In [6]:
```
import pandas as pd
import numpy as np
```

In [7]:
```
df = pd.read_csv('CarPrice.csv')
```

In [8]:
```
df
```

Out[8]:

	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewhe
0	1	3	alfa-romero giulia	gas	std	two	convertible	rv
1	2	3	alfa-romero stelvio	gas	std	two	convertible	rv
2	3	1	alfa-romero Quadrifoglio	gas	std	two	hatchback	rv
3	4	2	audi 100 ls	gas	std	four	sedan	fv
4	5	2	audi 100ls	gas	std	four	sedan	4v
...	...	...	...	...	...	...	...	
200	201	-1	volvo 145e (sw)	gas	std	four	sedan	rv
201	202	-1	volvo 144ea	gas	turbo	four	sedan	rv
202	203	-1	volvo 244dl	gas	std	four	sedan	rv
203	204	-1	volvo 246	diesel	turbo	four	sedan	rv
204	205	-1	volvo 264gl	gas	turbo	four	sedan	rv

205 rows × 26 columns

In [4]:
```
## DATA PREPROCESSING ##
```

In [5]:

```
df.head()
```

Out[5]:

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

5 rows × 26 columns

In [6]:

```
df.tail()
```

Out[6]:

	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel
200	201	-1	volvo 145e (sw)	gas	std	four	sedan	rwd
201	202	-1	volvo 144ea	gas	turbo	four	sedan	rwd
202	203	-1	volvo 244dl	gas	std	four	sedan	rwd
203	204	-1	volvo 246	diesel	turbo	four	sedan	rwd
204	205	-1	volvo 264gl	gas	turbo	four	sedan	rwd

5 rows × 26 columns

In [7]:
```
df.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
```

In [8]:
```
df.shape
```

Out[8]:
```
(205, 26)
```

In [9]:
```
df.columns
```

Out[9]:
```
Index(['car_ID', 'symboling', 'CarName', 'fueltype', 'aspiration',
       'doornumber', 'carbody', 'drivewheel', 'enginelocation', 'wheelba
se',
       'carlength', 'carwidth', 'carheight', 'curbweight', 'enginetype',
       'cylindernumber', 'enginesize', 'fuelsystem', 'boreratio', 'strok
e',
       'compressionratio', 'horsepower', 'peakrpm', 'citympg', 'highwaym
pg',
       'price'],
      dtype='object')
```

In [10]:

```
df.duplicated().sum()
```

Out[10]:

```
0
```

In [11]:

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

Out[11]:

```
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 [4]:

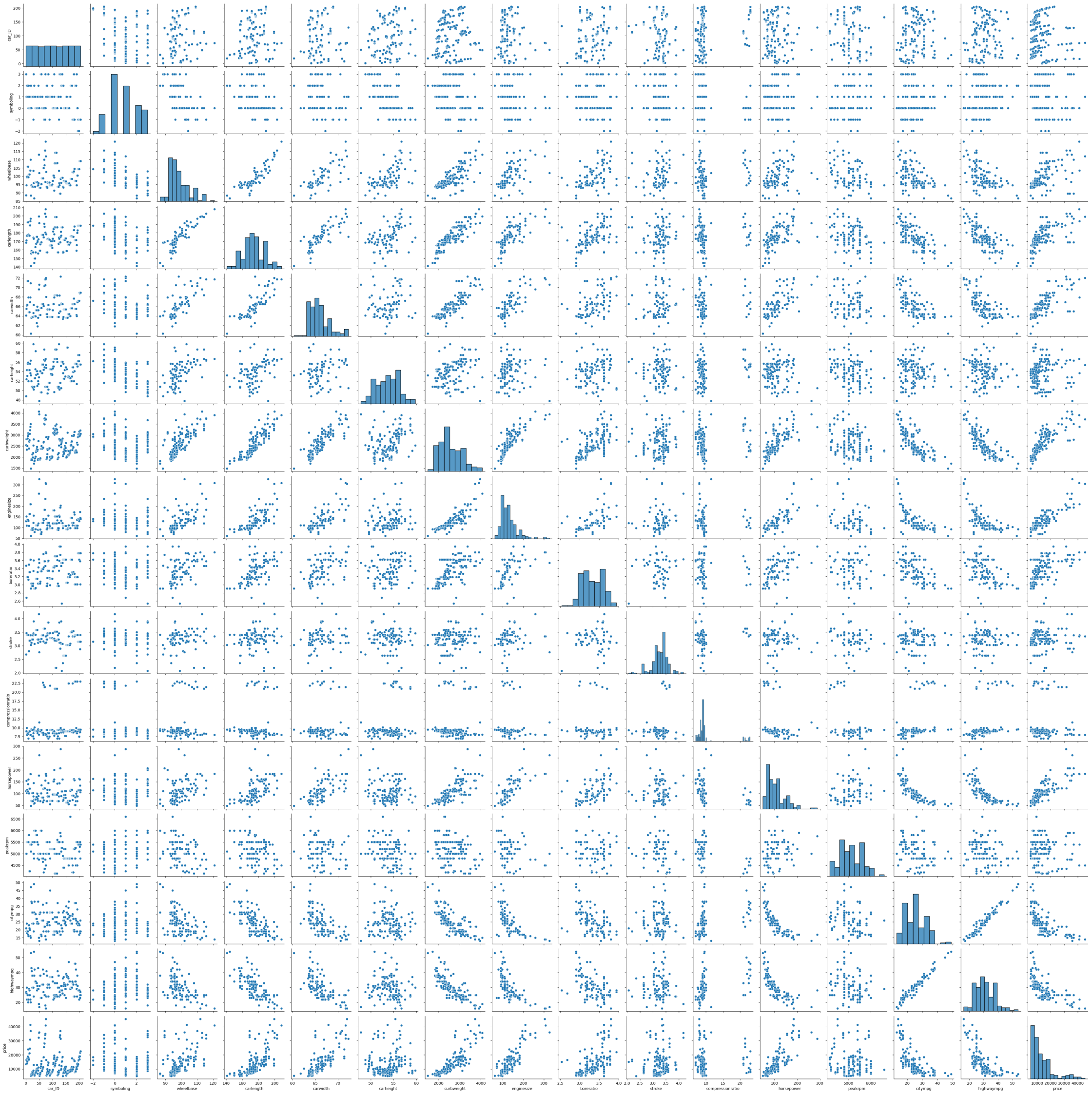
```
## VISUALIZATION ##

import matplotlib.pyplot as plt
import seaborn as sns

sns.pairplot(df)
```

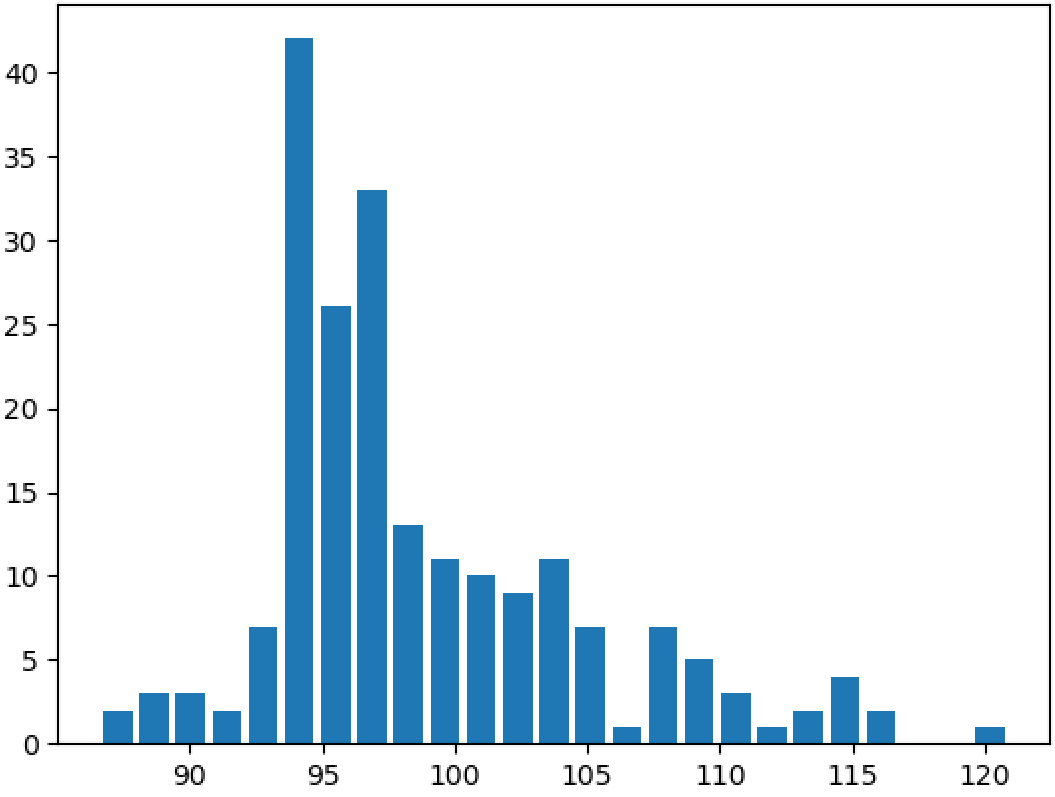
Out[4]:

```
<seaborn.axisgrid.PairGrid at 0x223fd082520>
```



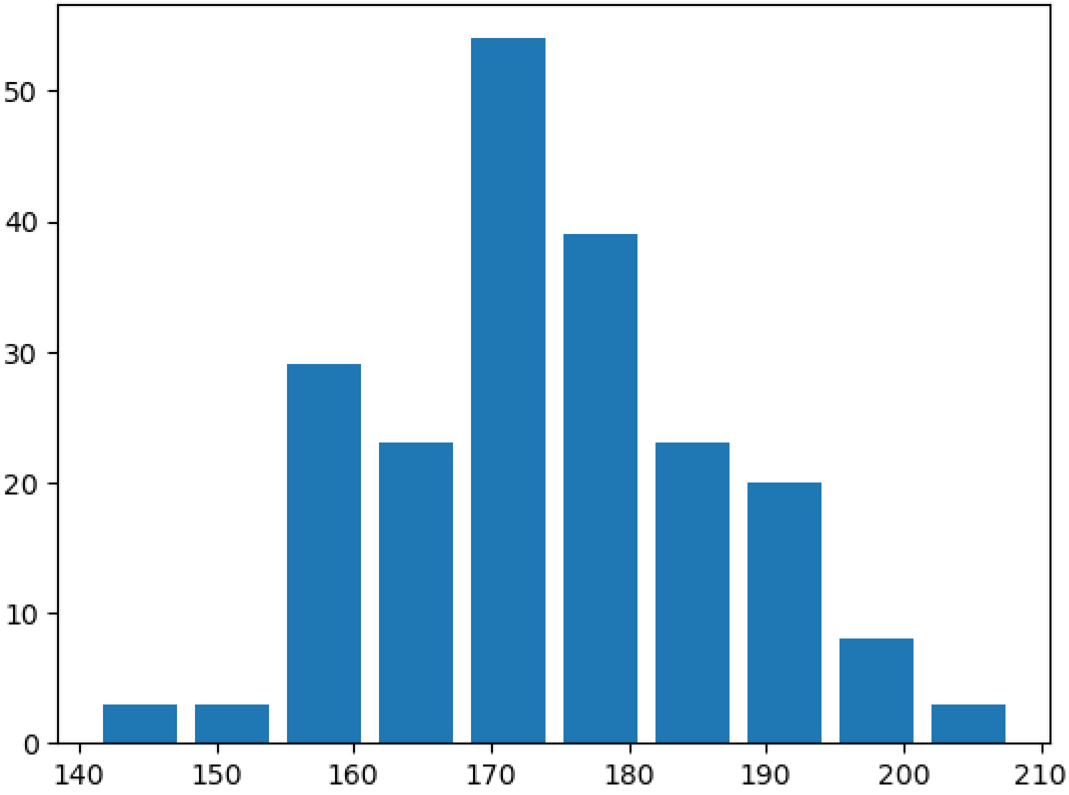
In [6]:

```
plt.hist(df["wheelbase"], bins=25, rwidth=0.8)
plt.show()
```



In [8]:

```
plt.hist(df["carlength"], bins=10, rwidth=0.8)
plt.show()
```



In [13]:

```
print(df.fueltype.value_counts())
print(df.aspiration.value_counts())
print(df.doornumber.value_counts())
print(df.carbody.value_counts())
print(df.drivewheel.value_counts())
print(df.fuelsystem.value_counts())
print(df.cylindernumber.value_counts())
```

```
gas       185
diesel     20
Name: fueltype, dtype: int64
std      168
turbo     37
Name: aspiration, dtype: int64
four    115
two      90
Name: doornumber, dtype: int64
sedan          96
hatchback      70
wagon          25
hardtop         8
convertible     6
Name: carbody, dtype: int64
fwd    120
rwd     76
4wd      9
Name: drivewheel, dtype: int64
mpfi    94
2bbl    66
idi     20
1bbl    11
spdi     9
4bbl     3
mfi      1
spfi     1
Name: fuelsystem, dtype: int64
four      159
six        24
five       11
eight       5
two         4
three       1
twelve      1
Name: cylindernumber, dtype: int64
```

In [14]:

```
df['horsepower'].value_counts()
```

```
Out[14]: 68     19
         70     11
         69     10
         116     9
         110     8
         95      7
         114     6
         160     6
         101     6
         62      6
         88      6
         145     5
         76      5
         97      5
         84      5
         90      5
         82      5
         102     5
         92      4
         111     4
         123     4
         86      4
         207     3
         73      3
         182     3
         121     3
         85      3
         152     3
         176     2
         94      2
         56      2
         112     2
         161     2
         184     2
         155     2
         156     2
         52      2
         100     2
         162     2
         140     1
         115     1
         134     1
         78      1
         142     1
         288     1
         143     1
         48      1
         200     1
         58      1
         55      1
         60      1
         175     1
         154     1
         72      1
         120     1
         64      1
         135     1
```

```
262     1
106     1
Name: horsepower, dtype: int64
```

In [15]:

```
df['stroke'].value_counts()
```

Out[15]:

```
3.400    20
3.230    14
3.150    14
3.030    14
3.390    13
2.640    11
3.290     9
3.350     9
3.460     8
3.110     6
3.270     6
3.410     6
3.070     6
3.580     6
3.190     6
3.500     6
3.640     5
3.520     5
3.860     4
3.540     4
3.470     4
3.255     4
3.900     3
2.900     3
3.100     2
4.170     2
2.800     2
2.190     2
3.080     2
2.680     2
2.360     1
3.160     1
2.070     1
3.210     1
3.120     1
2.760     1
2.870     1
Name: stroke, dtype: int64
```

In [16]:

```
df['compressionratio'].value_counts()
```

Out[16]:

```
9.00     46
9.40     26
8.50     14
9.50     13
9.30     11
8.70      9
8.00      8
9.20      8
7.00      7
8.60      5
21.00     5
8.40      5
7.50      5
23.00     5
9.60      5
21.50     4
7.60      4
10.00     3
22.50     3
8.30      3
8.80      3
7.70      2
8.10      2
9.10      1
9.31      1
7.80      1
9.41      1
21.90     1
22.00     1
22.70     1
10.10     1
11.50     1
Name: compressionratio, dtype: int64
```

In [17]:

```
df['citympg'].value_counts()
```

Out[17]:

```
31    28
19    27
24    22
27    14
17    13
26    12
23    12
21     8
25     8
30     8
38     7
28     7
16     6
37     6
22     4
29     3
15     3
20     3
18     3
14     2
34     1
35     1
32     1
36     1
45     1
13     1
49     1
47     1
33     1
Name: citympg, dtype: int64
```

In [18]:

```
df['highwaympg'].value_counts()
```

Out[18]:

```
25    19
38    17
24    17
30    16
32    16
34    14
37    13
28    13
29    10
33     9
22     8
31     8
23     7
27     5
43     4
42     3
26     3
41     3
19     2
39     2
18     2
16     2
20     2
36     2
47     2
46     2
54     1
17     1
53     1
50     1
Name: highwaympg, dtype: int64
```

In [19]:

```
## CHANGING THE CATEGORICAL ATTRIBUTES INTO NUMERIC DATA FOR BETTER ANALYS

df.replace({'fueltype':{'gas':0,'diesel':1}},inplace=True)
df.replace({'aspiration':{'std':0,'turbo':1}},inplace=True)
df.replace({'doornumber':{'two':0,'four':1}},inplace=True)
df.replace({'carbody':{'convertible':0,'hatchback':1,'sedan':2, 'wagon':3
df.replace({'drivewheel':{'rwd':0,'fwd':1,'4wd':2}},inplace=True)
df.replace({'fuelsystem':{'mpfi':0,'2bbl':1,'1bbl':2,'mfi':3, 'spf1':4, '
```

In [9]:

```
## hot encoding ##
df = pd.get_dummies(df,drop_first=True)
df
```

Out[9]:

	car_ID	symboling	wheelbase	carlength	carwidth	carheight	curbweight	enginesize
0	1	3	88.6	168.8	64.1	48.8	2548	130
1	2	3	88.6	168.8	64.1	48.8	2548	130
2	3	1	94.5	171.2	65.5	52.4	2823	152
3	4	2	99.8	176.6	66.2	54.3	2337	109
4	5	2	99.4	176.6	66.4	54.3	2824	136
...	...	...	...	...	...	...	...	...
200	201	-1	109.1	188.8	68.9	55.5	2952	141
201	202	-1	109.1	188.8	68.8	55.5	3049	141
202	203	-1	109.1	188.8	68.9	55.5	3012	173
203	204	-1	109.1	188.8	68.9	55.5	3217	145
204	205	-1	109.1	188.8	68.9	55.5	3062	141

205 rows × 191 columns

In [10]:

```
## Splitting the data ##

x = df.drop(['price'], axis=1)
y = df['price']
print(len(x), len(y))
print(x)
print(y)
print(x.shape)
print(y.shape)
```

```
205 205
     car_ID  symboling  wheelbase  carlength  carwidth  carheight  curbw
eight  \
0         1          3       88.6      168.8      64.1       48.8
2548
1         2          3       88.6      168.8      64.1       48.8
2548
2         3          1       94.5      171.2      65.5       52.4
2823
3         4          2       99.8      176.6      66.2       54.3
2337
4         5          2       99.4      176.6      66.4       54.3
2824
..      ...        ...        ...        ...       ...        ...
...
200     201         -1      109.1      188.8      68.9       55.5
2952
201     202         -1      109.1      188.8      68.8       55.5
3049
202     203         -1      109.1      188.8      68.9       55.5
3012
203     204         -1      109.1      188.8      68.9       55.5
3217
204     205         -1      109.1      188.8      68.9       55.5
3062

     enginesize  boreratio  stroke  ...  cylindernumber_three  \
0           130       3.47    2.68  ...                     0
1           130       3.47    2.68  ...                     0
2           152       2.68    3.47  ...                     0
3           109       3.19    3.40  ...                     0
4           136       3.19    3.40  ...                     0
..          ...        ...     ...  ...                   ...
200         141       3.78    3.15  ...                     0
201         141       3.78    3.15  ...                     0
202         173       3.58    2.87  ...                     0
203         145       3.01    3.40  ...                     0
204         141       3.78    3.15  ...                     0

     cylindernumber_twelve  cylindernumber_two  fuelsystem_2bbl  \
0                        0                   0                0
1                        0                   0                0
2                        0                   0                0
3                        0                   0                0
4                        0                   0                0
..                     ...                 ...              ...
200                      0                   0                0
201                      0                   0                0
202                      0                   0                0
203                      0                   0                0
204                      0                   0                0

     fuelsystem_4bbl  fuelsystem_idi  fuelsystem_mfi  fuelsystem_mpfi  \
0                  0               0               0                1
1                  0               0               0                1
2                  0               0               0                1
3                  0               0               0                1
```

```
4                  0               0               0                1
..               ...             ...             ...              ...
200                0               0               0                1
201                0               0               0                1
202                0               0               0                1
203                0               1               0                0
204                0               0               0                1

     fuelsystem_spdi  fuelsystem_spfi
0                  0                0
1                  0                0
2                  0                0
3                  0                0
4                  0                0
..               ...              ...
200                0                0
201                0                0
202                0                0
203                0                0
204                0                0

[205 rows x 190 columns]
0      13495.0
1      16500.0
2      16500.0
3      13950.0
4      17450.0
        ...
200    16845.0
201    19045.0
202    21485.0
203    22470.0
204    22625.0
Name: price, Length: 205, dtype: float64
(205, 190)
(205,)
```

In [11]:

```
## Training and Test Data ##
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test=train_test_split(x,y,test_size=1/10,rand
```

In [12]:

```
## Linear Regression ##

from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(x_train,y_train)
```

Out[12]:

```
LinearRegression()
```

In [24]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train, y_train)
```

Out[24]:
```
LinearRegression()
```

In [25]:
```
t_data_predic = regressor.predict(x_train)
```

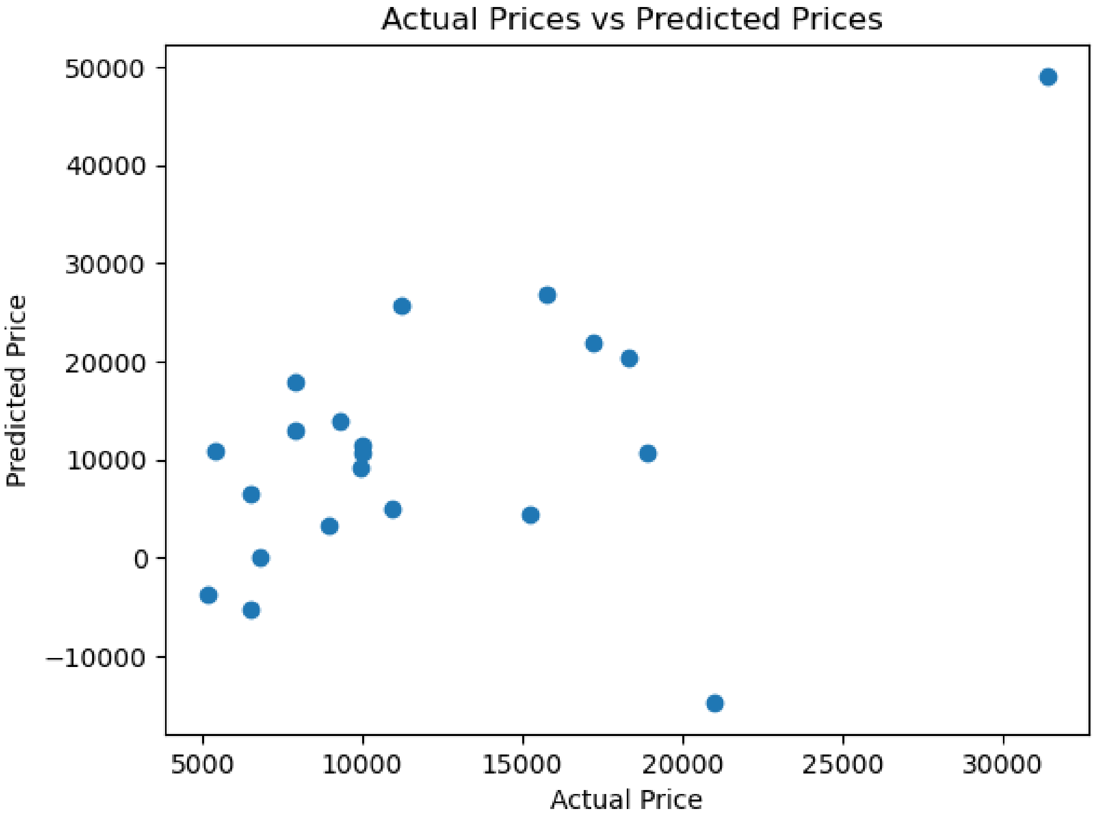
In [26]:
```
## Error Calculation ##

from sklearn import metrics
error_score = metrics.r2_score(y_train, t_data_predic)
print("R squared Error : ", error_score)
```

```
R squared Error :  0.9982334777472928
```

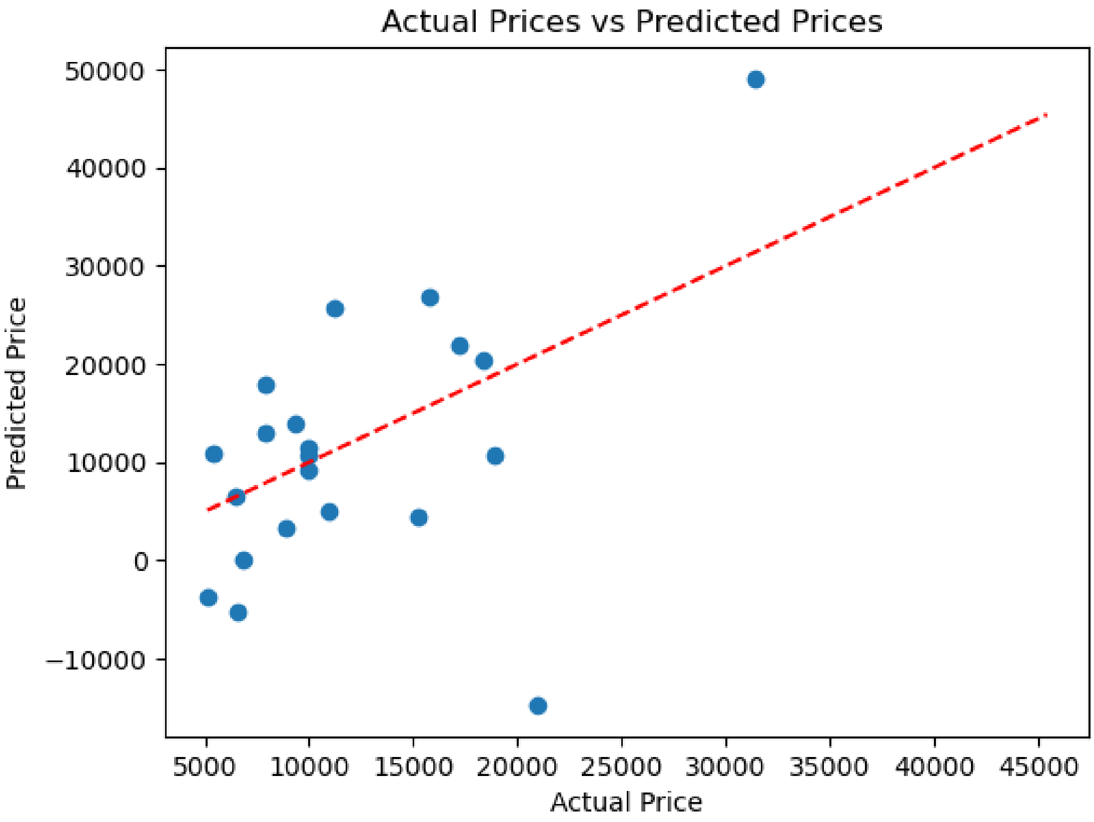
In [28]:
```
 ## prediction on Test data ##
t_data_predic = regressor.predict(x_test)
```

In [30]:
```
plt.scatter(y_test, t_data_predic)
plt.xlabel("Actual Price")
plt.ylabel("Predicted Price")
plt.title(" Actual Prices vs Predicted Prices")
plt.show()
```



In [31]:

```
plt.scatter(y_test, t_data_predic)
plt.xlabel("Actual Price")
plt.ylabel("Predicted Price")
plt.title(" Actual Prices vs Predicted Prices")
plt.plot([min(y_train), max(y_train)], [min(y_train), max(y_train)], 'r--
plt.show()
```



In [32]:

```
b = regressor.coef_
print("Coefficient :",b)
```

```
Coefficient : [ 1.96509785e+02  6.44936604e+02  6.43751612e+03 -1.846823
73e+02
  9.62864785e+02  3.51905038e+02  1.47578359e+02 -2.76898277e+02
  1.30471318e+03 -4.90883212e+02  1.01640031e+01  2.07277966e+01
 -2.61585478e+03 -1.46815247e+03 -9.51029627e+02 -2.58646302e+00
  2.42849538e+00  4.72045187e+02 -2.83591892e+02  2.23256072e+04
  1.60797850e+04  1.88882752e+04  2.16973985e+04  7.65695439e+03
  1.02018548e+04 -2.04272510e-09  5.52651730e+03 -3.35398909e-09
  2.37820393e+04  6.77573553e-10  3.50549631e+04  3.68539059e+04
  4.56932926e+04  3.43660050e+04  1.89276813e+03  2.29997800e+03
  7.59529123e+03  1.14404830e+02  2.14823528e+03  1.22054853e+04
  3.46955061e+03 -4.04518514e+03 -3.54134500e-11  9.33059554e+03
  9.01911563e+03  9.92581143e+03  7.63789167e+03  6.91740145e+03
  9.24904326e+03  4.63287726e+03  1.06968981e+04  1.42257065e+04
  7.39882614e+03  7.34533606e+03  2.02150903e+04  8.09450285e-10
  1.48353204e+04  1.24647316e+04  6.49764458e+03  1.56359652e+04
  1.57687221e+04  1.22894361e+04  1.61300267e+04  6.51504408e+03
  1.38243195e-10  1.06638524e+04  2.19892515e+04  1.88857612e+04
  2.13645926e+03  6.23869197e+03  4.12374372e+03  5.87180380e+03
  1.24491438e+04  6.07284177e+03  6.44050795e+03  4.32427242e+03
  7.00253541e+03  6.15724511e+03  1.40018140e+04 -3.27418093e-11
  5.56319813e+03 -2.10148582e+03 -3.23059905e+03  3.29237082e-10
 -2.72483607e+03 -3.98786800e+03 -3.48772963e+03 -3.39093791e+03
 -2.16371492e+03 -1.64521400e+03 -6.96119570e+02 -1.02350858e+03
 -2.73538678e+02 -3.50329927e+03 -9.36155914e+02  1.87282882e+02
  3.21383449e+02 -1.99899510e+03 -9.72803154e+02 -8.19781102e+02
 -7.49423634e-10  5.89554629e+02 -3.24370948e+03 -1.10929919e+03
 -2.15366884e+02  9.72617050e+01 -1.24322448e+03 -9.65972082e+03
 -1.08984822e+04 -1.05788393e+04 -1.18093123e+04 -9.08037847e+03
 -4.41656453e+03  5.61981388e+03  7.80571692e+03  6.92330409e+03
 -1.79391100e+03 -1.52837459e-09 -7.63639974e+03 -3.74613207e+03
 -2.66239910e+03 -5.25487633e+03 -5.66331032e+03 -2.34515357e+03
 -3.20881020e+03 -3.84748368e+03 -5.39207732e+03 -2.54882668e+03
 -5.36876918e+03 -2.18861445e+03 -1.77123462e+04 -1.57492574e+04
 -1.87165226e+04 -1.40786664e+04 -1.44430878e+04  4.69055864e-10
 -1.42069626e+04 -1.85044657e+04 -1.46322479e+04 -1.08615203e+04
 -1.51889079e+04 -1.56209964e+04 -1.23557763e+04 -1.42267862e+04
 -1.23749457e+04 -8.20364221e-10 -1.36697054e-09 -1.80671048e+04
 -1.74424691e+04 -1.91456875e+04 -1.72969657e+04 -1.92334812e+04
 -2.83890813e+04 -1.65182732e+04 -1.90617839e+04 -1.86088455e+04
 -1.69993612e+04 -1.79981202e+04 -1.51780806e+04 -1.50203418e+04
 -1.19135692e+04 -1.37402866e+04 -1.35996658e+04 -1.97564832e+04
 -1.95290640e+04 -1.73213477e+03 -1.20782794e+03 -2.20250857e+03
 -1.47386125e+03  2.03488349e+04  0.00000000e+00 -5.71433833e+03
 -2.69459650e+03 -1.02142105e+04  2.12168539e+03 -2.84346726e+03
  2.77294682e+03 -6.79229506e+03 -1.31774704e+04 -1.70530257e-13
  2.13645926e+03 -2.84346726e+03  2.12654196e+03 -5.77284403e+03
  4.63287726e+03  6.43751612e+03 -8.91630903e+03  1.49167956e+03
  1.27849341e+03]
```

In [33]:

```
a = regressor.intercept_
print("Intercept :",a)
```

```
Intercept : -45068.74680415792
```