

CAR.COM SINGLE AND MULTI-PAGE WEB SCRAPING

BY: IHECHILURU WINNER

[EMAIL](#)

INTRODUCTION

During the course of this project, I created a python script that basically extracted the

```
name
price
rating
rating_count
mileage
dealer name
```

Of all the cars present in this [link](#) and also in all the other pages that had the same in information.

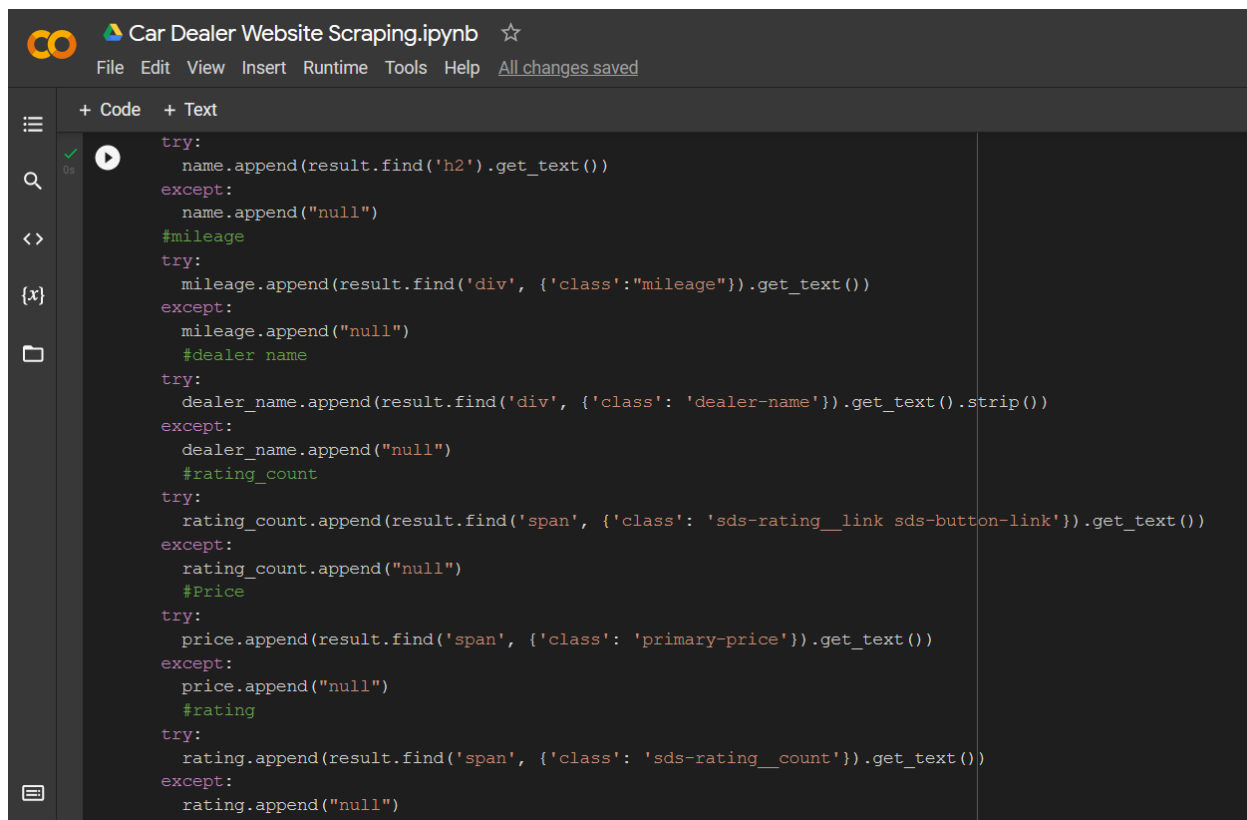
Below is a screenshot of what the website looks like.

The screenshot displays the Car.com website interface. At the top, a header reads "New and used Honda for sale" with a "Sort by Best match" dropdown. Below the header, there's a search bar showing "10,000+ matches" and a "Save search" button. A filter bar shows "Honda" selected. The "Basics" section includes filters for "Search within 20 miles", "ZIP", "New/used New & Used", and "Make Honda". A "Model" list on the left shows options like Accord, Accord Crosstour, Accord Hybrid, Accord Plug-In Hybrid, and CR-V. The main content area features two car listings. The first listing is for a "2021 Honda Pilot Elite" with 25,350 miles, priced at \$48,299, with a "Good Deal" badge and a "Home Delivery" option. The second listing is for a "2019 Honda Ridgeline Sport" with 13,184 miles, priced at \$34,790, with a "\$700 price drop" badge and a "Good Deal" badge. Both listings include a "Free CARFAX 1-Owner Report" link and a "Check availability" button.

SINGLE PAGE SCRAPING

There are a total of 20 cars per web page and for a single webpage first I developed a crawler that basically runs through just this page and gets all the information I need as regards the website.

Here's a screenshot of part of the script



```
try:
    name.append(result.find('h2').get_text())
except:
    name.append("null")
#mileage
try:
    mileage.append(result.find('div', {'class': "mileage"}).get_text())
except:
    mileage.append("null")
#dealer name
try:
    dealer_name.append(result.find('div', {'class': 'dealer-name'}).get_text().strip())
except:
    dealer_name.append("null")
#rating_count
try:
    rating_count.append(result.find('span', {'class': 'sds-rating__link sds-button-link'}).get_text())
except:
    rating_count.append("null")
#Price
try:
    price.append(result.find('span', {'class': 'primary-price'}).get_text())
except:
    price.append("null")
#rating
try:
    rating.append(result.find('span', {'class': 'sds-rating__count'}).get_text())
except:
    rating.append("null")
```

Data Cleaning

The output I got was details of 20 cars but then as you might have guessed this data at this point wasn't really ready for statistical operations, so I had to clean and prepare the data.

The picture below represents how the data was before and screengrabs of part of the script used in cleaning it. I would love to hop on a call with you and discuss with you in detail how I can prepare your data to better suit your requirements or purpose.

My mail has been attached at the beginning of this report, kindly reach me there to set up a call.

Car Dealer Website Scraping.ipynb
☆

File
Edit
View
Insert
Runtime
Tools
Help
All changes saved

+ Code
+ Text

[46]	9	2019 Honda CR-V LX	\$25,189	42,531 mi.	EchoPark Automotive Plano	(248 reviews)	4.9
	10	2019 Honda CR-V LX	\$30,975	14,883 mi.	DallasLeaseReturns.com	(4,071 reviews)	4.9
	11	2019 Honda Accord Sport	\$27,557	48,908 mi.	Bernardi Honda in Natick	(1,423 reviews)	4.6
	12	2019 Honda Pilot EX-L	\$34,000	36,156 mi.	Honda of Covington	(1,404 reviews)	4.7
	13	2013 Honda Accord EX-L	\$16,344	97,088 mi.	Prestige Auto Mart	(612 reviews)	4.5
	14	2019 Honda Ridgeline RTL-T	\$37,995	29,719 mi.	Modern Chevrolet of Burlington	(0 reviews)	null
	15	2019 Honda Accord Sport	\$28,700	23,267 mi.	Bernardi Honda in Natick	(1,423 reviews)	4.6
	16	2019 Honda Pilot Touring 8-Passenger	\$39,627	39,516 mi.	Hare Honda	(125 reviews)	3.9
	17	2002 Honda CR-V EX	\$5,807	154,838 mi.	Ford of Columbia	(214 reviews)	4.3
	18	2019 Honda Odyssey EX-L	\$37,998	25,095 mi.	Walser Nissan Burnsville	(366 reviews)	4.7
	19	2019 Honda Civic LX	\$21,500	39,022 mi.	Drive Direct	(12 reviews)	2.2

Data Cleaninig

[49] df['Rating_count'] = df['Rating_count'].apply(lambda x:x.strip('reviews').strip(' '))

▶

df['Price'] = df['Price'].apply(lambda x:x.strip('\$'))

[53] df['Mileage'] = df['Mileage'].apply(lambda x:x.strip('mi.'))

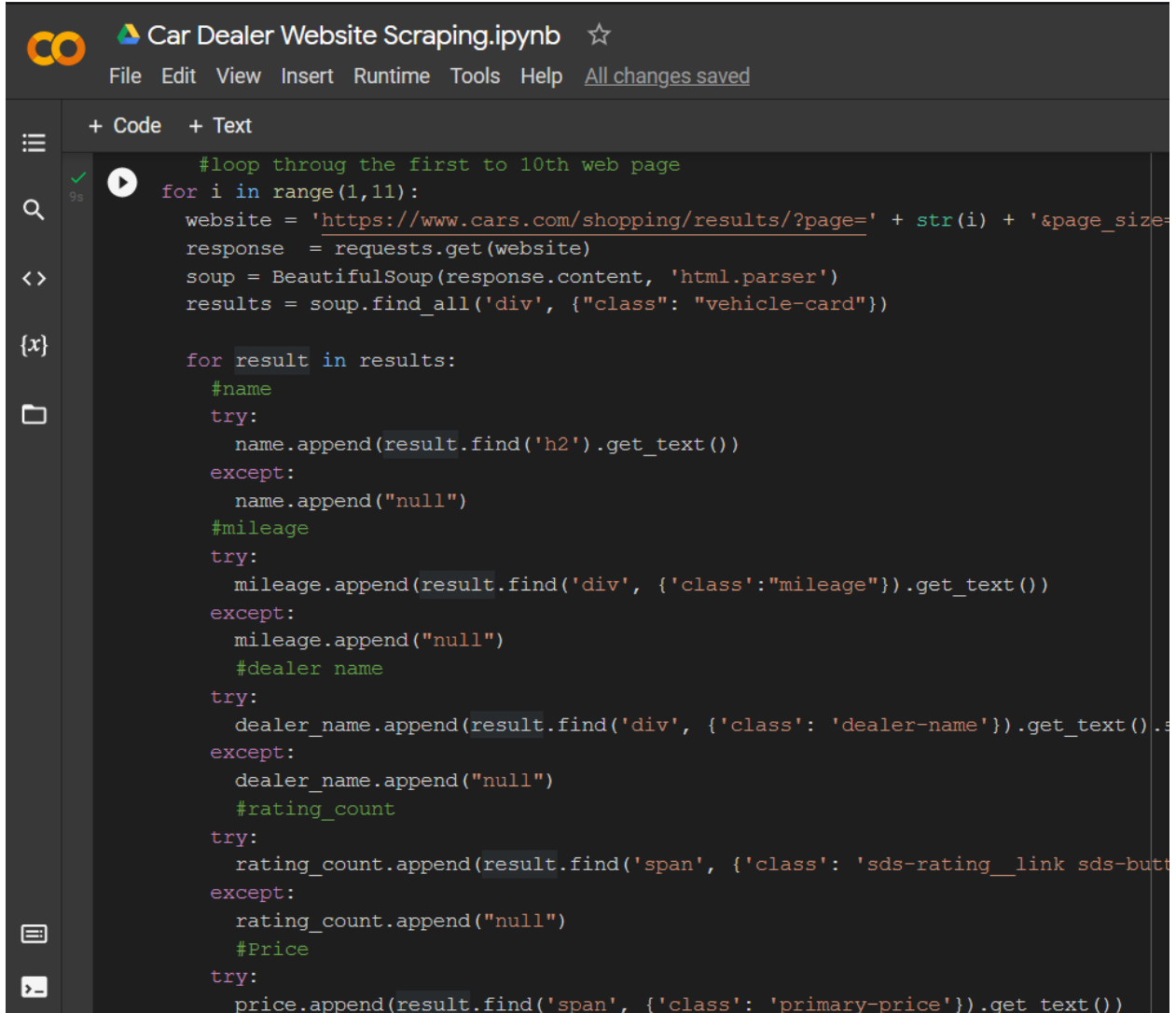
OUTPUT

Excel car.com_single_page_scraper - Saved						
File Home Insert Draw Page Layout Formulas						
<div> ↶ 📋 📌 <div> <div>Calibri</div> <div>11</div> </div> <div>B</div> <div>🔍</div> </div>						
A1	Name					
	A	B	C	D	E	F
1	Name	Price	Mileage	dealer_name	rating_count	Rating
2	2021 Honda	48,299	25,350	Vandergriff	1,341	4.7
3	2019 Honda	34,790	13,184	Ron Marhe	266	4.6
4	2019 Honda	29,000	12,553	MotorWor	1,677	4.7
5	2018 Honda	31,897	46,207	Honda of I	1,752	4.6
6	2019 Honda	38,927	36,463	Hare Honda	125	3.9
7	2019 Honda	18,799	6,639	AX Auto In	112	4.3
8	2013 Honda	18,523	84,703	Planet Hor	4,381	4.6
9	2020 Honda	37,300	14,786	Vandergriff	1,613	4.6
10	2019 Honda	27,400	46,515	Honda of I	2,477	4.8
11	2019 Honda	25,189	42,531	EchoPark	1,248	4.9
12	2019 Honda	30,975	14,883	DallasLeas	4,071	4.9
13	2019 Honda	27,557	48,908	Bernardi H	1,423	4.6
14	2019 Honda	34,000	36,156	Honda of C	1,404	4.7
15	2013 Honda	16,344	97,088	Prestige A	612	4.5
16	2019 Honda	37,995	29,719	Modern Cl	0	null
17	2019 Honda	28,700	23,267	Bernardi H	1,423	4.6
18	2019 Honda	39,627	39,516	Hare Honda	125	3.9
19	2002 Honda	5,807	154,838	Ford of Co	214	4.3
20	2019 Honda	37,998	25,095	Walser Nis	366	4.7
21	2019 Honda	21,500	39,022	Drive Dire	12	2.2

MULTI-PAGE WEB SCRAPING

In this aspect of the project, I scraped the same data but now from 10 pages in total (this number is scalable as per your requirements)

Code

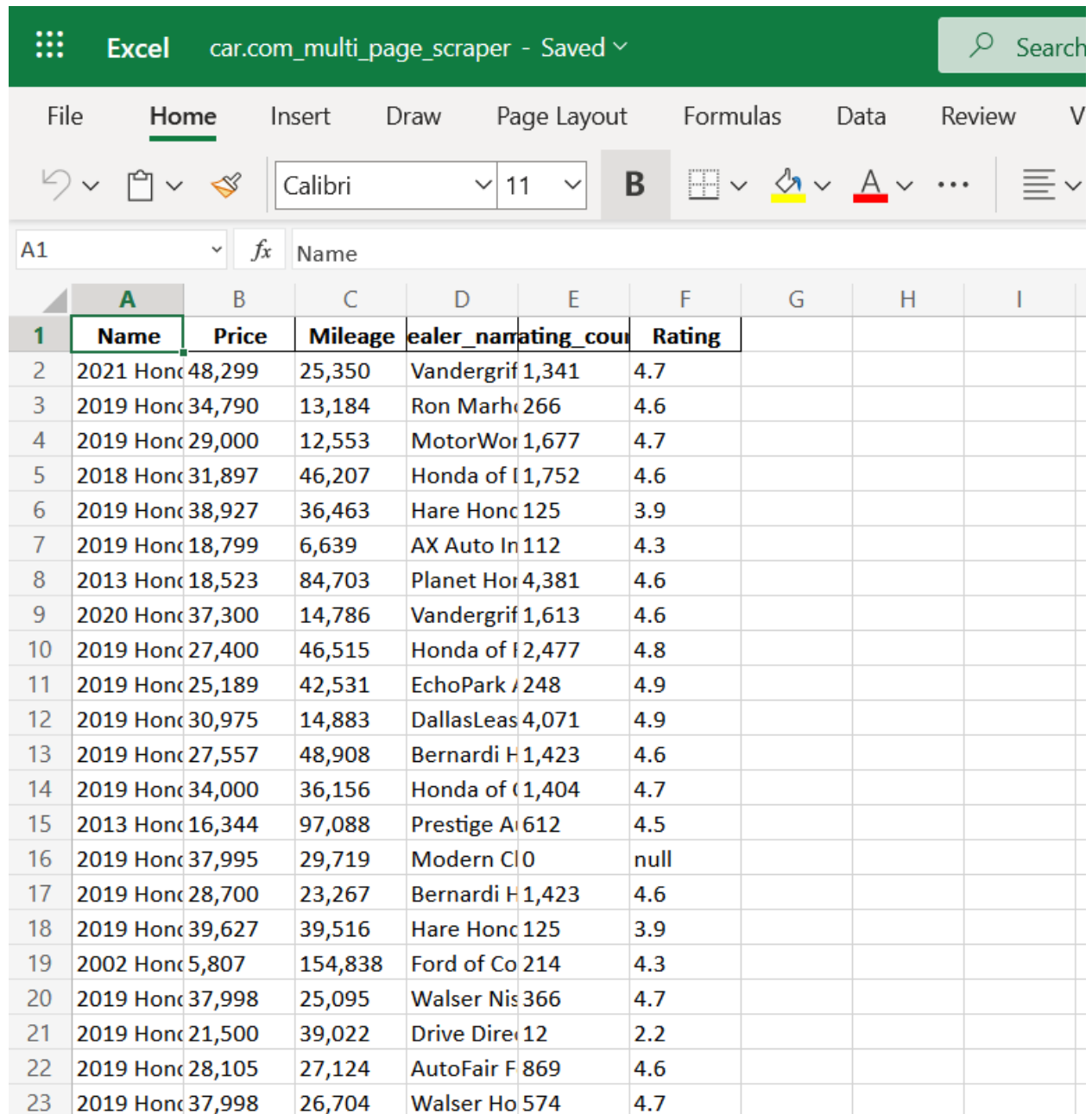


```
#loop through the first to 10th web page
for i in range(1,11):
    website = 'https://www.cars.com/shopping/results/?page=' + str(i) + '&page_size='
    response = requests.get(website)
    soup = BeautifulSoup(response.content, 'html.parser')
    results = soup.find_all('div', {"class": "vehicle-card"})

    for result in results:
        #name
        try:
            name.append(result.find('h2').get_text())
        except:
            name.append("null")
        #mileage
        try:
            mileage.append(result.find('div', {'class': "mileage"}).get_text())
        except:
            mileage.append("null")
        #dealer name
        try:
            dealer_name.append(result.find('div', {'class': 'dealer-name'}).get_text())
        except:
            dealer_name.append("null")
        #rating_count
        try:
            rating_count.append(result.find('span', {'class': 'sds-rating__link sds-but'}).get_text())
        except:
            rating_count.append("null")
        #Price
        try:
            price.append(result.find('span', {'class': 'primary-price'}).get_text())
```

As you might already think, we also cleaned the data we extracted and then the final output has a total of 200 entries.

Here's what the final output looks like:



The screenshot shows an Excel spreadsheet titled "car.com_multi_page_scraper - Saved". The ribbon is set to "Home". The formula bar shows "Name". The spreadsheet contains a table with 23 rows of car data. The columns are labeled A through I. The data is as follows:

	A	B	C	D	E	F	G	H	I
1	Name	Price	Mileage	Dealer Name	Rating	Count			
2	2021 Honda	48,299	25,350	Vandergrif	1,341	4.7			
3	2019 Honda	34,790	13,184	Ron Marhe	266	4.6			
4	2019 Honda	29,000	12,553	MotorWor	1,677	4.7			
5	2018 Honda	31,897	46,207	Honda of I	1,752	4.6			
6	2019 Honda	38,927	36,463	Hare Honc	125	3.9			
7	2019 Honda	18,799	6,639	AX Auto In	112	4.3			
8	2013 Honda	18,523	84,703	Planet Hor	4,381	4.6			
9	2020 Honda	37,300	14,786	Vandergrif	1,613	4.6			
10	2019 Honda	27,400	46,515	Honda of I	2,477	4.8			
11	2019 Honda	25,189	42,531	EchoPark	1,248	4.9			
12	2019 Honda	30,975	14,883	DallasLeas	4,071	4.9			
13	2019 Honda	27,557	48,908	Bernardi H	1,423	4.6			
14	2019 Honda	34,000	36,156	Honda of C	1,404	4.7			
15	2013 Honda	16,344	97,088	Prestige A	612	4.5			
16	2019 Honda	37,995	29,719	Modern Cl	0	null			
17	2019 Honda	28,700	23,267	Bernardi H	1,423	4.6			
18	2019 Honda	39,627	39,516	Hare Honc	125	3.9			
19	2002 Honda	5,807	154,838	Ford of Co	214	4.3			
20	2019 Honda	37,998	25,095	Walser Nis	366	4.7			
21	2019 Honda	21,500	39,022	Drive Dire	12	2.2			
22	2019 Honda	28,105	27,124	AutoFair F	869	4.6			
23	2019 Honda	37,998	26,704	Walser Ho	574	4.7			

Here's my [Email](#), contact me so we can set up a call and discuss how I can help you extract data from your web source or server and export it to whatever format you'd like.

Ihechiluru Winner.