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

from google.colab import drive
drive.mount("/content/gdrive")
```

```
Mounted at /content/gdrive
```

```
data = pd.read_csv('/content/gdrive/MyDrive/OIBSIP/Car_Price_Prediction.csv')
(data.head())
```

	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewl
0	1	3	alfa-romero giulia	gas	std	two	convertible	
1	2	3	alfa-romero stelvio	gas	std	two	convertible	
2	3	1	alfa-romero Quadrifoglio	gas	std	two	hatchback	
3	4	2	audi 100 ls	gas	std	four	sedan	
4	5	2	audi 100ls	gas	std	four	sedan	

5 rows × 26 columns

```
data.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   fueltype          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  boreratio         205 non-null    float64
 19  stroke            205 non-null    float64
 20  compressionratio  205 non-null    float64
 21  horsepower        205 non-null    int64  
 22  peakrpm           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
```

```
data.describe()
```

	car_ID	symboling	wheelbase	carlength	carwidth	carheight	curbweight	enginesize	boreratio	stroke	compress
count	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	2
mean	103.000000	0.834146	98.756585	174.049268	65.907805	53.724878	2555.565854	126.907317	3.329756	3.255415	
std	59.322565	1.245307	6.021776	12.337289	2.145204	2.443522	520.680204	41.642693	0.270844	0.313597	
min	1.000000	-2.000000	86.600000	141.100000	60.300000	47.800000	1488.000000	61.000000	2.540000	2.070000	

```
print(data.CarName.unique())
```

```
['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']
```

```
print(data.carbody.unique())
```

```
['convertible' 'hatchback' 'sedan' 'wagon' 'hardtop']
```

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

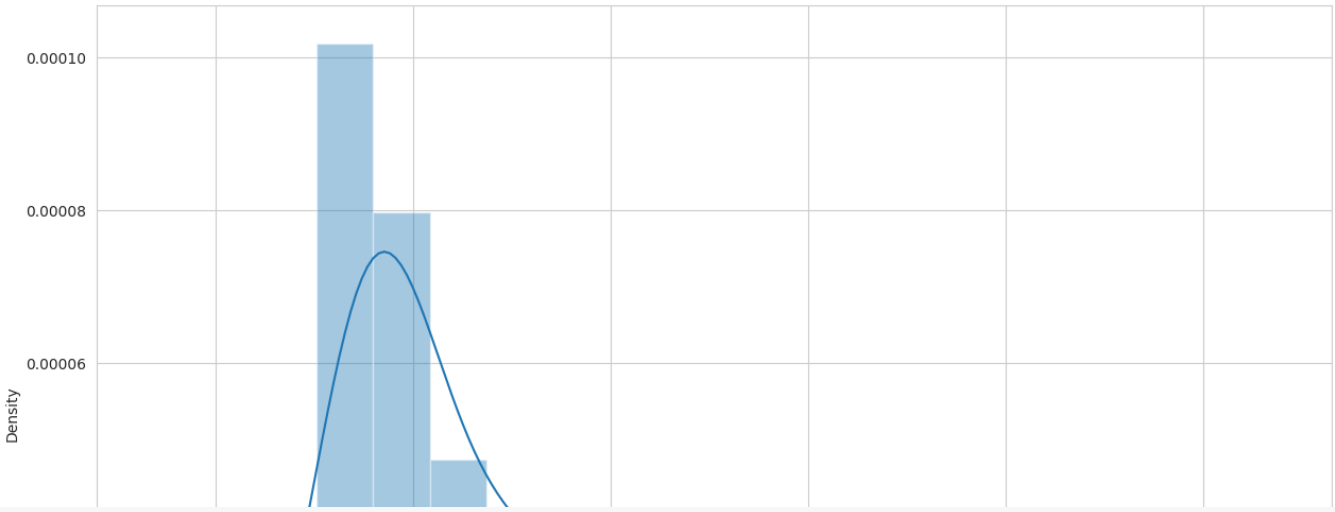
```
<ipython-input-7-304581fcc834>:3: UserWarning:

`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

  sns.distplot(data.price)
```



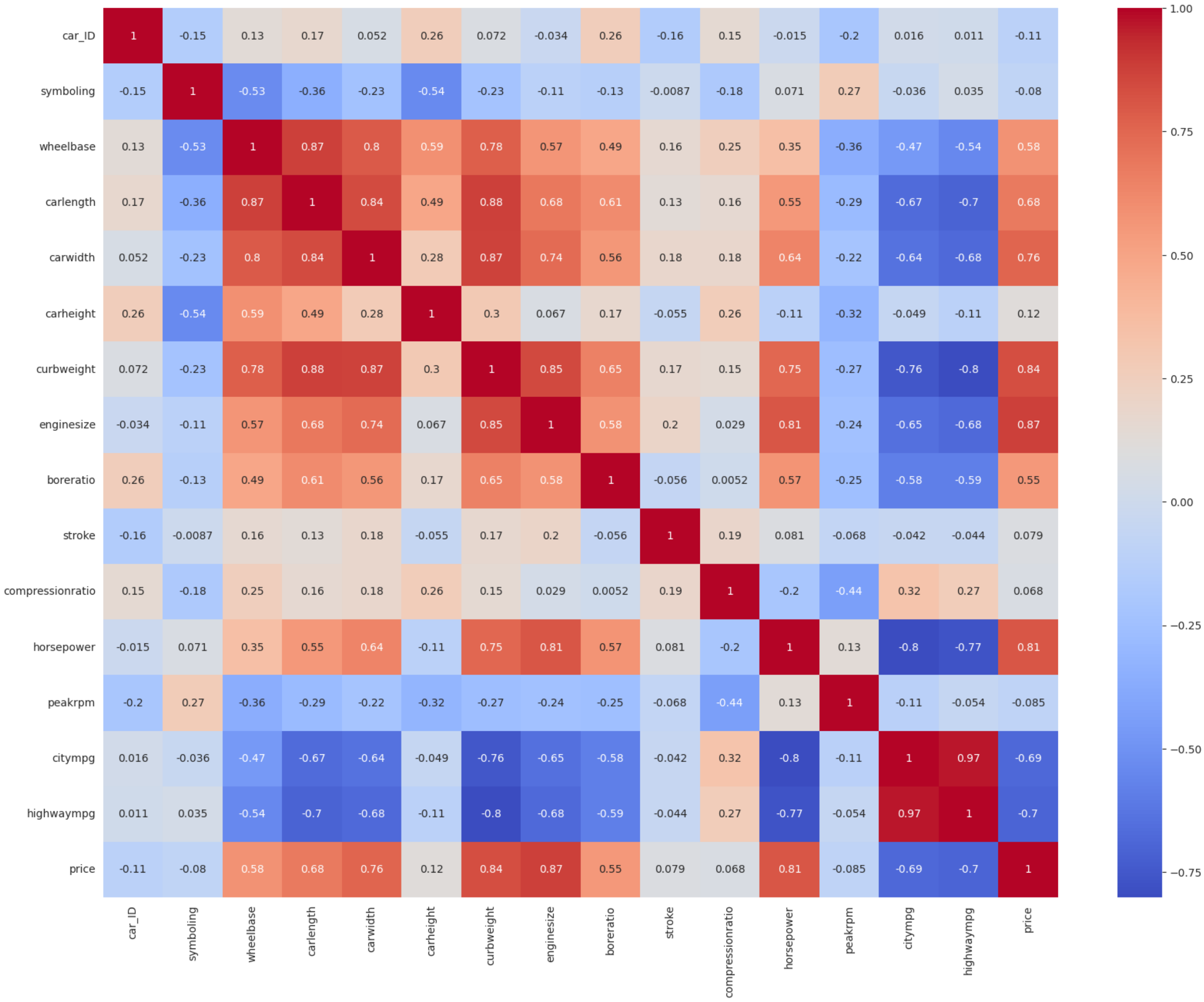
```
data.corr()
```

```
<ipython-input-8-c44ded798807>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
  data.corr()
```

	car_ID	symboling	wheelbase	carlength	carwidth	carheight	curbweight	enginesize	boreratio	stroke	compr
car_ID	1.000000	-0.151621	0.129729	0.170636	0.052387	0.255960	0.071962	-0.033930	0.260064	-0.160824	
symboling	-0.151621	1.000000	-0.531954	-0.357612	-0.232919	-0.541038	-0.227691	-0.105790	-0.130051	-0.008735	
wheelbase	0.129729	-0.531954	1.000000	0.874587	0.795144	0.589435	0.776386	0.569329	0.488750	0.160959	
carlength	0.170636	-0.357612	0.874587	1.000000	0.841118	0.491029	0.877728	0.683360	0.606454	0.129533	
carwidth	0.052387	-0.232919	0.795144	0.841118	1.000000	0.279210	0.867032	0.735433	0.559150	0.182942	
carheight	0.255960	-0.541038	0.589435	0.491029	0.279210	1.000000	0.295572	0.067149	0.171071	-0.055307	
curbweight	0.071962	-0.227691	0.776386	0.877728	0.867032	0.295572	1.000000	0.850594	0.648480	0.168790	
enginesize	-0.033930	-0.105790	0.569329	0.683360	0.735433	0.067149	0.850594	1.000000	0.583774	0.203129	
boreratio	0.260064	-0.130051	0.488750	0.606454	0.559150	0.171071	0.648480	0.583774	1.000000	-0.055909	
stroke	-0.160824	-0.008735	0.160959	0.129533	0.182942	-0.055307	0.168790	0.203129	-0.055909	1.000000	
compressionratio	0.150276	-0.178515	0.249786	0.158414	0.181129	0.261214	0.151362	0.028971	0.005197	0.186110	
horsepower	-0.015006	0.070873	0.353294	0.552623	0.640732	-0.108802	0.750739	0.809769	0.573677	0.080940	
peakrpm	-0.203789	0.273606	-0.360469	-0.287242	-0.220012	-0.320411	-0.266243	-0.244660	-0.254976	-0.067964	
citympg	0.015940	-0.035823	-0.470414	-0.670909	-0.642704	-0.048640	-0.757414	-0.653658	-0.584532	-0.042145	
highwaympg	0.011255	0.034606	-0.544082	-0.704662	-0.677218	-0.107358	-0.797465	-0.677470	-0.587012	-0.043931	
price	-0.109093	-0.079978	0.577816	0.682920	0.759325	0.119336	0.835305	0.874145	0.553173	0.079443	

```
plt.figure(figsize=(20, 15))
correlations = data.corr()
sns.heatmap(correlations, cmap="coolwarm", annot=True)
plt.show()
```

```
<ipython-input-9-555d4168b84a>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
  correlations = data.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

```
predict = "price"
data = data[["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)
```

```
<ipython-input-10-56df943d8afb>:7: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argu
  x = np.array(data.drop([predict], 1))
1.0
```

Colab paid products - Cancel contracts here

✓ 0s completed at 21:30 ● ×