STAT40800 Paul McEvoy #15202922 Python Project

Title: Online Car Sales Scraper

Introduction

For this project I have selected Strand #2: Data Wrangling.

I have a passing interest in car buying and the prices associated with various models second hand. I often noted how different brands and models often carry price premium. The changes in motor taxation in Ireland have also had an effect on cars of certain engine capacity and fuel type. Also, traditional brand favourites in Ireland anecdotally have moved towards Asian brands in recent years as price has become a key factor.

My aim was to retrieve as much data as I could on sales of second hand cars online and analyze whether some theories about how aspects like, age fuel type and mileage etc affect price but also how these factors affect each other. For instance; do diesel cars on average have more mileage, do certain brands still carry a price premium, do people from Cork really buy/sell more red cars than other counties. The project is split into two parts; 1. *Data Scraping* and 2. *Analysis*.

Data Scraping

Car price data in Ireland (particularly in the second-market) is not readily available to process and some effort is needed to extract the information from online resources. To get enough data I needed to select an online sales hub and scrape the HTML using python. Having browsed a few I settled on http://www.carsireland.ie/ as the best option to get what I needed. CarsIreland offered two main advantages:

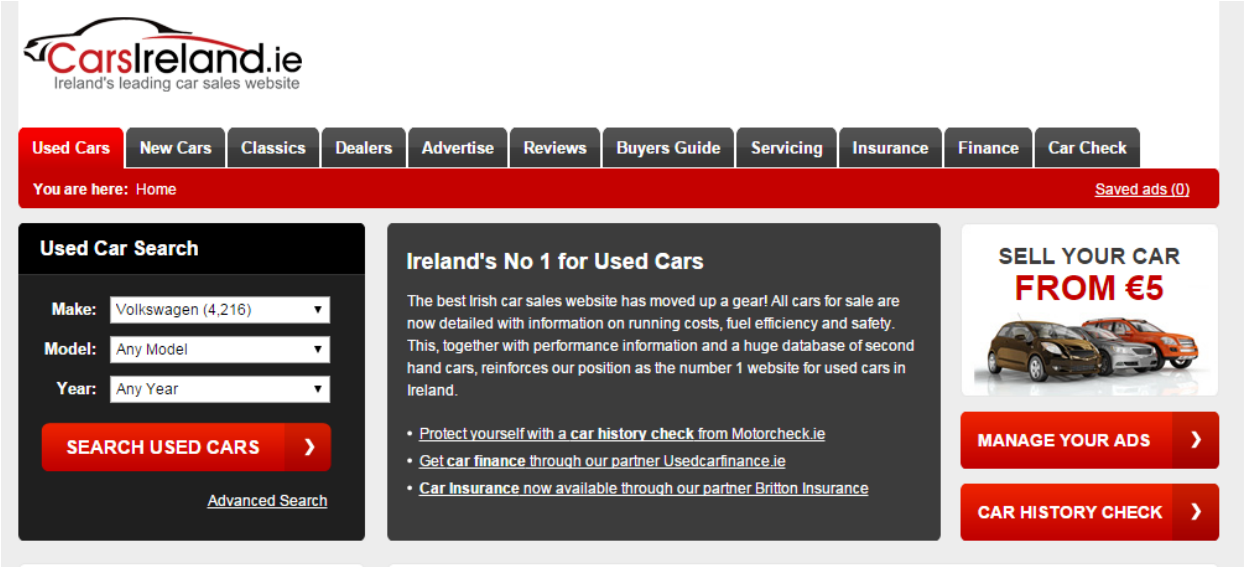


Figure 1 CarsIreland FrontPage

Firstly it allowed me to modify the URL to select whatever query I needed. Makes, age etc can all be placed in the URL to load the required contents. Some websites require search inputs in dropdown boxes etc and the results are displayed in JavaScript without changing the URL. This can be more complex to control from a python script.

Secondly, *CarsIreland* displays all results with a plenty of information related to the car without having the click through to the advert. This is a huge benefit. Other sites merely gave a snippet that forces the user to visit the advert itself to get further info. This saves creating a script that scrapes each advert separately but instead parses a full page of results.

To collect data I was specifically interested in the most popular car brands in Ireland and a window of 15 years - 1999 to 2014. This would give me the best representative sample of second-hand stock. I decided to discard 2015 sales as many of these can be dealer show-car sales, often with little mileage or use and skew the results. I also discarded pre-1999 cars on the basis that the price of a car can often follow a bath-tub curve where older cars may actually appreciate in value as certain desired models become rare. Even older again can be in the "classic" range and would skew results considerably.

I decided to work with the 5 most common brands in Ireland currently; *Volkswagen, Toyota, Nissan, Peugeot and Hyundai*. This selection represents about one third of all second-hand cars currently for sale in Ireland less than 15 years old.

The data I needed was not stored in a database available to me so I needed an interface would load a webpage of a desired URL and manually fetch the HTML for that search. Using Google to find the best candidate directed me to http://phantomjs.org/. From their site:

" PhantomJS is a headless WebKit scriptable with a JavaScript API."

Essentially it provides the user to a mechanism to background load a webpage, perform some action and then close. Such actions can be as simple as grabbing the HTML or even manually entering search terms or clicking buttons. My requirements were simple in that I just needed it to repeatedly load a series of URLs and return the HTML.

To execute the WebKit I just download an exe file and called it with:

```
driver = webdriver.PhantomJS() # This is the driver to fetch the webpage

driver.get(url)

 sleep(1)

html_source = driver.page_source
```

When testing PhantomJS initially I discovered the returned HTML was missing information compared to when I viewed it manually in a browser. It became clear that the JavaScript generated within the webpage often took some time to render and a 'sleep' command was needed to wait between page renders.

The aim now was to loop through each URL and parse each page of HTML. I needed to issue a "search" on a car make within an age range grab the information on each advert.

The first challenge was recognizing that each search resulted in number of pages of results updating the URL to increment the page number up to some maximum. The python package BeautifulSoup offered the parsing tools to allow me to extract the page number:

```
num_soup = BeautifulSoup(html_source,'html.parser')

# the page value is stored in a tag called "strong_tag"

for strong_tag in num_soup.findAll('strong'):

    pages = re.findall(r'\d+', strong_tag.text)
```

The 'pages' variable contained the total number of pages for that search. I could then use that update the URL and increment through each page of results.

Figure 2 below shows a sample search with a modified URL to include an age range that displays a HTML tag containing a page number total for that search. I then used this value to constrain a loop to cycle through every page and scrape data for that car brand. Note the location, colour, engine size information etc neatly presented in the results that I would scrape in the next part of the script.

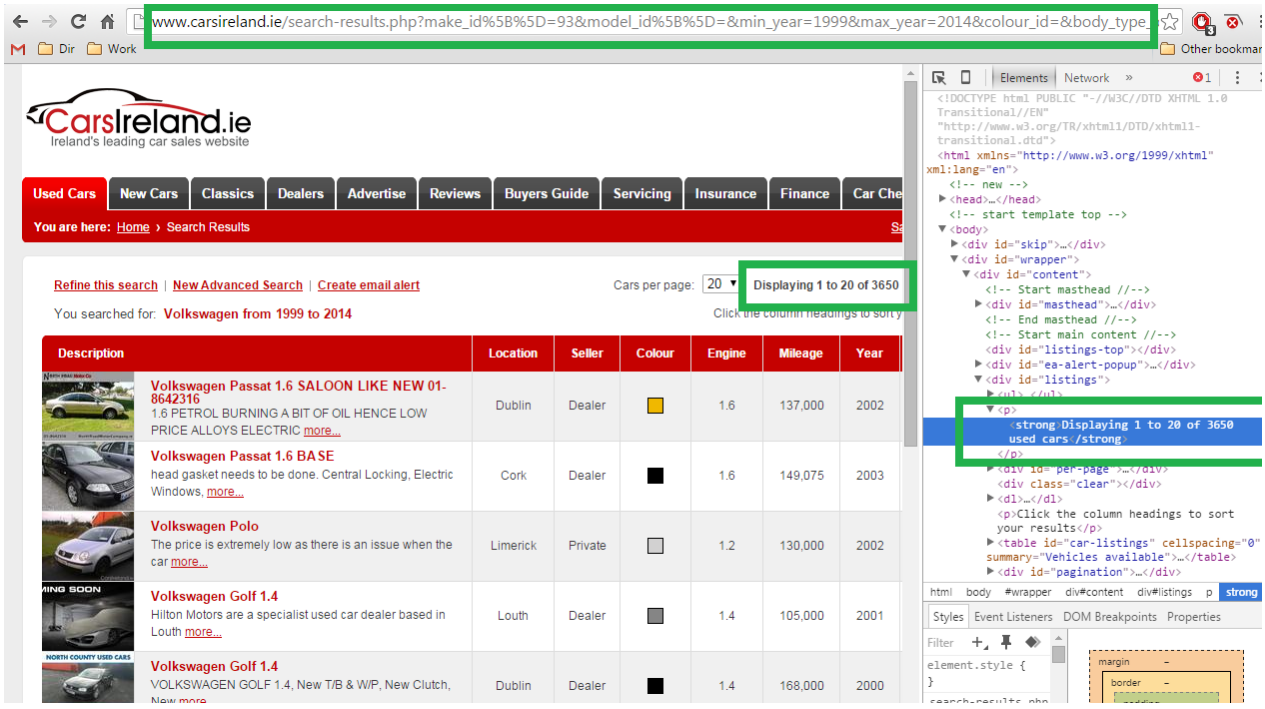


Figure 2 Sample search showing page fields and URL

One complication for each search was that car makes were applied to the URL in the form of an ID value rather than the text string of its name. I manually issued searches for each make until I had a full list. I was then able to create the format of the URL string to issue to *PhantomJS*:

```
url_clean = 'http://www.carsireland.ie/search-results.php?make_id=' + make_id +
'&model_id=&per_page=' + result_num_str + '&page=' + page_str +
'&min_year=1999&max_year=2014&search-used-submit=Search+Used+Cars'
```

By modifying make_id and page_str I would be able to issue searches for all available cars in the desired range. (result_num_str was set to 50 but I made it a variable in case I wanted to vary it later).

I created a function that took a page value, a make_id and a make name (used when populating the car info matrix). This function, scraped the HTML, parsed it with *BeautifulSoup*, extracted the desired tags, grouped the results into lists and dataframes and finally performed some cleaning to remove unwanted characters and other un-needed text.

The function then combined all the data and returned the results for that page, for that make to a dataframe:

```
car_info_matrix_df = pd.concat([makes_df, car_info_matrix_df, fuel_and_engine_df,
colour_list_df, car_info_matrix_price_df], axis=1)
```

Below is an example of the format of the data returned:

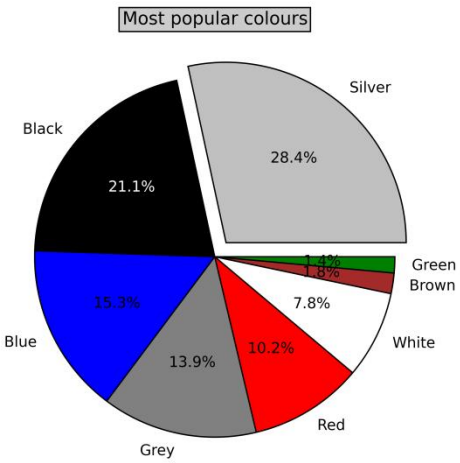
Index	make	county	mileage	year	fuel	enginesize	colour	price
5402	volkswagen	Kilkenny	51855	2013	Diesel	1.6	Black	17900
3963	volkswagen	Galway	136701	2007	Diesel	2.0	Blue	8500
3973	volkswagen	Meath	98000	2008	Petrol	1.4	Silver	8745
3972	volkswagen	Donegal	64556	2011	Diesel	1.6	White	8650
3971	volkswagen	Cork	64935	2008	Petrol	1.4	Blue	8595
3970	volkswagen	Donegal	10064	2012	Petrol	1.0	Red	8500
3969	volkswagen	Kildare	129000	2007	Diesel	1.5	Silver	8500
3968	volkswagen	Kerry	151407	2007	Diesel	2.0	Black	8500
3967	volkswagen	Limerick	93808	2008	Diesel	1.4	Grey	8500
3966	volkswagen	Galway	142103	2008	Diesel	1.9	Silver	8500
3965	volkswagen	Leitrim	180000	2008	Diesel	1.9	Blue	8500
3964	volkswagen	Tipperary	132000	2009	Diesel	2.0	Grey	8500
3962	volkswagen	Galway	102275	2009	Diesel	1.9	Silver	8500
3975	volkswagen	Kildare	83885	2012	Diesel	1.6	White	8750
3961	volkswagen	Dublin	106257	2008	Diesel	2.0	Purple	8500
3960	volkswagen	Galway	80000	2008	Petrol	1.4	Red	8500
3959	volkswagen	Westmeath	116000	2008	Diesel	1.9	Silver	8500
3958	volkswagen	Tipperary	146647	2008	Diesel	1.9	Red	8500
3957	volkswagen	Leitrim	127000	2008	Diesel	1.9	Black	8500
3956	volkswagen	Clare	102000	2008	Diesel	1.9	Black	8500
3955	volkswagen	Galway	166212	2008	Diesel	1.9	Grey	8500

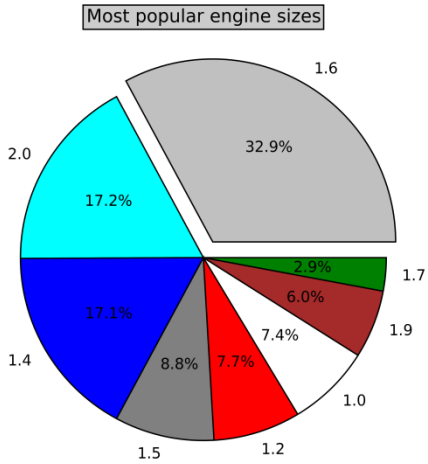
Figure 3 Example data for page scrape

This only represented the top of one page. The function was called repeatedly until it reached the max page for that make before continuing onto the next make in the list. Once all makes were retrieved the data was cleaned once more to fix some indexing, correct the column names and remove some out-lying results that would skew the final analysis. Once this was complete the dataframe was dumped to a CSV file. The step was important as scraping for all models required about 200 web searches and took about 5 minutes. Additionally, the repeated nature of the webpage requests ran the risk of being interpreted by CarsIreland as a commercial data harvest or possibly a DoS action that might cause the server to block my IP. From this point on I could analyze the data from the CSV and re-scrape only if I needed fresh advert information.

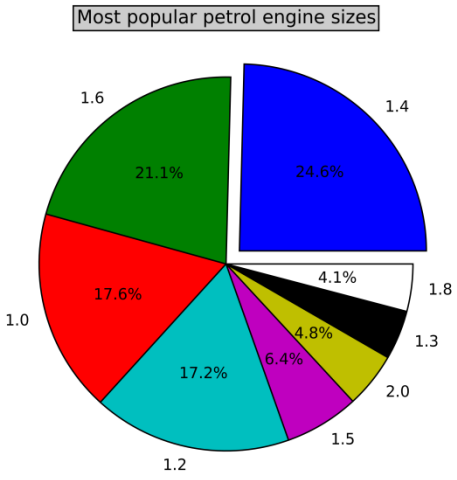
Analysis

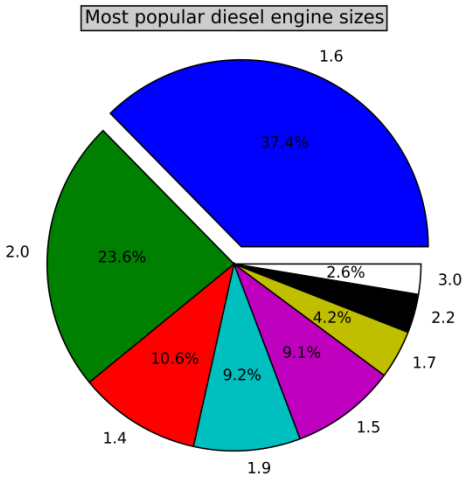
I was interested in answering a few questions around popularity. I already knew what the most common brands were but not the most common engine sizes and colours etc. Also, how did these vary across counties and what were the most important factors in determining price.





The charts above indicate there is a distinct preference for silver and black cars in Ireland. Brown is a surprising entry in the top eight. Engine size clearly favours 1.6 litre. It was interesting to see 2 litre engines so high, I would have expected smaller engines to be more popular. My guess here is that since diesel engines are typically higher capacity this might account for the result. To confirm I split out comparisons between engine sizes between fuel types.





Contrary to what I expected, diesel engines were even more popular in the 1.6 litre bracket! One final theory to explain this I felt might have been related to the age of the car. More modern cars now come with smaller diesel engines than were required 10+ years ago. Improvements in power delivery and efforts to reduce emissions have forced manufacturers to reduce diesel engine sizes. To determine if this is true, I compared diesel engine size against the age of the car.

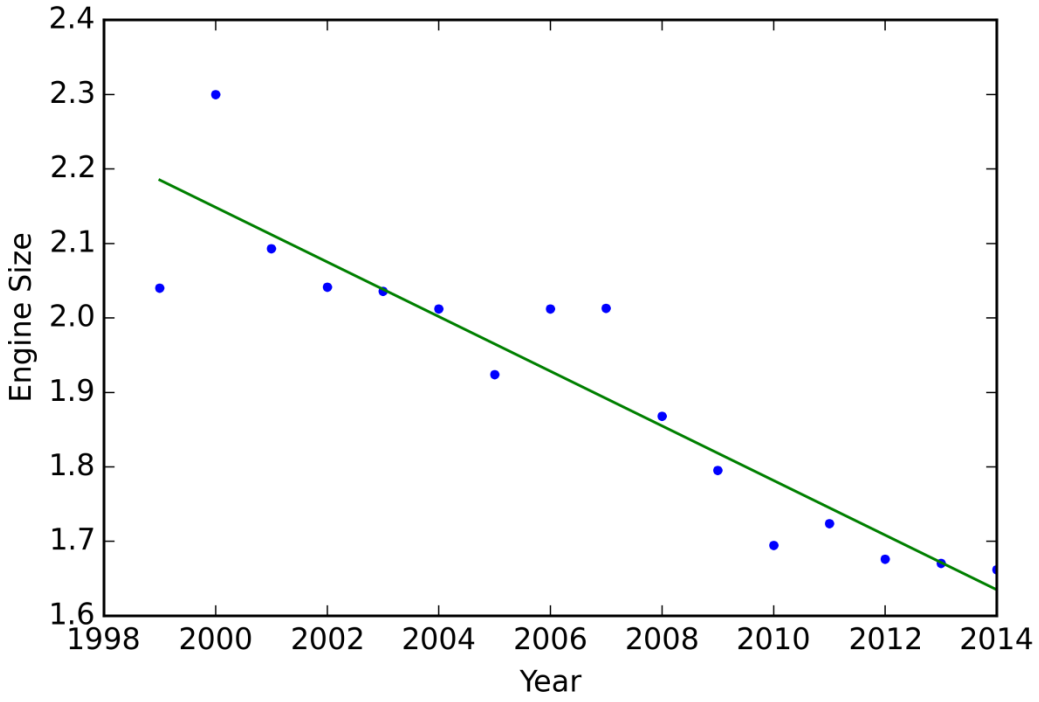


Figure 4 Diesel Engine size vs Car age

As I expected, the average size of the diesel engine has decreased over the last 15 years.

Lastly I took a look at some other interesting info. Counties with the highest mileage - it seems midlands drivers like to drive long distance. Looking for a car with low miles? Search in Leinster.

county	mileage	county	mileage
Longford	108843	Westmeath	70122
Leitrim	100766	Dublin	70045
Tipperary	87859	Kilkenny	69284
Clare	85822	Carlow	67097
Cork	85738	Kildare	64050

Figure 5 Highest and lowest counties by mileage

Colour preferences. Do people in Cork sell more red cars?

Cork car colours	percentage	Dublin car colours	percentage
Silver	31	Silver	25
Black	16	Black	20
Blue	15	Blue	14
Grey	12	Grey	13
Red	10	Red	9
Galway car colours	**percentage**	**Tippearary car colours**	**percentage**
Silver	26	Silver	30
Black	19	Black	16
Blue	14	Blue	15
Grey	13	Grey	14
Red	10	Red	9

No more than any other county it seems.

Finally, I was interested to see what factor had the strongest correlation(Pearson) in determining price.

Age vs Price correlation: **0.826**

Mileage vs Price correlation: -0.571

Engine Size vs Price correlation: 0.19

Clearly, the age of the car is the most important factor in determining price. The negative correlation on mileage makes sense - the higher the mileage, the lower the price. It would seem engine size also has some correlation on price but other factors dominate.

Conclusion

It is evident that there is value in taking a snapshot of car sales data in the second-hand market, at least to confirm conventional beliefs on how age and mileage affect an asking price. However, there is also support from the data that car engine sizes are definitely decreasing. Is this due to the **tax rates** on **engine sizes** or emissions or both?

Colour seems to have little variability across counties but an obvious weakness in the data is that I am looking at is second-hand car sales. Cars move around between owners across counties and a more accurate colour preference could be obtained from new car sales per county.

Lastly, it would seem that **age** is still the dominant factor in determining price. Although this is expected, it would be interesting to compare cars within each year across a range of mileages. i.e., how does mileage affect all cars from 2005? Do newer cars with higher mileage still demand a higher price than older ones with fewer miles? It would also be interesting compare with other countries that don't include registration year on their plates. Does mileage become more important? These are questions that could be answered in a more advanced analysis.

Appendices

CarsIreland: http://www.carsireland.ie/

PhantomJS Webkit: http://phantomjs.org/

Full code can be found within the submission or at Dropbox: cars_ireland_scraper.py

Note: To run the Python code, a package called *selenium* needs to be installed. Run: *pip install selenium* in a command line. Additionally, the PhantomJS executable is required and can be downloaded from the website listed above.