Final Project Data Checkpoint

--Yiming Luo

Project Code

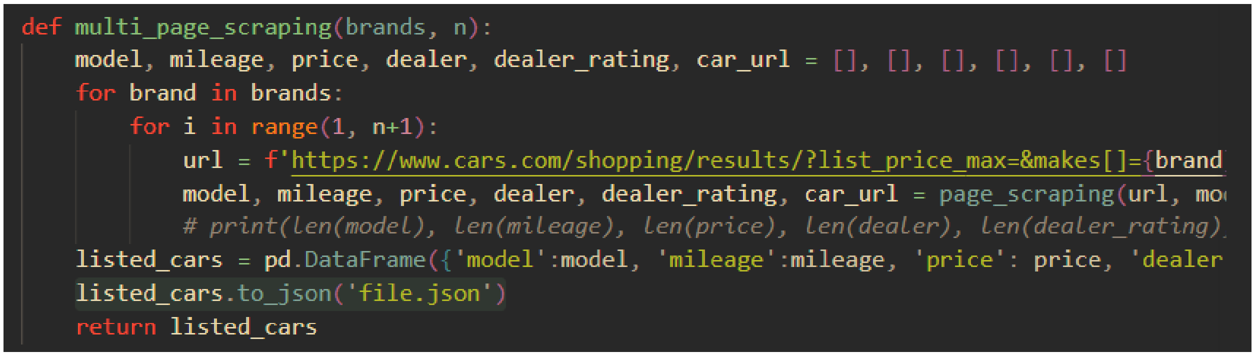
URL: https://github.com/LYM98/507-Final-Project.git

Data Sources

For this project, I only used a single source: a used car sale website. This information about the data source is presented in the table below.

Origin	https://www.cars.com/
Format	html
Method of accessing and caching	Scraping data from the website using python (requests and beautifulsoap). The scraped data will be stored in a json file for later use.
Number of available records	50000
Number of retrieved records	200
Attributes	Year: the year of the car Brand: the brand of the car Model: the specific model of the car Mileage: the current mileage Price: the selling price Dealer: the dealer's name Dealer rating: the rating of the car dealer Car URL: the URL of the page of the car

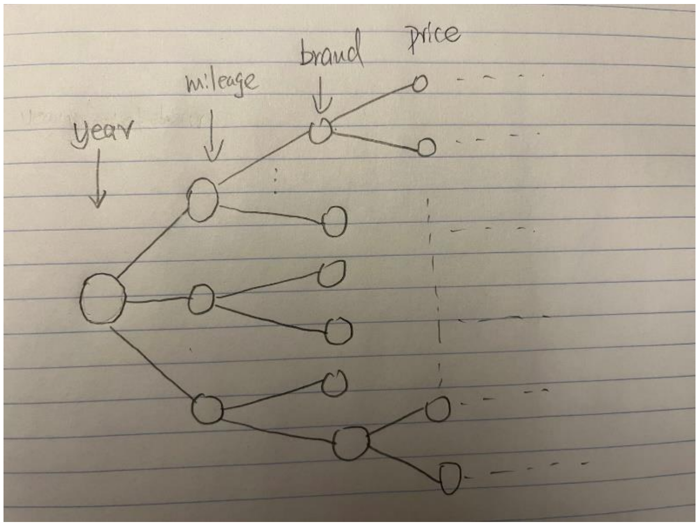
The following screenshot contains evidence of how I cached the data I scraped from the website. The line of the code is highlighted in the picture below.



```
def multi_page_scraping(brands, n):
    model, mileage, price, dealer, dealer_rating, car_url = [], [], [], [], [], []
    for brand in brands:
        for i in range(1, n+1):
            url = f'https://www.cars.com/shopping/results/?list_price_max=&makes[]={brand
            model, mileage, price, dealer, dealer_rating, car_url = page_scraping(url, mo
            # print(len(model), len(mileage), len(price), len(dealer), len(dealer_rating)
    listed_cars = pd.DataFrame({'model':model, 'mileage':mileage, 'price': price, 'dealer
    listed_cars.to_json('file.json')
    return listed_cars
```

Data Structure

For the data structure, I have a plan but haven't implemented it yet. I will be organizing my data into a giant tree. Even though I scraped seven attributes for each car, I will be using five of them, which are year, brand, mileage, price, and dealer rating, in the tree as conditions. The other two will only be shown in the results after users answer all questions. In the tree, each level stands for one attribute. Each node in the tree might have more than two children depending on the number of options each attribute has. The program will iterate through the tree using the question-answer method. After answering all questions, the result will be presented in the terminal.



Interaction and Presentation Plan

To access the data, the user will be asked to answer some questions. The number of questions depends on the depth of the tree. Those questions will be asked in the form of multiple choices. The number of answers depends on the options of each attribute. Those answers will guide the system to a leaf node which contains cars that meet all requirements.

Once the user answers all questions, a list of cars that meet the requirements will be presented in the terminal in the form of a table. All seven attributes in the car will be presented in the table. Each car will associate with an index. Users will be able to choose the index, and the corresponding page of the car will be shown on the browser.