In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
```

In [2]:

```
df = pd.read_csv( 'C:\\Users\\radhe\\OneDrive\\Desktop\\all folder\\carprice.c
print(df)
```

```
     car_ID  symboling                   CarName fueltype aspiration  \
0         1          3        alfa-romero giulia      gas        std   
1         2          3       alfa-romero stelvio      gas        std   
2         3          1  alfa-romero Quadrifoglio      gas        std   
3         4          2               audi 100 ls      gas        std   
4         5          2                audi 100ls      gas        std   
..      ...        ...                       ...      ...        ...   
200     201         -1           volvo 145e (sw)      gas        std   
201     202         -1               volvo 144ea      gas      turbo   
202     203         -1               volvo 244dl      gas        std   
203     204         -1                 volvo 246   diesel      turbo   
204     205         -1               volvo 264gl      gas      turbo   

    doornumber      carbody drivewheel enginelocation  wheelbase  ...  \
0          two  convertible        rwd          front       88.6  ...   
1          two  convertible        rwd          front       88.6  ...   
2          two    hatchback        rwd          front       94.5  ...   
3         four        sedan        fwd          front       99.8  ...   
4         four        sedan        4wd          front       99.4  ...   
..         ...          ...        ...            ...        ...  ...   
200       four        sedan        rwd          front      109.1  ...   
201       four        sedan        rwd          front      109.1  ...   
202       four        sedan        rwd          front      109.1  ...   
203       four        sedan        rwd          front      109.1  ...   
204       four        sedan        rwd          front      109.1  ...   

     enginesize  fuelsystem  boreratio  stroke compressionratio horsepower  \
0           130        mpfi       3.47    2.68              9.0        111   
1           130        mpfi       3.47    2.68              9.0        111   
2           152        mpfi       2.68    3.47              9.0        154   
3           109        mpfi       3.19    3.40             10.0        102   
4           136        mpfi       3.19    3.40              8.0        115   
..          ...         ...        ...     ...              ...        ...   
200         141        mpfi       3.78    3.15              9.5        114   
201         141        mpfi       3.78    3.15              8.7        160   
202         173        mpfi       3.58    2.87              8.8        134   
203         145         idi       3.01    3.40             23.0        106   
204         141        mpfi       3.78    3.15              9.5        114   

     peakrpm citympg  highwaympg    price  
0       5000      21          27  13495.0  
1       5000      21          27  16500.0  
2       5000      19          26  16500.0  
3       5500      24          30  13950.0  
4       5500      18          22  17450.0  
..       ...     ...         ...      ...  
200     5400      23          28  16845.0  
201     5300      19          25  19045.0  
202     5500      18          23  21485.0  
203     4800      26          27  22470.0  
204     5400      19          25  22625.0  

[205 rows x 26 columns]
```

In [3]:

```
df.describe()
```

Out[3]:

| | car_ID | symboling | wheelbase | carlength | carwidth | carheight | curbweight | er |
|---|---|---|---|---|---|---|---|---|
| **count** | 205.000000 | 205.000000 | 205.000000 | 205.000000 | 205.000000 | 205.000000 | 205.000000 | 20 |
| **mean** | 103.000000 | 0.834146 | 98.756585 | 174.049268 | 65.907805 | 53.724878 | 2555.565854 | 12 |
| **std** | 59.322565 | 1.245307 | 6.021776 | 12.337289 | 2.145204 | 2.443522 | 520.680204 | 4 |
| **min** | 1.000000 | -2.000000 | 86.600000 | 141.100000 | 60.300000 | 47.800000 | 1488.000000 | 6 |
| **25%** | 52.000000 | 0.000000 | 94.500000 | 166.300000 | 64.100000 | 52.000000 | 2145.000000 | 9 |
| **50%** | 103.000000 | 1.000000 | 97.000000 | 173.200000 | 65.500000 | 54.100000 | 2414.000000 | 12 |
| **75%** | 154.000000 | 2.000000 | 102.400000 | 183.100000 | 66.900000 | 55.500000 | 2935.000000 | 14 |
| **max** | 205.000000 | 3.000000 | 120.900000 | 208.100000 | 72.300000 | 59.800000 | 4066.000000 | 32 |

In [4]:

```
df.isnull().sum()
```

Out[4]:

```
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
price               0
dtype: int64
```

In [5]:
```
df.CarName.unique()
```

Out[5]:
```
array(['alfa-romero giulia', 'alfa-romero stelvio',
       'alfa-romero Quadrifoglio', 'audi 100 ls', 'audi 100ls',
       'audi fox', 'audi 5000', 'audi 4000', 'audi 5000s (diesel)',
       'bmw 320i', 'bmw x1', 'bmw x3', 'bmw z4', 'bmw x4', 'bmw x5',
       'chevrolet impala', 'chevrolet monte carlo', 'chevrolet vega 2300',
       'dodge rampage', 'dodge challenger se', 'dodge d200',
       'dodge monaco (sw)', 'dodge colt hardtop', 'dodge colt (sw)',
       'dodge coronet custom', 'dodge dart custom',
       'dodge coronet custom (sw)', 'honda civic', 'honda civic cvcc',
       'honda accord cvcc', 'honda accord lx', 'honda civic 1500 gl',
       'honda accord', 'honda civic 1300', 'honda prelude',
       'honda civic (auto)', 'isuzu MU-X', 'isuzu D-Max ',
       'isuzu D-Max V-Cross', 'jaguar xj', 'jaguar xf', 'jaguar xk',
       'maxda rx3', 'maxda glc deluxe', 'mazda rx2 coupe', 'mazda rx-4',
       'mazda glc deluxe', 'mazda 626', 'mazda glc', 'mazda rx-7 gs',
       'mazda glc 4', 'mazda glc custom l', 'mazda glc custom',
       'buick electra 225 custom', 'buick century luxus (sw)',
       'buick century', 'buick skyhawk', 'buick opel isuzu deluxe',
       'buick skylark', 'buick century special',
       'buick regal sport coupe (turbo)', 'mercury cougar',
       'mitsubishi mirage', 'mitsubishi lancer', 'mitsubishi outlander',
       'mitsubishi g4', 'mitsubishi mirage g4', 'mitsubishi montero',
       'mitsubishi pajero', 'Nissan versa', 'nissan gt-r', 'nissan rogue',
       'nissan latio', 'nissan titan', 'nissan leaf', 'nissan juke',
       'nissan note', 'nissan clipper', 'nissan nv200', 'nissan dayz',
       'nissan fuga', 'nissan otti', 'nissan teana', 'nissan kicks',
       'peugeot 504', 'peugeot 304', 'peugeot 504 (sw)', 'peugeot 604sl',
       'peugeot 505s turbo diesel', 'plymouth fury iii',
       'plymouth cricket', 'plymouth satellite custom (sw)',
       'plymouth fury gran sedan', 'plymouth valiant', 'plymouth duster',
       'porsche macan', 'porcshce panamera', 'porsche cayenne',
       'porsche boxter', 'renault 12tl', 'renault 5 gtl', 'saab 99e',
       'saab 99le', 'saab 99gle', 'subaru', 'subaru dl', 'subaru brz',
       'subaru baja', 'subaru r1', 'subaru r2', 'subaru trezia',
       'subaru tribeca', 'toyota corona mark ii', 'toyota corona',
       'toyota corolla 1200', 'toyota corona hardtop',
       'toyota corolla 1600 (sw)', 'toyota carina', 'toyota mark ii',
       'toyota corolla', 'toyota corolla liftback',
       'toyota celica gt liftback', 'toyota corolla tercel',
       'toyota corona liftback', 'toyota starlet', 'toyota tercel',
       'toyota cressida', 'toyota celica gt', 'toyouta tercel',
       'vokswagen rabbit', 'volkswagen 1131 deluxe sedan',
       'volkswagen model 111', 'volkswagen type 3', 'volkswagen 411 (sw)',
       'volkswagen super beetle', 'volkswagen dasher', 'vw dasher',
       'vw rabbit', 'volkswagen rabbit', 'volkswagen rabbit custom',
       'volvo 145e (sw)', 'volvo 144ea', 'volvo 244dl', 'volvo 245',
       'volvo 264gl', 'volvo diesel', 'volvo 246'], dtype=object)
```

In [6]:

```
sns.set_style("whitegrid")
plt.figure(figsize=(15, 10))
sns.distplot(df.price)
plt.show()
```

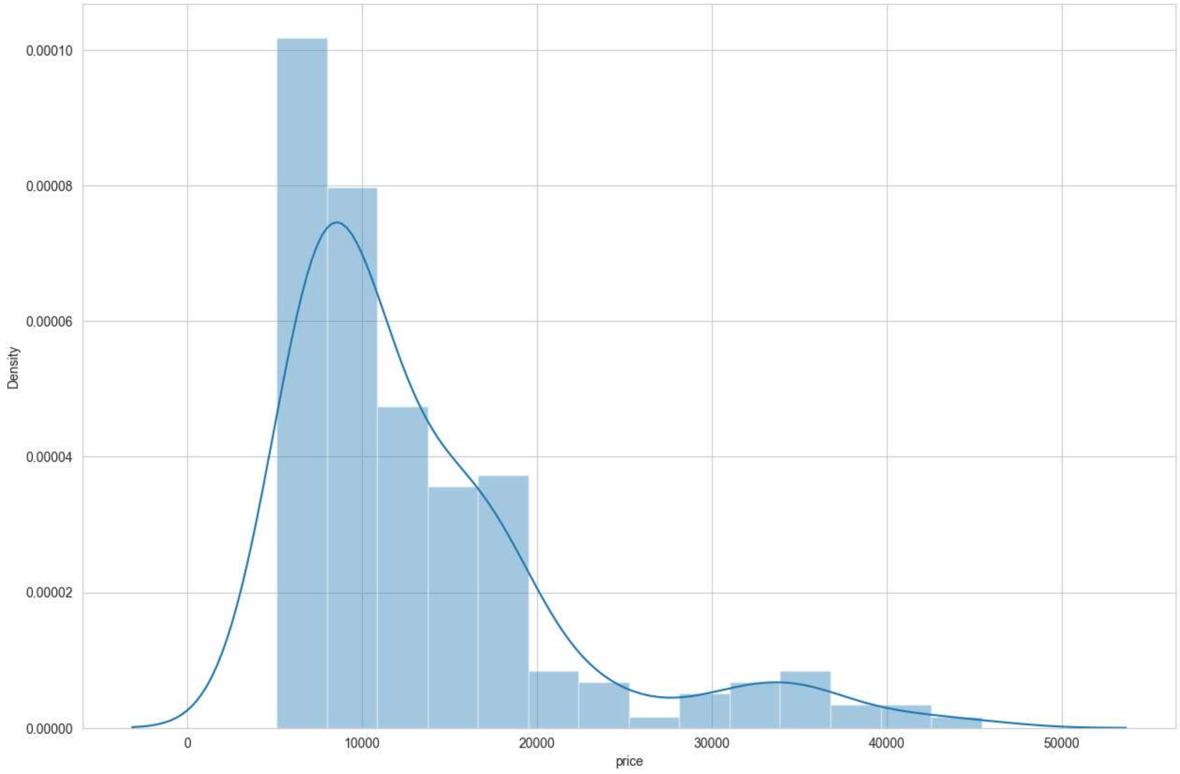
```
C:\Users\radhe\AppData\Local\Temp\ipykernel_20496\3769725905.py:3: UserWarnin
g:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df.price)
```



In [7]:

```
df.corr()
```

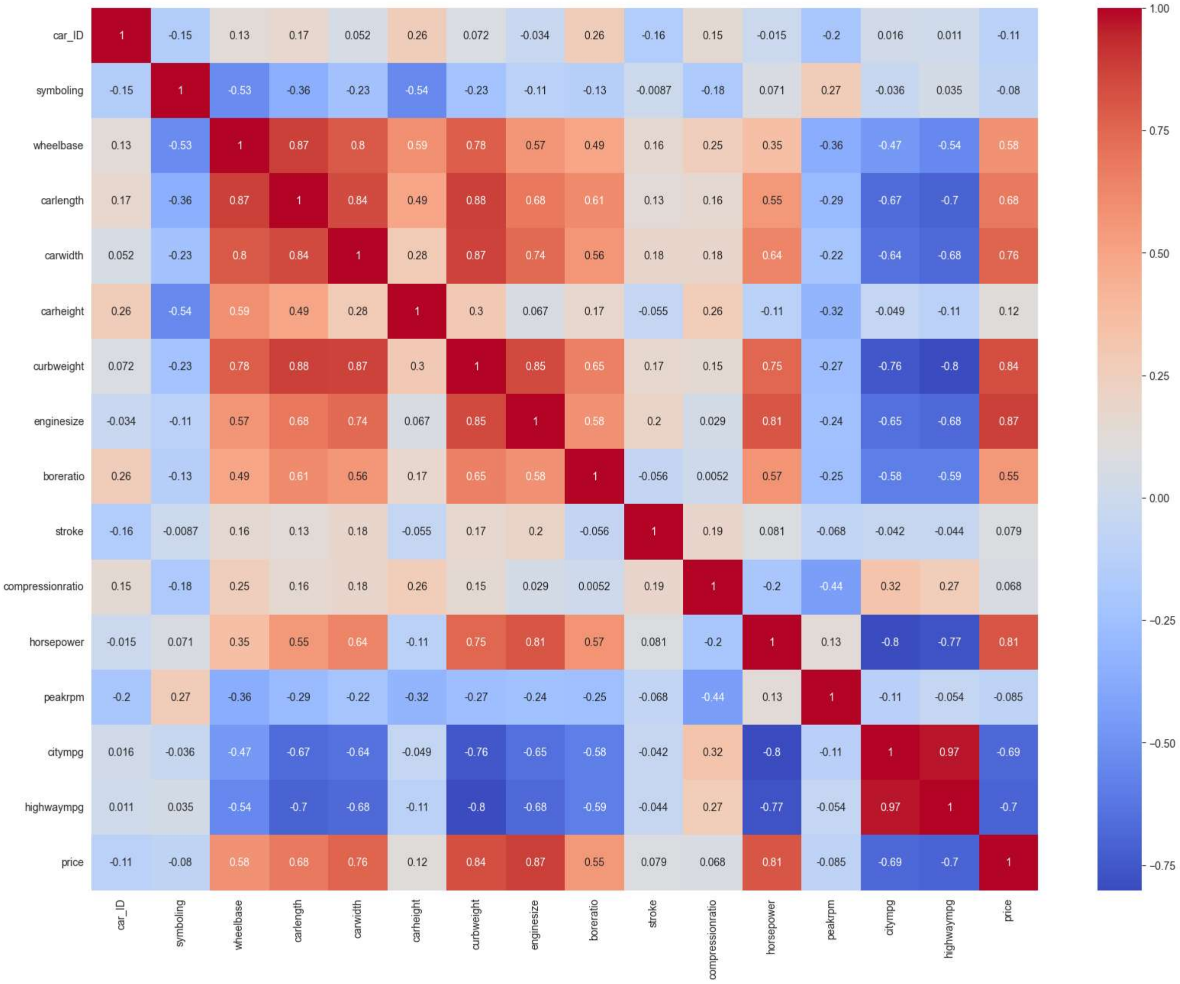
```
C:\Users\radhe\AppData\Local\Temp\ipykernel_20496\1134722465.py:1: FutureWarn
ing: The default value of numeric_only in DataFrame.corr is deprecated. In a
future version, it will default to False. Select only valid columns or specif
y the value of numeric_only to silence this warning.
  df.corr()
```

Out[7]:

| | car_ID | symboling | wheelbase | carlength | carwidth | carheight | curbweight |
|---|---|---|---|---|---|---|---|
| **car_ID** | 1.000000 | -0.151621 | 0.129729 | 0.170636 | 0.052387 | 0.255960 | 0.071962 |
| **symboling** | -0.151621 | 1.000000 | -0.531954 | -0.357612 | -0.232919 | -0.541038 | -0.227691 |
| **wheelbase** | 0.129729 | -0.531954 | 1.000000 | 0.874587 | 0.795144 | 0.589435 | 0.776386 |
| **carlength** | 0.170636 | -0.357612 | 0.874587 | 1.000000 | 0.841118 | 0.491029 | 0.877728 |
| **carwidth** | 0.052387 | -0.232919 | 0.795144 | 0.841118 | 1.000000 | 0.279210 | 0.867032 |
| **carheight** | 0.255960 | -0.541038 | 0.589435 | 0.491029 | 0.279210 | 1.000000 | 0.295572 |
| **curbweight** | 0.071962 | -0.227691 | 0.776386 | 0.877728 | 0.867032 | 0.295572 | 1.000000 |
| **enginesize** | -0.033930 | -0.105790 | 0.569329 | 0.683360 | 0.735433 | 0.067149 | 0.850594 |
| **boreratio** | 0.260064 | -0.130051 | 0.488750 | 0.606454 | 0.559150 | 0.171071 | 0.648480 |
| **stroke** | -0.160824 | -0.008735 | 0.160959 | 0.129533 | 0.182942 | -0.055307 | 0.168790 |
| **compressionratio** | 0.150276 | -0.178515 | 0.249786 | 0.158414 | 0.181129 | 0.261214 | 0.151362 |
| **horsepower** | -0.015006 | 0.070873 | 0.353294 | 0.552623 | 0.640732 | -0.108802 | 0.750739 |
| **peakrpm** | -0.203789 | 0.273606 | -0.360469 | -0.287242 | -0.220012 | -0.320411 | -0.266243 |
| **citympg** | 0.015940 | -0.035823 | -0.470414 | -0.670909 | -0.642704 | -0.048640 | -0.757414 |
| **highwaympg** | 0.011255 | 0.034606 | -0.544082 | -0.704662 | -0.677218 | -0.107358 | -0.797465 |
| **price** | -0.109093 | -0.079978 | 0.577816 | 0.682920 | 0.759325 | 0.119336 | 0.835305 |

In [8]:
```
plt.figure(figsize=(20, 15))
correlations = df.corr()
sns.heatmap(correlations, cmap="coolwarm", annot=True)
plt.show()
```

```
C:\Users\radhe\AppData\Local\Temp\ipykernel_20496\3130584114.py:2: FutureWarn
ing: 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 specif
y the value of numeric_only to silence this warning.
  correlations = df.corr()
```



| | car_ID | symboling | wheelbase | carlength | carwidth | carheight | curbweight | enginesize | boreratio | stroke | compressionratio | horsepower | peakrpm | citympg | highwaympg | price |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| car_ID | 1 | -0.15 | 0.13 | 0.17 | 0.052 | 0.26 | 0.072 | -0.034 | 0.26 | -0.16 | 0.15 | -0.015 | -0.2 | 0.016 | 0.011 | -0.11 |
| symboling | -0.15 | 1 | -0.53 | -0.36 | -0.23 | -0.54 | -0.23 | -0.11 | -0.13 | -0.0087 | -0.18 | 0.071 | 0.27 | -0.036 | 0.035 | -0.08 |
| wheelbase | 0.13 | -0.53 | 1 | 0.87 | 0.8 | 0.59 | 0.78 | 0.57 | 0.49 | 0.16 | 0.25 | 0.35 | -0.36 | -0.47 | -0.54 | 0.58 |
| carlength | 0.17 | -0.36 | 0.87 | 1 | 0.84 | 0.49 | 0.88 | 0.68 | 0.61 | 0.13 | 0.16 | 0.55 | -0.29 | -0.67 | -0.7 | 0.68 |
| carwidth | 0.052 | -0.23 | 0.8 | 0.84 | 1 | 0.28 | 0.87 | 0.74 | 0.56 | 0.18 | 0.18 | 0.64 | -0.22 | -0.64 | -0.68 | 0.76 |
| carheight | 0.26 | -0.54 | 0.59 | 0.49 | 0.28 | 1 | 0.3 | 0.067 | 0.17 | -0.055 | 0.26 | -0.11 | -0.32 | -0.049 | -0.11 | 0.12 |
| curbweight | 0.072 | -0.23 | 0.78 | 0.88 | 0.87 | 0.3 | 1 | 0.85 | 0.65 | 0.17 | 0.15 | 0.75 | -0.27 | -0.76 | -0.8 | 0.84 |
| enginesize | -0.034 | -0.11 | 0.57 | 0.68 | 0.74 | 0.067 | 0.85 | 1 | 0.58 | 0.2 | 0.029 | 0.81 | -0.24 | -0.65 | -0.68 | 0.87 |
| boreratio | 0.26 | -0.13 | 0.49 | 0.61 | 0.56 | 0.17 | 0.65 | 0.58 | 1 | -0.056 | 0.0052 | 0.57 | -0.25 | -0.58 | -0.59 | 0.55 |
| stroke | -0.16 | -0.0087 | 0.16 | 0.13 | 0.18 | -0.055 | 0.17 | 0.2 | -0.056 | 1 | 0.19 | 0.081 | -0.068 | -0.042 | -0.044 | 0.079 |
| compressionratio | 0.15 | -0.18 | 0.25 | 0.16 | 0.18 | 0.26 | 0.15 | 0.029 | 0.0052 | 0.19 | 1 | -0.2 | -0.44 | 0.32 | 0.27 | 0.068 |
| horsepower | -0.015 | 0.071 | 0.35 | 0.55 | 0.64 | -0.11 | 0.75 | 0.81 | 0.57 | 0.081 | -0.2 | 1 | 0.13 | -0.8 | -0.77 | 0.81 |
| peakrpm | -0.2 | 0.27 | -0.36 | -0.29 | -0.22 | -0.32 | -0.27 | -0.24 | -0.25 | -0.068 | -0.44 | 0.13 | 1 | -0.11 | -0.054 | -0.085 |
| citympg | 0.016 | -0.036 | -0.47 | -0.67 | -0.64 | -0.049 | -0.76 | -0.65 | -0.58 | -0.042 | 0.32 | -0.8 | -0.11 | 1 | 0.97 | -0.69 |
| highwaympg | 0.011 | 0.035 | -0.54 | -0.7 | -0.68 | -0.11 | -0.8 | -0.68 | -0.59 | -0.044 | 0.27 | -0.77 | -0.054 | 0.97 | 1 | -0.7 |
| price | -0.11 | -0.08 | 0.58 | 0.68 | 0.76 | 0.12 | 0.84 | 0.87 | 0.55 | 0.079 | 0.068 | 0.81 | -0.085 | -0.69 | -0.7 | 1 |

In [9]:

```
predict = "price"
data = df[["symboling", "wheelbase", "carlength",
             "carwidth", "carheight", "curbweight",
             "enginesize", "boreratio", "stroke",
             "compressionratio", "horsepower", "peakrpm",
             "citympg", "highwaympg", "price"]]
x = np.array(data.drop([predict], 1))
y = np.array(data[predict])

from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2)

from sklearn.tree import DecisionTreeRegressor
model = DecisionTreeRegressor()
model.fit(xtrain, ytrain)
predictions = model.predict(xtest)

from sklearn.metrics import mean_absolute_error
model.score(xtest, predictions)
```

```
C:\Users\radhe\AppData\Local\Temp\ipykernel_20496\4071005399.py:7: FutureWarn
ing: In a future version of pandas all arguments of DataFrame.drop except for
the argument 'labels' will be keyword-only.
  x = np.array(data.drop([predict], 1))
```

Out[9]:

```
1.0
```

In [ ]: