Otomoto.pl: WebScraping project

May 7, 2022

Jakub Gotlib 382358, Jakub Kołoczek 439898, Jakub Niedziela 412466

Motivation and introduction

When in the end of XIX century, Étienne Lenoir, Nicolaus Otto, Karl Benz, Gottlieb Daimler, Wilhelm Maybach, and Rudolph Diesel were building first engines little did they know how groundbreaking their discoveries truly were. And yet here we are. After more than a hundred years still using cars with their mechanical hearts constructed along with general patterns proposed by the mentioned engineers. The importance of cars seems to be very persistent, from a point of view of individuals as well as companies. The latter ones are mainly interested in purchasing cars in order to use them as tools needed to increase their profits. For some of them, cars play a crucial role in generating profit, e.g. shipping companies. As far as ordinary people are considered, we differ in ways we treat our vehicles. For some of us, cars are like family members while others see in cars nothing more than a relatively fast way of traveling from point A to point B. No matter the reasons, a decision of whether to buy a car is still one of the most important. And, thus, it has to be carefully made. This is the main reason we decided to scrape *otomoto.pl* website. We believe that having access to an impressive amount of information may help in making a rational decision and, as a result, turns out to be a profitable transaction for ordinary people as well as businesses.

The website we chose, otomoto.pl, contains a few hundred cars' offers from all over Poland which make it one of the most important places to look for a new vehicle on the Polish internet. Users have access to a decent number of filters that are developed in order to display them as interesting offers as possible. What is more, otomoto.pl is also a place to find mechanical parts needed for repairs in already used cars.

Technical details of the scraper

Our three scrapers follow a similar founding concept. They firstly travel through the next pages of *otomoto.pl* collecting links to offers. Then they use those links to access a site containing details of a given offer. From there, parameters or quantities (such as number of driven kilometers, engine's capacity, and so forth) can be downloaded. After scraping one link (a single offer) a program moves to another and again all information is extracted.

While the main idea is similar for all scrapers, the programming implementations are fairly different. For instance, in a code using Scrapy module, two separate spiders are responsible for mentioned tasks. The first one collects links while the second one (using those links) focuses on extracting offers' parameters. Additionally Scrapy crawler does not need links to all the offers, it simply collects all links from each numbered page that contain the phrase "oferta", by using Rule and LinkExtractor classes. Then it automatically passes these links to parse function, which collects data about the car.

Quite a different implementation is applied in a program based on the Selenium package. After filling the filtering criteria, scraper visits pages with a few dozens of offers each and collects the links. The Selenium scraper can navigate through the website thanks to clicking buttons, e.g. next page button to change page. Therefore, one loop prepares a list of links and the second one makes use of them to download offers' details.

A scraper prepared using the Beautiful Soup package goes through three main tasks. Firstly, links to all the *otomoto.pl* pages containing a list of cars' offers are generated. From those pages direct links to offers are extracted (second step). And from there, parameters of a given car can be collected.

Example analysis

Since our scrapers are able to obtain several important details of offers available on the *otomoto.pl* website, the outcome of their execution can become a subject of fairly interesting analysis. To begin with, one might be eager to discover some basic descriptive statistics for each car make, for example. Such information is valuable because it makes it possible to compare the attractiveness of a given offer. A potential customer can evaluate whether a deal seems to be better than alternatives. In order to do so, one may want to look at a graph similar to that shown in Figure 1. The box plot presents a summary of prices for a few popular car's makes.

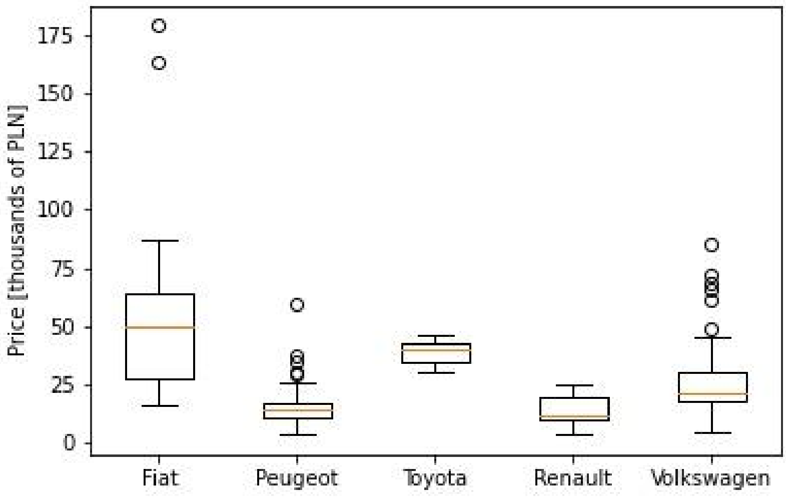

Figure 1. The box plot of cars' prices for a few popular car manufacturers

However, computing simple descriptive statistics is not everything one can do with a dataset consisting of a dozen of variables (such as price, age of a car, number of driven kilometers as so on). Such cross-sectional data may be used to prepare an econometric model, for instance. Perhaps someone would like to explain car prices with a simple linear regression model.

Potential analysis does not have to be limited to methods of classical econometrics. The dataset collected by our scrapers can be used as an input of sophisticated machine learning tools. For instance, it seems to be a perfect dataset for an artificial neural network that could predict the price of a car using its characteristics.

Work division

Our group consists of exactly three people so the division of work was fairly straightforward. One person was responsible for preparing one scraper. And so Jakub Niedziela created a code with Scrapy, Jakub Kołoczek prepared a program that used the Beautiful Soup package and Jakub Gotlib wrote a Selenium scraper.