Car Price Prediction

import required Libraries

In [1]:

```
import csv
import pandas as pd
import numpy as np

import requests
from bs4 import BeautifulSoup

import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make_column_transformer
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score
```

Web Scraper to get car price, name, mile, body style, year on truecar website

In [2]:

```
name = []
price = []
mile = []
style = []
year = []

make = ['audi' , 'bmw', 'ferrari', 'jeep', 'maserati']
for i in make:
    page = 1
    while page != 5:
        url = (f"https://www.truecar.com/used-cars-for-sale/listings/{i}//?page
        print(url)
        response = requests.get(url)

        # Web scraping on truecar website(get model,price,mile)
        soup = BeautifulSoup(response.text, 'html.parser')
        for span in soup.find_all('span', attrs={'class':"vehicle-header-make-m
            name.append(span.get_text())

        for div in soup.find_all('div', attrs={'class':"heading-3 margin-y-1 fo
            price.append(div.get_text())

        for div in soup.find_all('div', attrs={'class':"d-flex w-100 justify-co
            mile.append(div.get_text())

        for div in soup.find_all('div', attrs={'data-test':"vehicleCardTrim"}):
            style.append(div.get_text())

        for span in soup.find_all('span', attrs={'class':"vehicle-card-year fon
            year.append(span.get_text())

        page = page + 1
```

```
https://www.truecar.com/used-cars-for-sale/listings/audi//?page=
1 (https://www.truecar.com/used-cars-for-sale/listings/audi//?pa
ge=1)
https://www.truecar.com/used-cars-for-sale/listings/audi//?page=
2 (https://www.truecar.com/used-cars-for-sale/listings/audi//?pa
ge=2)
https://www.truecar.com/used-cars-for-sale/listings/audi//?page=
3 (https://www.truecar.com/used-cars-for-sale/listings/audi//?pa
ge=3)
https://www.truecar.com/used-cars-for-sale/listings/audi//?page=
4 (https://www.truecar.com/used-cars-for-sale/listings/audi//?pa
ge=4)
https://www.truecar.com/used-cars-for-sale/listings/bmw//?page=1
(https://www.truecar.com/used-cars-for-sale/listings/bmw//?page=
1)
```

```
https://www.truecar.com/used-cars-for-sale/listings/bmw//?page=2 (https://www.truecar.com/used-cars-for-sale/listings/bmw//?page=2)
https://www.truecar.com/used-cars-for-sale/listings/bmw//?page=3 (https://www.truecar.com/used-cars-for-sale/listings/bmw//?page=3)
https://www.truecar.com/used-cars-for-sale/listings/bmw//?page=4 (https://www.truecar.com/used-cars-for-sale/listings/bmw//?page=4)
https://www.truecar.com/used-cars-for-sale/listings/ferrari//?page=1 (https://www.truecar.com/used-cars-for-sale/listings/ferrari//?page=1)
https://www.truecar.com/used-cars-for-sale/listings/ferrari//?page=2 (https://www.truecar.com/used-cars-for-sale/listings/ferrari//?page=2)
https://www.truecar.com/used-cars-for-sale/listings/ferrari//?page=3 (https://www.truecar.com/used-cars-for-sale/listings/ferrari//?page=3)
https://www.truecar.com/used-cars-for-sale/listings/ferrari//?page=4 (https://www.truecar.com/used-cars-for-sale/listings/ferrari//?page=4)
https://www.truecar.com/used-cars-for-sale/listings/jeep//?page=1 (https://www.truecar.com/used-cars-for-sale/listings/jeep//?page=1)
https://www.truecar.com/used-cars-for-sale/listings/jeep//?page=2 (https://www.truecar.com/used-cars-for-sale/listings/jeep//?page=2)
https://www.truecar.com/used-cars-for-sale/listings/jeep//?page=3 (https://www.truecar.com/used-cars-for-sale/listings/jeep//?page=3)
https://www.truecar.com/used-cars-for-sale/listings/jeep//?page=4 (https://www.truecar.com/used-cars-for-sale/listings/jeep//?page=4)
https://www.truecar.com/used-cars-for-sale/listings/maserati//?page=1 (https://www.truecar.com/used-cars-for-sale/listings/maserati//?page=1)
https://www.truecar.com/used-cars-for-sale/listings/maserati//?page=2 (https://www.truecar.com/used-cars-for-sale/listings/maserati//?page=2)
https://www.truecar.com/used-cars-for-sale/listings/maserati//?page=3 (https://www.truecar.com/used-cars-for-sale/listings/maserati//?page=3)
https://www.truecar.com/used-cars-for-sale/listings/maserati//?page=4 (https://www.truecar.com/used-cars-for-sale/listings/maserati//?page=4)
```

Convert name list to two model and company lists

In [3]:

```
model = []
company = []

for i in name:
    company.append(i.split(' ' , 1)[0])
    model.append(i.split(' ' , 1)[1])
```

Convert all lists to one list

In [4]:

```
rows = [list(x) for x in zip(company, model, price, mile, style, year)]
print(rows)
```

```
'2019'], ['Audi', 'A4', '$13,499', '105,225 miles', 'Premium S
edan 2.0T quattro Automatic', '2014'], ['Audi', 'Q5', '$35,00
0', '31,138 miles', 'Premium Plus', '2018'], ['Audi', 'Q5',
'$29,994', '38,684 miles', 'Premium Plus', '2018'], ['Audi',
'S6', '$25,000', '91,200 miles', 'Prestige Sedan', '2013'],
['Audi', 'A3', '$14,999', '129,508 miles', 'Prestige Sedan 2.0
T quattro', '2015'], ['Audi', 'Q7', '$28,010', '75,000 miles',
'Premium Plus 3.0S TFSI', '2017'], ['Audi', 'A5', '$17,494',
'112,801 miles', 'Premium Plus Coupe Automatic', '2016'], ['Au
di', 'S5', '$28,399', '69,178 milesDiscount Available', 'Premi
um Plus Coupe Automatic', '2015'], ['Audi', 'Q5', '$40,711',
'25,392 milesUpfront Price Available', 'Prestige Summer of Aud
i', '2018'], ['Audi', 'Q5', '$31,523', '59,254 milesDiscount A
vailable', 'Premium Plus', '2018'], ['Audi', 'Q5', '$34,000',
'40,133 milesDiscount Available', 'Premium Plus', '2018'], ['A
udi', 'Q5', '$53,750', '5,380 miles', 'Prestige', '2019'], ['A
udi', 'Q5', '$36,969', '31,984 miles', 'Premium Plus', '201
9'], ['Audi', 'A8', '$19,750', '93,984 milesUpfront Price Avai
lable', 'L 4.0T', '2014'], ['Audi', 'A4', '$15,999', '52,500 m
iles', 'Premium Plus Sedan 2.0T quattro Automatic', '2013'],
```

Create a csv file and write data to it

In [5]:

```
column = ['company', 'model', 'price', 'mile', 'style', 'year']
rows = [list(x) for x in zip(company, model, price, mile, style, year)]
with open('cars_info.csv', 'w') as f:
    write = csv.writer(f)
    write.writerow(column)
    write.writerows(rows)
```

Read dataset

In [6]:

```
dataset = pd.read_csv("cars_info.csv")
dataset.head()
```

Out[6]:

	company	model	price	mile	style	year
0	Audi	Q5	$37,998	31,188 miles	Premium	2019
1	Audi	S4	$23,880	90,397 milesDiscount Available	Prestige Sedan S tronic	2013
2	Audi	A3	$32,604	20,259 miles	Premium Sedan 2.0 quattro	2018
3	Audi	Q5	$28,400	64,995 miles	Premium Plus	2018
4	Audi	Q7	$20,974	64,340 milesDiscount Available	Premium Plus 3.0T quattro	2015

data normalization file

In [7]:

```
dataset['price'] = [i[1:].replace(',', '') for i in dataset['price']]
dataset['price']=dataset['price'].astype(int)
```

In [8]:

```
dataset['mile'] = [i.split(' ', 1)[0].replace(',', '') for i in dataset['mile']
dataset['mile']=dataset['mile'].astype(int)
```

In [9]:

```
dataset['style'] = dataset['style'].str.split().str.slice(start=0,stop=3).str.j
```

Number of rows and columns of the file

In [10]:

```
dataset.shape
```

Out[10]:

```
(615, 6)
```

Show file

In [11]:

```
dataset
```

Out[11]:

	company	model	price	mile	style	year
0	Audi	Q5	37998	31188	Premium	2019
1	Audi	S4	23880	90397	Prestige Sedan S	2013
2	Audi	A3	32604	20259	Premium Sedan 2.0	2018
3	Audi	Q5	28400	64995	Premium Plus	2018
4	Audi	Q7	20974	64340	Premium Plus 3.0T	2015
...	...	...	...	...	...	...
610	Maserati	Ghibli	31999	25273	Sedan RWD	2015
611	Maserati	Ghibli	34888	39849	S Q4 AWD	2016
612	Maserati	Ghibli	27494	75864	S Q4 AWD	2014
613	Maserati	Levante	64984	5886	3.0L	2020
614	Maserati	Ghibli	40599	35162	S Q4 AWD	2018

615 rows × 6 columns

In [12]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 615 entries, 0 to 614
Data columns (total 6 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   company  615 non-null    object
 1   model    615 non-null    object
 2   price    615 non-null    int32 
 3   mile     615 non-null    int32 
 4   style    615 non-null    object
 5   year     615 non-null    int64 
dtypes: int32(2), int64(1), object(3)
memory usage: 24.1+ KB
```

In [13]:

```
dataset.describe(include = 'all')
```

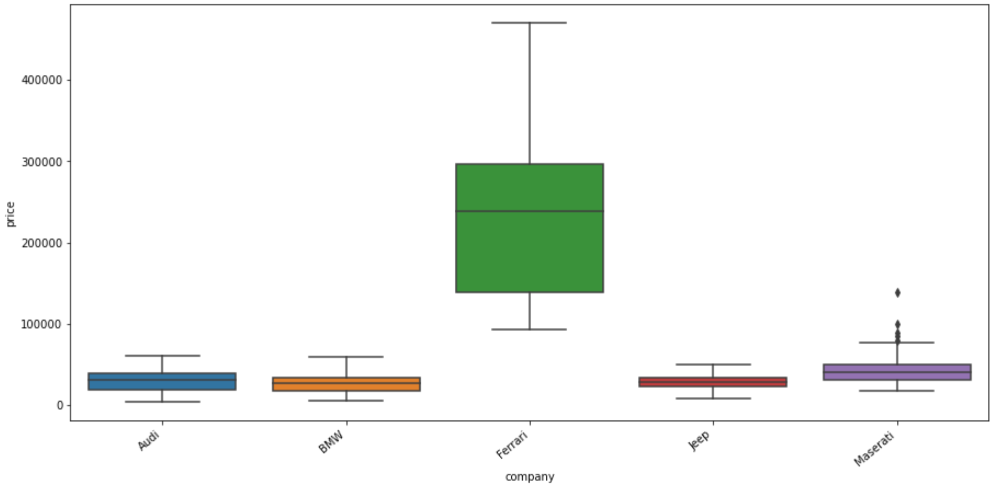
Out[13]:

	company	model	price	mile	style	year
count	615	615	615.000000	615.000000	615	615.000000
unique	5	56	NaN	NaN	153	NaN
top	Jeep	Q5	NaN	NaN	Convertible	NaN
freq	123	57	NaN	NaN	42	NaN
mean	NaN	NaN	72457.479675	50483.240650	NaN	2016.082927
std	NaN	NaN	92765.615885	38790.969328	NaN	3.696722
min	NaN	NaN	3999.000000	70.000000	NaN	2000.000000
25%	NaN	NaN	24991.500000	21585.000000	NaN	2014.000000
50%	NaN	NaN	34695.000000	41290.000000	NaN	2018.000000
75%	NaN	NaN	57223.000000	70437.500000	NaN	2019.000000
max	NaN	NaN	469880.000000	199926.000000	NaN	2022.000000

relationship of Company with Price

In [14]:

```
plt.subplots(figsize = (15,7))
ax = sns.boxplot(x = 'company',y ='price',data = dataset)
ax.set_xticklabels(ax.get_xticklabels(), rotation = 40, ha = 'right')
plt.show()
```

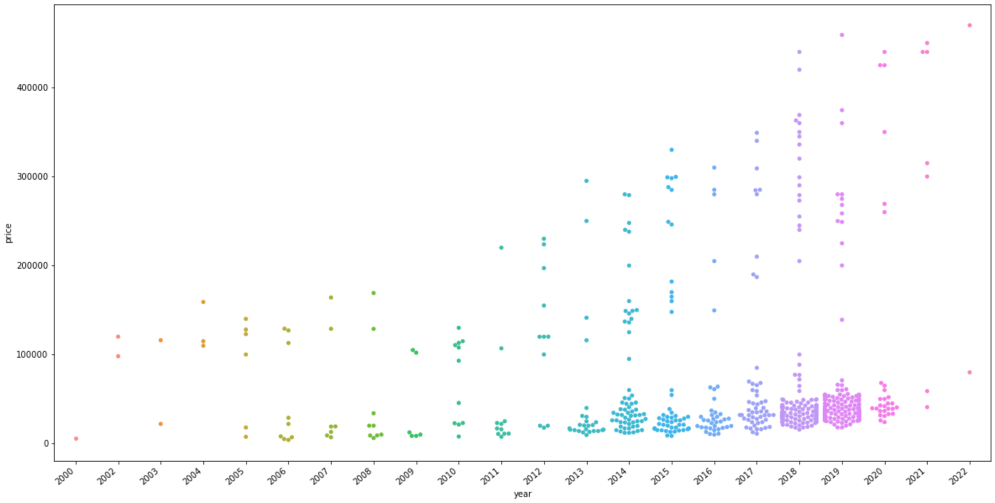


Relationship of year with Price

In [15]:

```
plt.subplots(figsize = (20,10))
ax=sns.swarmplot(x = 'year',y = 'price',data = dataset)
ax.set_xticklabels(ax.get_xticklabels(), rotation = 40, ha = 'right')
plt.show()
```

```
C:\Users\negin\anaconda3\lib\site-packages\seaborn\categorical.p
y:1296: UserWarning: 8.0% of the points cannot be placed; you ma
y want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\negin\anaconda3\lib\site-packages\seaborn\categorical.p
y:1296: UserWarning: 9.8% of the points cannot be placed; you ma
y want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\negin\anaconda3\lib\site-packages\seaborn\categorical.p
y:1296: UserWarning: 42.5% of the points cannot be placed; you m
ay want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\negin\anaconda3\lib\site-packages\seaborn\categorical.p
y:1296: UserWarning: 43.2% of the points cannot be placed; you m
ay want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



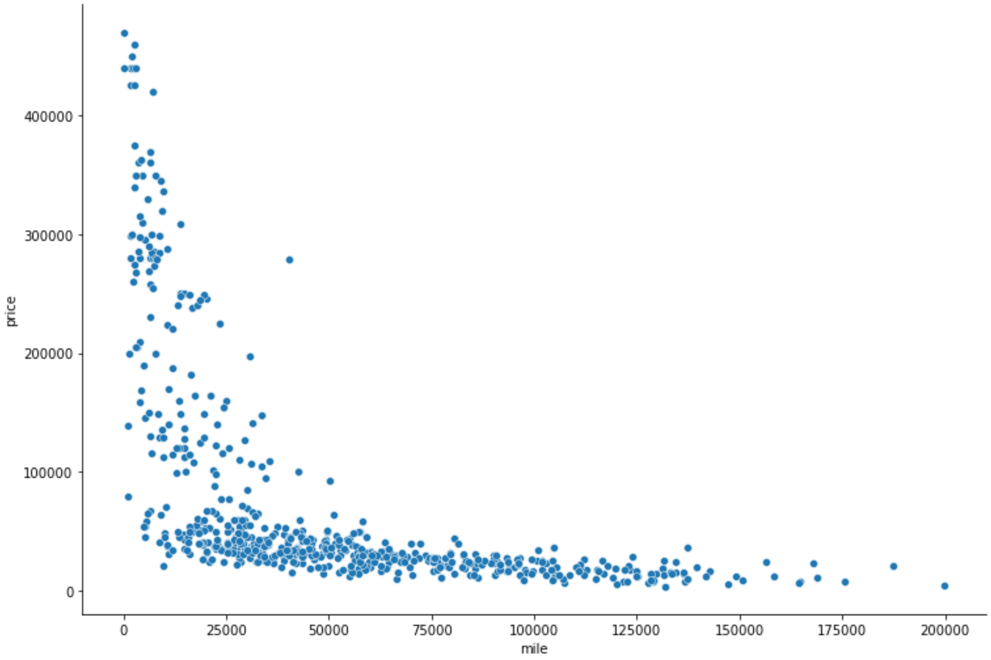
Relationship of mile with Price

In [16]:

```
sns.relplot(x ='mile',y ='price', data = dataset, height = 7, aspect = 1.5)
```

Out[16]:

```
<seaborn.axisgrid.FacetGrid at 0x2a3c9f943a0>
```



Create models for price forecasting

In [17]:

```
X = dataset[['company', 'model', 'mile', 'style', 'year']]
y = dataset['price']
```

In [18]:

```
X
```

Out[18]:

	company	model	mile	style	year
0	Audi	Q5	31188	Premium	2019
1	Audi	S4	90397	Prestige Sedan S	2013
2	Audi	A3	20259	Premium Sedan 2.0	2018
3	Audi	Q5	64995	Premium Plus	2018
4	Audi	Q7	64340	Premium Plus 3.0T	2015
...	...	...	...	...	...
610	Maserati	Ghibli	25273	Sedan RWD	2015
611	Maserati	Ghibli	39849	S Q4 AWD	2016
612	Maserati	Ghibli	75864	S Q4 AWD	2014
613	Maserati	Levante	5886	3.0L	2020
614	Maserati	Ghibli	35162	S Q4 AWD	2018

615 rows × 5 columns

In [19]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3)
```

In [20]:

```
ohe = OneHotEncoder()
ohe.fit(X[['model','company','style']])
```

Out[20]:

```
OneHotEncoder()
```

In [21]:

```
column_trans = make_column_transformer((OneHotEncoder(categories = ohe.categori
                                    remainder = 'passthrough')
lr = LinearRegression()
```

In [22]:

```
pipe = make_pipeline(column_trans, lr)
pipe.fit(X_train, y_train)
```

Out[22]:

```
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(remainder='passthrough',
                                   transformers=[('onehotencode
r',
                                                  OneHotEncoder
(categories=[array(['3 Series', '360', '4 Series', '430', '458 I
talia', '488',
       '5 Series', '599 GTB Fiorano', '6 Series', '612 Scagliett
i',
       '7 Series', '812 Superfast', 'A3', 'A4', 'A5', 'A6', 'A
8',
       'California', 'California T', 'Cherokee', 'Compass', 'Cou
pe'...
       'Unlimited Sahara', 'Unlimited Sahara (JK)',
       'Unlimited Sahara (JL)', 'Unlimited Sahara High',
       'Unlimited Sport', 'Unlimited Sport (JL)', 'Unlimited Spo
rt S',
       'sDrive28i FWD', 'sDrive28i RWD', 'sDrive30i RWD', 'sDriv
e35i RWD',
       'xDrive28i', 'xDrive28i AWD', 'xDrive30i', 'xDrive30i AW
D',
       'xDrive35i AWD', 'xDrive40i'], dtype=object)]),
                                                  ['model', 'com
pany',
                                                   'style'])])),
                ('linearregression', LinearRegression())])
```

In [23]:

```
y_pred = pipe.predict(X_test)
r2_score(y_test,y_pred)
```

Out[23]:

```
0.9776308884541367
```

In [24]:

```
pipe.predict(pd.DataFrame(columns = X_test.columns, data = np.array(['BMW','X5'
```

Out[24]:

```
array([52573.51828096])
```

Create functions to specify the model and style of the car

In [25]:

```
def model(model_input):
    d = dataset[(dataset['company'] == model_input) & (dataset['model'])]
    return (d['model'].unique())

def body_style(style_input):
    d = dataset[(dataset['company'] == style_input ) & (dataset['style'])]
    return(d['style'].unique())
```

Get car details from user

In [27]:

```
companies = dataset['company'].unique()
print('Welcom to Car price prediction')
print('Enter the details of the car you want to sell.')
print()
# company
print('Choose one of the companies:\n ', companies)
icompony = input('company : ')
print()

# model
print('Choose one of the following models:\n ', model(icompony))
imodel = input('model: ')
print()

# mile
imile = input('How many miles?')
print()

#style
print('Choose one of the following Body style:\n ', body_style(icompony))
istyle = input('body style : ')
print()

# year
iyear = input('Choose a year :')
```

```
Welcom to Car price prediction
Enter the details of the car you want to sell.

Choose one of the companies:
  ['Audi' 'BMW' 'Ferrari' 'Jeep' 'Maserati']
company : BMW

Choose one of the following models:
  ['4 Series' 'M4' 'X5' 'X3' '3 Series' '5 Series' 'X1' '7 Ser
ies' 'X6' 'X2'
 '6 Series' 'X6 M' 'M3']
model: X6 M

How many miles?22000

Choose one of the following Body style:
  ['430i xDrive Gran' 'Coupe' 'xDrive40i' '3.0si AWD' '328i Se
dan SULEV'
 '530e iPerformance Plug-In' 'sDrive28i RWD' 'xDrive28i AWD'
 'xDrive30i AWD' 'sDrive30i RWD' '330i Sedan RWD' '545i Sedan'
 'sDrive35i RWD' 'sDrive28i FWD' '335i Sedan' '330i xDrive Sed
an'
 'xDrive35i AWD' '528i xDrive Sedan' '340i xDrive Gran' '740i
L' '750Li'
 '540i RWD' 'xDrive28i' '328i Convertible' '328i Sedan RWD' '6
50i Coupe'
 '340i xDrive Sedan' '328i Sedan' '328i xDrive Sedan' '535i xD
rive Sedan'
 '325i Sedan' '320i xDrive Sedan' '330i RWD' '320i Sedan RWD'
'xDrive30i'
 '430i Gran Coupe' '430i xDrive Coupe' '535i Sedan'
 'Sports Activity Coupe' '4.8i AWD' '650i Convertible' '330i x
Drive Gran'
 '428i xDrive Gran' '535i Sedan RWD' '3.0i AWD']
body style : xDrive40i

Choose a year :2019
```

Price forecast

In [28]:

```
answer = pipe.predict(pd.DataFrame(columns = X_test.columns, data = np.array([i
print('Pridiction: %s$ ' % int(answer))
```

```
Pridiction: 49754$
```