CAR PRICE PREDICTION WITH MACHINE LEARNING

Car price prediction is one of the major research areas in machine learning.So, here I am going to train a model for car price prediction.

Steps to build the model

1.Import the libraries 2.Import Dataset 3.Exploratory Data Analysis 4.Training the model 5.Prediction 6.Model evaluation

1.Import the libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error
```

2.Import the dataset

```
sample = pd.read_csv('CarPrice_Assignment.csv')
```

```
sample.head()
```

```
   car_ID  symboling                   CarName fueltype aspiration
doornumber  \
0       1          3        alfa-romero giulia      gas        std
two
1       2          3       alfa-romero stelvio      gas        std
two
2       3          1  alfa-romero Quadrifoglio      gas        std
two
3       4          2               audi 100 ls      gas        std
four
4       5          2                audi 100ls      gas        std
four

       carbody drivewheel enginelocation  wheelbase  ...
enginesize  \
0  convertible        rwd          front       88.6  ...         130

1  convertible        rwd          front       88.6  ...         130
```

```
2    hatchback        rwd          front       94.5  ...         152

3        sedan        fwd          front       99.8  ...         109

4        sedan        4wd          front       99.4  ...         136


   fuelsystem  boreratio  stroke compressionratio horsepower  peakrpm
citympg  \
0        mpfi       3.47    2.68              9.0        111     5000
21
1        mpfi       3.47    2.68              9.0        111     5000
21
2        mpfi       2.68    3.47              9.0        154     5000
19
3        mpfi       3.19    3.40             10.0        102     5500
24
4        mpfi       3.19    3.40              8.0        115     5500
18

   highwaympg    price
0          27  13495.0
1          27  16500.0
2          26  16500.0
3          30  13950.0
4          22  17450.0

[5 rows x 26 columns]

sample.tail()

     car_ID  symboling          CarName fueltype aspiration doornumber
\
200     201         -1  volvo 145e (sw)      gas        std       four

201     202         -1      volvo 144ea      gas      turbo       four

202     203         -1      volvo 244dl      gas        std       four

203     204         -1        volvo 246   diesel      turbo       four

204     205         -1      volvo 264gl      gas      turbo       four


    carbody drivewheel enginelocation  wheelbase  ...  enginesize
fuelsystem  \
200   sedan        rwd          front      109.1  ...         141
mpfi
201   sedan        rwd          front      109.1  ...         141
```

```
mpfi
202   sedan        rwd          front      109.1  ...         173
mpfi
203   sedan        rwd          front      109.1  ...         145
idi
204   sedan        rwd          front      109.1  ...         141
mpfi

     boreratio  stroke compressionratio horsepower  peakrpm citympg  \
200       3.78    3.15              9.5        114     5400      23
201       3.78    3.15              8.7        160     5300      19
202       3.58    2.87              8.8        134     5500      18
203       3.01    3.40             23.0        106     4800      26
204       3.78    3.15              9.5        114     5400      19

     highwaympg    price
200          28  16845.0
201          25  19045.0
202          23  21485.0
203          27  22470.0
204          25  22625.0

[5 rows x 26 columns]
```

```
sample.head(10)
```

```
   car_ID  symboling                   CarName fueltype aspiration
doornumber  \
0       1          3        alfa-romero giulia      gas        std
two
1       2          3       alfa-romero stelvio      gas        std
two
2       3          1  alfa-romero Quadrifoglio      gas        std
two
3       4          2               audi 100 ls      gas        std
four
4       5          2                audi 100ls      gas        std
four
5       6          2                  audi fox      gas        std
two
6       7          1                audi 100ls      gas        std
four
7       8          1                 audi 5000      gas        std
four
8       9          1                 audi 4000      gas      turbo
four
9      10          0       audi 5000s (diesel)      gas      turbo
two

       carbody drivewheel enginelocation  wheelbase  ...
```

```
enginesize  \
0  convertible        rwd          front       88.6  ...         130

1  convertible        rwd          front       88.6  ...         130

2    hatchback        rwd          front       94.5  ...         152

3        sedan        fwd          front       99.8  ...         109

4        sedan        4wd          front       99.4  ...         136

5        sedan        fwd          front       99.8  ...         136

6        sedan        fwd          front      105.8  ...         136

7        wagon        fwd          front      105.8  ...         136

8        sedan        fwd          front      105.8  ...         131

9    hatchback        4wd          front       99.5  ...         131


   fuelsystem  boreratio  stroke compressionratio horsepower  peakrpm
citympg  \
0        mpfi       3.47    2.68              9.0        111     5000
21
1        mpfi       3.47    2.68              9.0        111     5000
21
2        mpfi       2.68    3.47              9.0        154     5000
19
3        mpfi       3.19    3.40             10.0        102     5500
24
4        mpfi       3.19    3.40              8.0        115     5500
18
5        mpfi       3.19    3.40              8.5        110     5500
19
6        mpfi       3.19    3.40              8.5        110     5500
19
7        mpfi       3.19    3.40              8.5        110     5500
19
8        mpfi       3.13    3.40              8.3        140     5500
17
9        mpfi       3.13    3.40              7.0        160     5500
16

   highwaympg      price
0          27  13495.000
1          27  16500.000
2          26  16500.000
```

```
3          30  13950.000
4          22  17450.000
5          25  15250.000
6          25  17710.000
7          25  18920.000
8          20  23875.000
9          22  17859.167

[10 rows x 26 columns]
```

```
sample.tail(10)
```

```
     car_ID  symboling          CarName fueltype aspiration doornumber
\
195     196         -1      volvo 144ea      gas        std       four

196     197         -2      volvo 244dl      gas        std       four

197     198         -1        volvo 245      gas        std       four

198     199         -2      volvo 264gl      gas      turbo       four

199     200         -1     volvo diesel      gas      turbo       four

200     201         -1  volvo 145e (sw)      gas        std       four

201     202         -1      volvo 144ea      gas      turbo       four

202     203         -1      volvo 244dl      gas        std       four

203     204         -1        volvo 246   diesel      turbo       four

204     205         -1      volvo 264gl      gas      turbo       four


    carbody drivewheel enginelocation  wheelbase  ...  enginesize
fuelsystem  \
195   wagon        rwd          front      104.3  ...         141
mpfi
196   sedan        rwd          front      104.3  ...         141
mpfi
197   wagon        rwd          front      104.3  ...         141
mpfi
198   sedan        rwd          front      104.3  ...         130
mpfi
199   wagon        rwd          front      104.3  ...         130
mpfi
200   sedan        rwd          front      109.1  ...         141
mpfi
```

```
201   sedan        rwd          front      109.1  ...         141
mpfi
202   sedan        rwd          front      109.1  ...         173
mpfi
203   sedan        rwd          front      109.1  ...         145
idi
204   sedan        rwd          front      109.1  ...         141
mpfi

     boreratio  stroke compressionratio horsepower  peakrpm citympg  \
195       3.78    3.15              9.5        114     5400      23
196       3.78    3.15              9.5        114     5400      24
197       3.78    3.15              9.5        114     5400      24
198       3.62    3.15              7.5        162     5100      17
199       3.62    3.15              7.5        162     5100      17
200       3.78    3.15              9.5        114     5400      23
201       3.78    3.15              8.7        160     5300      19
202       3.58    2.87              8.8        134     5500      18
203       3.01    3.40             23.0        106     4800      26
204       3.78    3.15              9.5        114     5400      19

     highwaympg    price
195          28  13415.0
196          28  15985.0
197          28  16515.0
198          22  18420.0
199          22  18950.0
200          28  16845.0
201          25  19045.0
202          23  21485.0
203          27  22470.0
204          25  22625.0

[10 rows x 26 columns]

sample.shape

(205, 26)

sample.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

sample.describe()

           car_ID   symboling   wheelbase   carlength    carwidth
carheight  \
count  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
std     59.322565    1.245307    6.021776   12.337289    2.145204
2.443522
min      1.000000   -2.000000   86.600000  141.100000   60.300000
47.800000
25%     52.000000    0.000000   94.500000  166.300000   64.100000
52.000000
50%    103.000000    1.000000   97.000000  173.200000   65.500000
54.100000
75%    154.000000    2.000000  102.400000  183.100000   66.900000
55.500000
max    205.000000    3.000000  120.900000  208.100000   72.300000
59.800000

        curbweight  enginesize   boreratio      stroke
compressionratio  \
count   205.000000  205.000000  205.000000  205.000000
205.000000
mean   2555.565854  126.907317    3.329756    3.255415
```

```
10.142537
std     520.680204   41.642693    0.270844    0.313597
3.972040
min    1488.000000   61.000000    2.540000    2.070000
7.000000
25%    2145.000000   97.000000    3.150000    3.110000
8.600000
50%    2414.000000  120.000000    3.310000    3.290000
9.000000
75%    2935.000000  141.000000    3.580000    3.410000
9.400000
max    4066.000000  326.000000    3.940000    4.170000
23.000000
```

	horsepower	peakrpm	citympg	highwaympg	price
count	205.000000	205.000000	205.000000	205.000000	205.000000
mean	104.117073	5125.121951	25.219512	30.751220	13276.710571
std	39.544167	476.985643	6.542142	6.886443	7988.852332
min	48.000000	4150.000000	13.000000	16.000000	5118.000000
25%	70.000000	4800.000000	19.000000	25.000000	7788.000000
50%	95.000000	5200.000000	24.000000	30.000000	10295.000000
75%	116.000000	5500.000000	30.000000	34.000000	16503.000000
max	288.000000	6600.000000	49.000000	54.000000	45400.000000

```
sample.isnull().sum()
```

```
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
```

```
sample['car_ID'].unique()
```

```
array([  1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,
13,
        14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,
26,
        27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,  38,
39,
        40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,
52,
        53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,
65,
        66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,
78,
        79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,
91,
        92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 103,
104,
       105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116,
117,
       118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129,
130,
       131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142,
143,
       144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155,
156,
       157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168,
169,
       170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181,
182,
       183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194,
195,
       196, 197, 198, 199, 200, 201, 202, 203, 204, 205], dtype=int64)
```

```
sample['CarName'].unique()
```

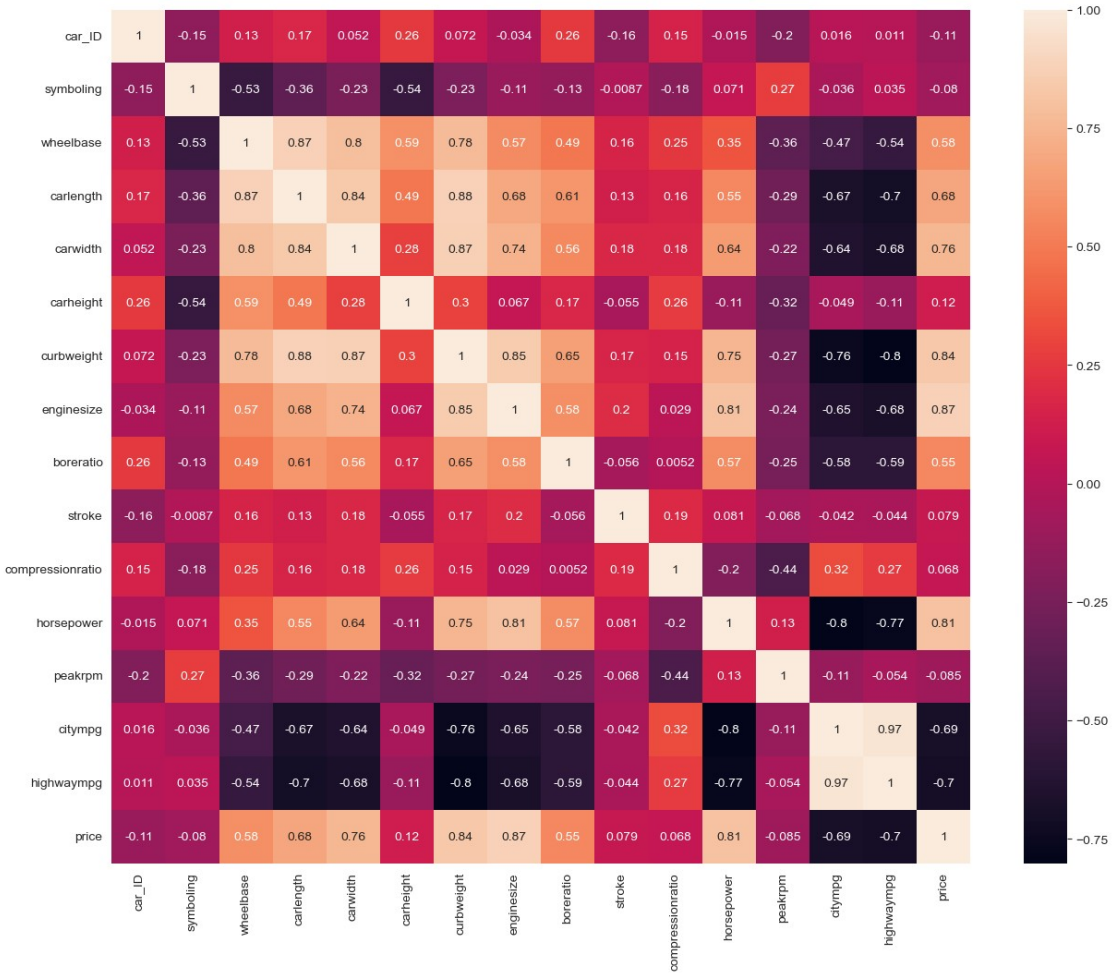
```
array(['alfa-romero giulia', 'alfa-romero stelvio',
       'alfa-romero Quadrifoglio', 'audi 100 ls', 'audi 100ls',
       'audi fox', 'audi 5000', 'audi 4000', 'audi 5000s (diesel)',
       'bmw 320i', 'bmw x1', 'bmw x3', 'bmw z4', 'bmw x4', 'bmw x5',
       'chevrolet impala', 'chevrolet monte 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)
```

3.Exploratory Data Analysis

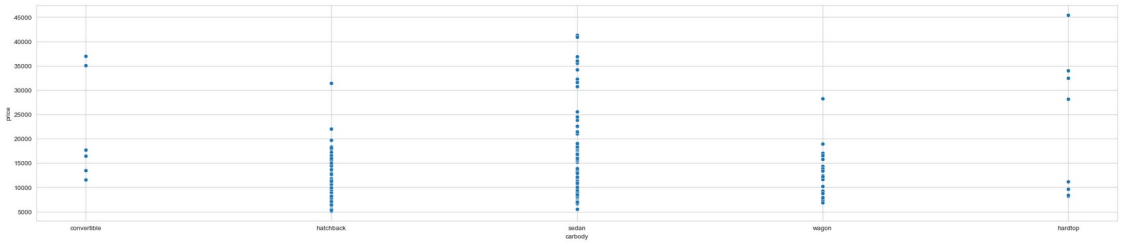
```
plt.figure(figsize=(15, 12))
sns.heatmap(sample.corr(),annot=True)
```

```
<AxesSubplot:>
```




```
plt.rcParams['figure.figsize']=(30,6)
sns.scatterplot(x='carbody', y='price', data=sample)
```

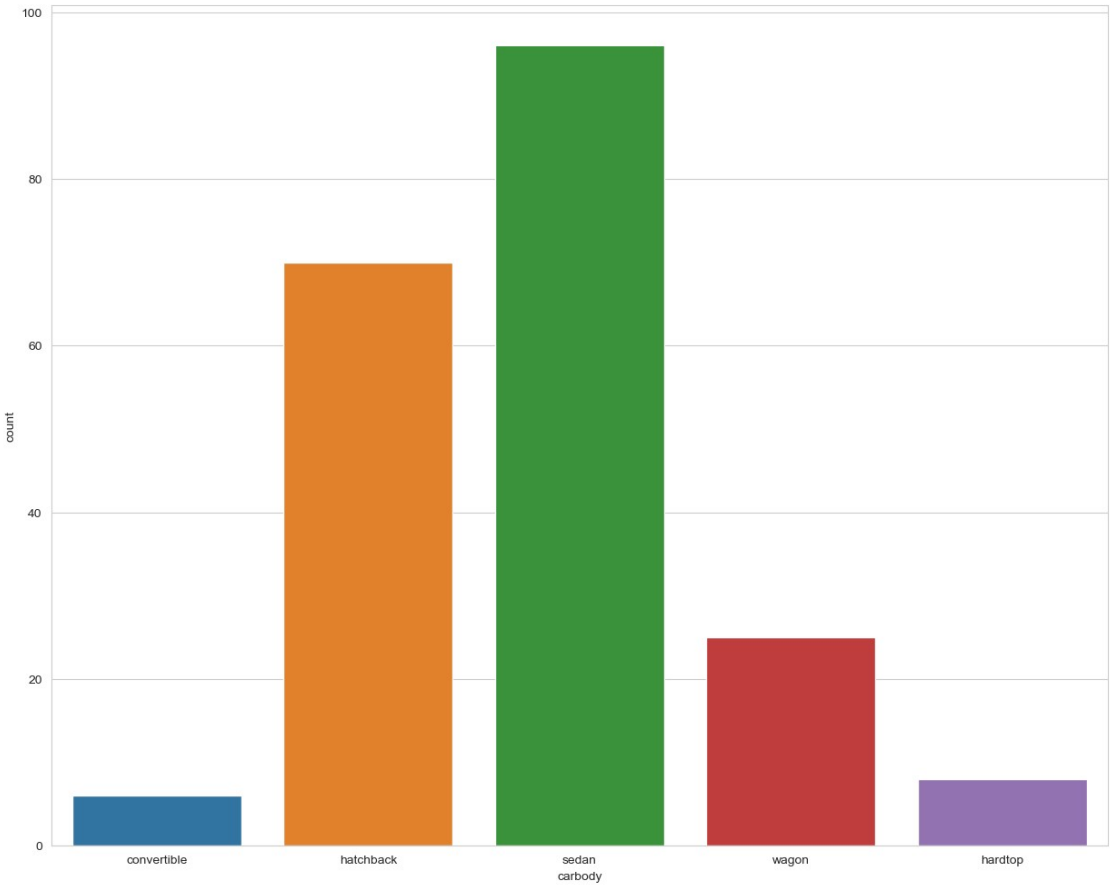
```
<AxesSubplot:xlabel='carbody', ylabel='price'>
```




```
plt.figure(figsize=(15, 12))
sns.countplot(sample['carbody'])
```

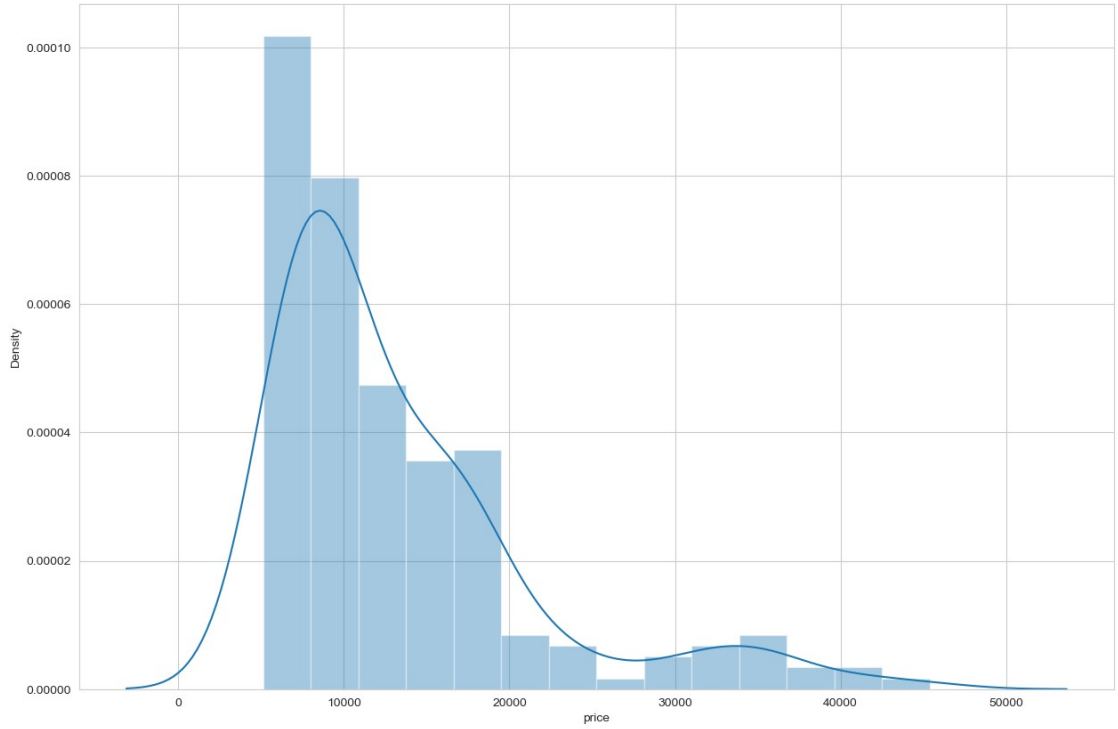
```
C:\Users\priya\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='carbody', ylabel='count'>
```




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

```
C:\Users\priya\anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



4.Training the model

```
x =
sample.drop(columns=["price","CarName","fueltype","aspiration","doornu
mber","carbody","drivewheel","enginelocation","enginetype","cylindernu
mber","fuelsystem",])
y = sample["price"]
x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size=0.2,random_state = 0)

print('Training X Shape:', x_train.shape)
print('Training Y Shape:', y_train.shape)
print('Testing X Shape:', x_test.shape)
print('Testing Y Shape:', y_test.shape)
```

```
Training X Shape: (164, 15)
Training Y Shape: (164,)
Testing X Shape: (41, 15)
Testing Y Shape: (41,)
```

```
model=RandomForestRegressor()

rfe = RandomForestRegressor(n_estimators=40,max_depth=20)
rfe = rfe.fit(x, y)
```

5.Predictions

```
predictions = rfe.predict(x_test)
```

```
errors = abs(predictions - y_test)

print('Mean Absolute Error:', round(np.mean(errors), 2), 'degrees.')
```

```
Mean Absolute Error: 675.3 degrees.
```

6.Model Evaluation

```
mape = 100 * (errors / y_test)

accuracy = 100 - np.mean(mape)
print('Accuracy using RandomForestRegressor:', round(accuracy, 2),
'%.')
```

```
Accuracy using RandomForestRegressor: 95.14 %.
```

CONCLUSION:

Thus, created a car price prediction model using Random Forest Regressor with an accuracy of 95%