	0	...	0	
4007	2020.0	33000.0	62999.0	0	0	0	0	0	0	0	...	0	
4008	2020.0	43000.0	40000.0	0	0	0	0	1	0	0	...	0	

3867 rows × 3559 columns

In [64]:
```
from sklearn.linear_model import LinearRegression
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_squared_error
```

In [66]:
```
X = df.drop("price", axis=1)
y = df["price"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [68]:
```
reg = linear_model.LinearRegression()
reg.fit(X_train, y_train)
```

Out[68]:
```
▾ LinearRegression

LinearRegression()
```

In [70]:
```
X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3093 entries, 2257 to 3285
Columns: 3558 entries, model_year to Title_Yes
dtypes: float64(2), uint8(3556)
memory usage: 10.6 MB
```

In [72]:
```
y_train.info()
```

```
<class 'pandas.core.series.Series'>
Int64Index: 3093 entries, 2257 to 3285
Series name: price
Non-Null Count  Dtype
--------------  -----
3093 non-null   float64
dtypes: float64(1)
memory usage: 48.3 KB
```

In [74]:
```
X_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 774 entries, 2252 to 713
Columns: 3558 entries, model_year to Title_Yes
dtypes: float64(2), uint8(3556)
memory usage: 2.6 MB
```

In [76]:
```
y_test.info()
```

```
<class 'pandas.core.series.Series'>
Int64Index: 774 entries, 2252 to 713
Series name: price
Non-Null Count  Dtype
--------------  -----
774 non-null    float64
dtypes: float64(1)
memory usage: 12.1 KB
```

In [80]:
```
Predictions = reg.predict(X_test)
print(Predictions)
```

```
[ 6.03593546e+11  5.42201597e+04  5.98291162e+03  8.70289209e+03
  1.46668470e+12 -1.57469011e+11  4.13672690e+04  2.25258179e+04
 -4.03027059e+11  5.73335366e+04  5.37340414e+11  3.77245883e+10
 -5.81053646e+11  9.29959705e+10 -4.28975520e+11 -3.02447198e+12
 -3.65349897e+12 -6.48481751e+10 -1.45048897e+12  9.97348584e+03
 -1.37244181e+12 -5.35923678e+11  1.45425581e+04 -2.25767133e+11
  5.01667260e+11 -2.04519716e+12  7.55458127e+11  6.03605640e+04
  3.42566772e+04  5.06733159e+11  1.91267144e+04  1.62472651e+04
 -2.02341260e+03 -9.66995154e+11  4.88899702e+04  2.06631047e+12
  9.93774527e+11  2.17909822e+11  8.70547515e+04  5.13340522e+11
 -7.58801238e+11  2.47035073e+04 -9.89640598e+10  1.97399370e+04
  2.58957828e+12 -1.19874265e+12  1.03359351e+04 -6.59821592e+10
  1.02311401e+04  2.89609155e+04  1.48418294e+13  2.86030815e+04
  2.20305815e+04 -2.31893739e+11  8.63955659e+04  6.40818960e+04
  2.69979429e+04  8.89565327e+04  9.92508350e+03 -6.47732290e+11
  4.93113608e+04  9.46366929e+04 -6.54742682e+11  5.10597564e+11
  7.24428374e+04 -1.14697817e+04 -3.18763150e+11  7.15918012e+11
  2.45285483e+04 -3.94587891e+11 -3.64841926e+10  1.15670699e+12
  2.85462397e+04 -6.54742741e+11  1.88224194e+04  1.78735132e+04
```

In [82]:
```
r2 = r2_score(y_test, Predictions)
mse = mean_squared_error(y_test, Predictions)

print(f"R² Score: {r2}")
print(f"Mean Squared Error: {mse}")
```

```
R² Score: -618224981686816.5
Mean Squared Error: 2.039025580807906e+24
```