Bartłomiej HADASIK

Reduction of information asymmetry in e-commerce: the web scraping approach

Publishing House of the University of Economics
in Katowice

BARTŁOMIEJ HADASIK

REDUCTION OF INFORMATION ASYMMETRY IN E-COMMERCE: THE WEB SCRAPING APPROACH

Katowice 2024

Scientific publication

Projekt dofinansowany ze środków budżetu państwa, przyznanych przez Ministra Edukacji i Nauki w ramach Programu „Doskonała Nauka II" (nr umowy MONOG/SP/0101/2023/01)

The project was co-financed from the state budget funds granted by the Minister of Education and Science (Poland) within the framework of the „Doskonała Nauka II" programme (contract no. MONOG/SP/0101/2023/01)

Ministry of Science and Higher Education Republic of Poland

Doskonała Nauka

Reviewer
Janusz Wielki

Publishing editor
Beata Kwiecień

Typesetting
Daria Liszowska

Cover design
Emilia Gumulak

Cover picture © duiwoy – Photogenica

Printed by
EXDRUK Spółka Cywilna Wojciech Żuchowski, Adam Filipiak
ul. Rysia 6, 87-800 Włocławek
e-mail: biuroexdruk@gmail.com, www.exdruk.com

ISBN 978-83-7875-889-1
e-ISBN 978-83-7875-890-7

http://doi.org/10.22367/9788378758907

Publishing House of the University of Economics in Katowice
ul. 1 Maja 50, 40-287 Katowice, tel.: +48 32 257-76-33
www.wydawnictwo.ue.katowice.pl, e-mail: wydawnictwo@ue.katowice.pl
Facebook: @wydawnictwouekatowice

Contents

I would like to sincerely thank
Mrs. Maria Mach-Król, PhD, Associate Professor
(University of Economics in Katowice)
for instilling in me the exhilaration of scientific development, being my guru,
as well as her deepest support, professionalism, and faith,
Mrs. Julia Włodarczyk, PhD, Associate Professor
(University of Economics in Katowice)
for acquainting me with the secrets of economics, and the remarkable kindness,
as well as my Parents and Grandparents, especially my $\boxed{\textit{beloved Grandma}}$
for their love, comfort, hopefulness, encouragement, and everything else.

Introductory word

The realm of science and its progress have always been built on research, and ultimately on the experience that precedes it. In order to arrive at scientific results appropriately, investigations must be conducted and those are done when data is accessible. In the modern era of digital world and society, data is gathered much simpler than before the widespread availability of computers and broadband networks. Sadly, it is a challenge for a beginner researcher to access databases gathered by various organizations since they are safeguarded and available only to a small audience (sometimes for an additional price). As data collecting becomes much simpler when we have access to IT solutions of the 21st century, it is even more convenient with a utilization of an application that automatically gathers and organizes data. Such an automized database building technique may become notably beneficial when we have a desire to collect unstructured data from a given period and from a specific website, independently from the industry. This is where web scraping – a strategy that includes obtaining data from websites, is handy. In actuality, data extraction (especially approaches linked to the very web scraping) comprises of a large variety of distinct methods and technologies, such as data analysis, natural language syntax analysis, and information security. To get the most out of their advantages, it is of paramount importance to understand how they function.

The role of information in the purchasing process has been extensively described in the literature. In doing so, attention was often drawn to the problem of information asymmetry – when the individual customer is informationally in a weaker position than the seller. This problem becomes particularly important in online shopping. The purpose of this work is to create an automated tool based on the web scraping technique that is designed to reduce the information asymmetry occurring in the buyer-seller relationship. The plane for depicting the phenomenon of information asymmetry and the established web scraping tool is the automotive sector, with a particular focus on the essence of classifieds portal as a platform for matching buyers with sellers. The case of the largest automotive classifieds portal in Poland, which is OTOMOTO, was used in this study. The theoretical backdrop of this research, which serves as its beginning point, will be the problem of the uncertainty of judgments, coming from information asymmetry, an example of which is described in the groundbreaking essay by Akerlof (1970). In this work, the baseline environment for illustrating the problem of information asymmetry is also the automotive industry. In order to achieve the goal of this study, the following research questions were posed:

RQ1. What are the implications of information asymmetry for judgment uncertainty in online transactions, and how can they be mitigated?

RQ2. How can web scraping tools be designed to specifically address the challenges of information asymmetry in the e-commerce sector?

RQ3. What is the potential impact of reducing information asymmetry through web scraping on the overall efficiency and fairness of the e-commerce market, especially in automotive industry?

This book is organized as follows. Chapter 1 outlines the theoretical background with specific attention dedicated to the issue of information asymmetry as articulated in Akerlof (1970). Chapter 2 discusses the theoretical foundation of data extraction from internet resources (with particular focus on web scraping), their characteristics, particularly legal as well as ethical issues, and the necessity to deploy data collection technologies in the research setting. In Chapter 3, a tool for data extraction created together with a suitable database to be able to harvest data from the OTOMOTO advertising site is discussed. The Chapter also provides technical elements including the Python language upon which the constructed tool is predicated. Chapter 3 additionally covers a practical portion of the research in which a sample evaluation of the automotive industry in Poland is done, which draws on the data gathered from OTOMOTO advertisement portal with the assistance of the built web scraping tool.

The book can be found useful for researchers, academics, and data scientists, offering scholarly insights into reducing information asymmetry in e-commerce through web scraping. E-commerce practitioners and business owners in the automotive sector can gain competitive advantages by applying the book's practical guidance for market analysis. The employment of the created web scraping tool, once quantitative data is retrieved, can be used by, e.g., data analysts, for the advanced analysis of the particular market, the verification of research hypotheses and the facilitation of decision-making processes. Policy makers, regulators, and legal professionals will find valuable perspectives on the legal implications of web scraping in enhancing information transparency. On the other hand, everyday customers of online stores may benefit from the theoretical and practical value that this book brings, especially with their willingness to compare offers posted in advertisements, further analyze them, and make the right purchase decision for themselves based on more complete access to information (or put another way: minimized uncertainty among buyers).

This monograph is an adaptation of the author’s master’s thesis with the same title, which was defended in July 2021 at the University of Economics in Katowice (Poland) under the supervision of Associate Professor Maria Mach-‑Król, PhD. The thesis was defended with a very good result and served as the basis for issuing a Master’s degree diploma with distinction to the author. The thesis was awarded the second degree prize in the 2022 nationwide competition of diploma theses in the field of economic informatics, which was awarded by the Scientific Society of Economic Informatics (Częstochowa, Poland).

Chapter 1. E-commerce in the face of information asymmetry

Information might be regarded as the fundamental backbone, or among them, of the continuously expanding civilization of the internet-based economy. Contemporary technologies make access to information more accessible and prevalent yet boosting societal disparities also converts into the still happening challenge of choice uncertainty, arising from the demonstrated uneven access to information. This Chapter addresses the phenomena of information asymmetry in both the theoretical and the practical setting of decision making in configurations of uncertainty, and illustrates the difficulties that the e-commerce business has to confront in this area. Furthermore, the information portal is portrayed as an e-commerce tool that might possibly help to future development in lowering the incidence of uneven access to information.

1.1. Uncertainty and information asymmetry in the digital economy

The growing globalization coupled with the socio-economic and technological advancements contributed to the rapid rise in the information technology and systems development industry (Schmidt et al., 2009). Novel developments in communication with the end customer were introduced, which stemmed from technological advances and the dissolution of global borders across nations and therefore specialized markets. Such new technical forms have enlarged the scope of contacting new customers throughout the world through altered information flows, which serves as a strong evidence of the modification of what the customer desires. Thus, the shift from the economy of the industrial revolution to the consumer centered economy characteristic of the information age is obvious (Senge et al., 2001; Carlsson, 2004).

The alterations in market demands and a constantly evolving society, one of whose key desires is information, imply that more factors ought to be studied than previously in order to arrive at proper judgments. Decision making is not exclusively the realm of our contemporary information culture, but decisions have been made for ages. They may be applied to both minor and commonplace things yet also to more distant and abstract ideas. Unfortunately, it frequently appears that individuals do not have comprehensive information or understanding regarding the occurrence of specific instances or associated elements. This is where the notion of making judgments under situations of imperfect knowledge arises, and this brings us to the notion of risk and uncertainty that do not constitute identical concepts. A distinction between risk and uncertainty was initially drawn by Frank H. Knight (1921) in his pertinent book "Risk, Uncertainty, and Profit":

- risk is present when future events occur with measurable probability;
- uncertainty is present when the likelihood of the occurrence of future events is indefinite or incalculable.

In the world of digital economy, the notion of uncertainty is met frequently compared to the concept of risk, since in a continuously changing globalized environment many aspects are uncapable of being quantitatively evaluated (or more in general terms: to be represented in an absolute/measurable fashion). In addition, uncertainty, as a feeling of doubt, induces the decision maker to spend extra time or to postponement from making the decision altogether (Lipshitz & Strauss, 1997). Making decisions under situations of uncertainty becomes more challenging when compared to settings of risk since it is impossible or barely possible to assess the likelihood with which a specific event will happen (Redziak, 2013). It is undisputed that the digital economy (sometimes synonymously termed as the "new economy") highlights measurable economic metrics such as rapid growth, low rates of inflation and minimal unemployment (Kling & Lamb, 1999; Gumah & Jamaluddin, 2006). Typically however, in the present global state of affairs, we confront unforeseen and exceedingly difficult to assess occurrences. They might include political upheaval in certain nations or regions in the world, but also meteorological anomalies and climate change, or lastly perpetually shifting consumer technology and societal developments.

Prior to performing a more thorough examination of the issue of decision making under circumstances of uncertainty in the digital economy, it is worthwhile commencing with an attempt to define the idea of "digital economy". According to a World Economic Forum (WEF, 2015) report, it can be inferred that the digital economy is a recently-emerging phenomenon that is becoming more significant given forecasts of double-digit yearly growth worldwide, with the global South seeing especially substantial expansion. WEF rapporteurs properly noted that this is a phenomena closely tied to the digital economy, which is the province of highly advanced or rapidly emerging nations, but its more comprehensive explanation was offered at the end of the previous century by Kling and Lamb (1999, pp. 17-18). They establish the notion of the digital economy as “the use of information to interact and communicate in a globalized, high-tech economy”. Tapscott (1995), in his term, pays attention to the seamless procedure for transition from the traditional (analog) model of information flow to the present scenario – which is quick and digital. Yet, it was pointed out that the process of digitizing information in the age of digitization is not necessarily strongly connected to the Internet (van den Berg, 2007). It is noteworthy that the digital economy impacts the life of each person but also of the society as entirety, which is changed towards an electronic society (or the so-called “e-society”) (Carlsson, 2004; Ulieru & Verdon, 2009). E-commerce, as an indivisible subsegment of the digital economy, is deemed by scholars to be the top priority when it comes to measuring (Mesenbourg, 2001).

The most prevalent root of feelings of doubt is the absence of knowledge or means of obtaining it (Conrath, 1967). The problem of uncertainty may also emerge in an inverse way: whenever there is an abundance of knowledge that is often conflicting (Weick 1979, 1995). In the age of broad access to the web, it is the availability of information that suggests a regular trouble with decision making that a stakeholder of the digital economy is confronted with. The uncertainty in reaching a decision in a digital world is much higher, since an overabundance of opposing information promotes a blurring of the boundaries between important knowledge and fake news. The purpose of the decision maker is to eliminate uncertainty as much as feasible using a logical and sensible attitude and conduct (Griffin, 2006, p. 192).

After the final decision has been reached, the decision maker depends on maximizing the advantages (Kubiczek, 2019). In the field of economics, however, there are various approaches to the criteria for making optimum judgments under circumstances of incomplete information, when benefits are being maximized:

- Wald’s pessimism criterion[1] – is based on the assumption that the decision maker should always be pessimistic, that is, assume the maliciousness of factors inside and outside the organization (Redziak, 2013, p. 106),
- Hurwitz’s optimism criterion[2] – it is assumed that the decision maker should rank the strategy based on the weighted average of the security level and the level of optimism (Hurwicz, 1951, 1952),

[1] Wald’s pessimism criterion is also described as “the rule of highest security” (Bolesta-Kukułka, 2003, p. 190).

[2] Hurwitz’s optimism criterion is also described as “the rule of greatest gain” (Bolesta-Kukułka, 2003, p. 222).

- Savage's regret criterion – it consists in selecting a decision to minimize the maximum relative loss based on a relative loss matrix (Savage, 1961),
- Szaniawski's prudential (caution) criterion – takes into account the weighted sum of the smallest and the average decision utility (Redziak, 2013, p. 111),
- Laplace's equal likelihood criterion[3] – assumes that since the decision maker is not able to determine which scenario will ultimately occur, he may assume that all states of nature are equally probable (Render et al., 2006).

In the age of digitization, so as to eliminate the sensation of uncertainty, it is required to filter the acquired information so that only the most trustworthy one remains. The process of getting accurate data, nonetheless, is frequently exceedingly expensive (Kubiczek, 2018).

To cope with the appearance of uncertainty while making a choice, Smithson (1989, p. 153) delivers a 3-stage method, which is consistent with established Western norms, to combat this occurrence. First, the degree of ignorance should be decreased as substantially as possible by collecting comprehensive knowledge and interpreting that information. At this juncture, it is essential to clarify the idea of "ignorance" which Smithson (1989, p. 1) views as "either the absence or the distortion of 'true' knowledge, and uncertainty as some form of incompleteness in information or knowledge". Then, if step one is incomplete, obtaining maximal control or predictability can be attained by learning or appropriately responding to the environment. Finally, if the amount of ignorance is irreducible, uncertainty should be dealt statistically.

Still, it should be taken in consideration that occasionally it is hard to totally eradicate the uncertainty issue. This is when a random, unforeseeable, and novel phenomena needs to be dealt with. One instance is the emergence of the COVID-19 pandemic, which is regarded as "world-changing" and an extraordinarily tough undertaking for policymakers – from politicians through healthcare services to businesspeople and upper management (Dwivedi et al., 2020). Investigators of this emergence demonstrate towards the fact that the information on the SARS-CoV-2 coronavirus (and hence the prevailing pandemic) is imperfect, i.e., incomplete, imprecise, uncertain, unreliable, vague, or partially true, and also associated with an overwhelming feeling of uncertainty and high risk (Dwivedi et al., 2020; Lodge & Boin, 2020).

[3] Laplace's equal likelihood criterion is also described as "the rule of insufficient reason" (Render et al., 2006).

The notion of sustainability is also vital to the digital economy. The level of development of the information society reflects the possible sustainability via harmony and balance among the economic, social, and environmental aspects (Buchanan et al., 2005). The idea of sustainable development has been adequately stated by Kajtazi (2010, p. 149)[4]: "an international interdisciplinary effort to blend environmental and economic goals which requires different meanings, in different contexts, and at different times". Until recently, both principles (digital economy and sustainable development) began to be classified as new connected aspects in the realm of artificial intelligence, as highlighted by Kajtazi (2010). She also states that a great deal of organizational information transactions which are carried out via ICT are to replace the outdated paper-based approaches. In this process, however, several substantial barriers have arisen: needless additional labor was produced particularly because of the digitalization of the business, and therefore inefficient information became asymmetric (Kajtazi, 2010). This may be deemed an inadequate development of sustainability under the conditions of the digital economy. Therefore, it is worthwhile taking a look to the idea of information asymmetry. The following part of this book will help understanding this occurrence and hence provide steps that may be put forth in order to lessen this phenomenon.

One of the most engrossing and intriguing instruments utilized by contemporary economics is the theory of information asymmetry. Thinking about the word "asymmetry", the right thesis of a "lack of coherence, harmony" or "imbalance" easily arises. In general, information asymmetry is apparent when the vendor (the firm that sells the goods) has more information about the product being offered than the consumer intending to purchase it (Akerlof, 1970). In addition, when the cost for acquiring information about a particular product differs across consumers, the phenomenon of information asymmetry emerges as well (Stigler, 1961). Presently, IT technologies come to the rescue, eliminating the situation of information asymmetry in the seller-consumer connection. Within the context of access to information, information technology may be characterized as a system, such as the Internet, which can be utilized by firms to transmit information that can in turn be used by consumers (Kulkarni, 2000). It is also obvious currently – the Internet is a frequently utilized IT tool (system), which was designed especially to obtain information. Research reveals that the

[4] The definition was formed on the basis of the works of McFarland et al. (1996) and Buchanan et al. (2005).

majority (approximately ¾)[5] of academics and university students prefer the Internet as the source of information (Rangaswamy et al., 2017).

As previously mentioned, the purpose of searching the Internet is to obtain information that may become useful. For example, if we want to buy a specific product (or a product type), irrelevant whether stationary or online, we explore internet resources, wanting to get as much information as possible about it. In this case, we often come across product reviews, as well as user reviews (about the product or seller), which basically turns out to be useful when making purchasing decisions. The information that a person searches for and obtains from online resources, although it affects the buyer's way of thinking about the transaction and the pre-trade environment, in many cases is not balanced. As a result, the consumer encounters information asymmetry that will always be unfavorable to him. This situation causes consternation in making own decisions, not only those regarding purchases (Harford, 2005; Kajatzi, 2010).

One of the finest instances of information asymmetry in uncertainty was provided by Akerlof in 1970 in the work entitled "The Market for Lemons: Quality Uncertainty and the Market Mechanism", which culminated in the Nobel Prize in Economics. Akerlof, in this work, addresses information asymmetry on the case of the automotive industry between two actors: a customer and a seller. In the described case, the salesman who wants to sell a used (or even a new) vehicle has more knowledge about it than the client. Due to this, he has a secret edge over the customer, since he can only provide information that would be favorable to him, and conceal, for example, car flaws that could dissuade a possible buyer. Akerlof's theory is that the whole knowledge about the object being sold (in the example: the automobile) is unilaterally polarized towards the seller. This confirms his viewpoint that without full information, the consumer might acquire the eponymous "lemon", which is a car in dreadful condition. Akerlof's mentioned paper has helped to authoring this work and prompted the author to examine the automobile business, in particular the used motor vehicles sector.

The problem of information asymmetry, which was found by Akerlof, obviously, does not just take place in the automobile sales market, but has also been documented in numerous businesses, even those without direct ties with the area of economy.

[5] Specifically: 74.44%.

The incidence of information asymmetry does not solely affect the purchaser, but differing information resources could be relevant to the bidder (Jaremen & Nawrocka, 2015). This phenomena strongly indicates the market dysfunction and the formation of transaction costs, which arises from poor allocation of resources (Stiglitz, 2002; DiLorenzo, 2011). It could appear that in order to lessen or perhaps entirely eradicate the asymmetry of information, it is required to access and collect information. However, as stated by Jaremen and Nawrocka (2015), expanding the information resource will not decrease the prevalence of this phenomenon. This is owing, for example, to the restricted capabilities of an earlier (i.e., before acquiring the information) meticulous assessment of the quantity, quality, and cost of information essential to make a choice. Moreover, a sunk cost problem emerges here, since the quantity and quality of information typically do not directly transfer into the breadth of its usage (Kasper & Streit, 1999). It should be stated that giving information at the right time in the right location, and in the appropriate form is not an easy task and requires the application of efficient procedures and instruments. It is problematic since development is intimately tied to uncertainty in solving issues (Heisig et al., 2020).

Information asymmetry has multiple negative implications worth noting:

- Negative selection – a person who does not have full information about the product (or more broadly: good), aided by partial or erroneous information, may pick a variation that is not perfect for him. With further information about a specific good, he might choose a different, more optimum option. Then knowledge, and even wisdom, could potentially attained. The missing, skewed, or incomplete knowledge leads to a poor allocation of one's resources in the Pareto sense (Santarek, 2017).
- Moral abuse – market actors who have more information about a good in the economic connection, by withholding parts of it, may influence the other player of the relationship, who does not have complete information.
- Costly verification of the condition – a person who does not have comprehensive knowledge needs to verify the subject matter of the economic connection, which may result in higher expenses, as noted earlier (Kubiczek, 2018).

It ought to be highlighted, however, that the regulated transmission of knowledge may have good impacts by minimizing its asymmetry. Such a circumstance may take happen, for example, during the announcement of a takeover of a certain company by another, because it is compelling both to investors and analysts. It conduces to the spread of a considerable volume of information. Such a condition produces a considerable disparity of knowledge between those from the internal structures of the organization (e.g., management) and those outside of it (e.g., investors). The position of the bidder itself is then being reevaluated (Draper & Paudyal, 2008).

Nevertheless, there are strategies to lessen the detrimental impact and influences of information asymmetry. These include governmental measures aimed at providing equitable access to information, or practices of a firm targeted at inspiring the customer's trust. Such actions include taking care of the reputation and good image of the business, having certifications and diplomas certifying personnel credentials, as well as standardization networks (hotels, shops, restaurants, etc.), examining complaints or offering guarantees. It is important to recognize that one of the strategies in countering the stated phenomena may be the construction of appropriate IT solutions in the sphere of information accessibility or processing. For example, Freedman and Jin (2011), with the use of data gathered from the Prosper.com website, not only identified the extent and variations of information asymmetry but also made steps to reduce its occurrence and quantify the risk. Therefore, it might be stated that the online scraping tool (used to collect data from the Internet) assists to considerably reduce the incidence of uneven access to information, and therefore to lessen the sense of uncertainty in economic terms.

Information asymmetry, which is closely connected to the phenomena of decision making under uncertainty, is an economic issue that was articulated decades ago and has been already explored. The aforementioned are multifaceted difficulties that practically every sphere of life has to cope with, including the e-commerce business. However, as e-commerce is a core component of the digital economy, there are current solutions, such as the previously mentioned data extraction, that seek to mitigate negative impacts. Unequal access to information, however, is not the only issue that the e-commerce business has to overcome, which states as an axis of the following Subsection 1.2.

1.2. Challenges of the e-commerce sector

It is worthwhile beginning with the realization that e-commerce cannot exist without access to the Internet, and it is always considered as an aspect of globalization, not only in the international setting. One of the primary issues that the e-commerce sector needs to tackle is equitable and widespread access to the Web. In the Old Continent, the problem of access to the Internet is modest, as the coverage was as high as 90% in 2019, which is a jump of 26 percentage points when contrasted with 2009 (Eurostat, 2020). In contrary, the Internet coverage ratio to population (the so-called "Penetration Rate") in June 2020 was as follows: in Far East Asia 58.8%, in Africa 42.2%, and in North America 90.3% (Miniwatts Marketing Group, 2020). Further, the so-called universal service concept mandates that throughout the European Union, customers must be able to obtain excellent quality electronic communication services at affordable costs such as basic access to the Internet (Kamiński, 2003). However, when we look at undeveloped or emerging regions, the lack of access to the Internet is substantial for the people of these places and styles one of the most critical impediments to the expansion of the online retail sector.

The researchers also note that apart from exceptionally poor access to the Internet, developing nations still have to grapple with tax challenges, including an ambiguous fiscal system stemming from inadequate legislation and weak infrastructure (Reddy et al., 2014). These are the variables of the macroeconomic environment which adversely impact the sector. Beyond the previously mentioned negative macroeconomic reasons impacting the e-commerce sector, the literature also contains others, such as globalization distorting competition and the ensuing inclinations for oligopolistic companies (Hadasik, 2020). These restrictions are especially essential in the international environment, when trying to sell cross-border. It is worth mentioning that drivers for merchants and customers for cross-border e-trade vary as stated by Kool et al. (2011). Moreover, they also remark that in the case of foreign sales, the biggest challenges for retailers can be the lack of relevance, IT skills scarcity or cross-border payments and logistics, but also administrative and legal issues.

COVID-19 pandemic is additionally recognized as a crucial element impacting the e-commerce sector. Coronavirus outbreak has been a volatile and unpredictable phenomenon, which has prompted a shift in consumer behavior (Kubiczek et al., 2021) and produced additional uncertainty in the markets. Due to the dynamic evolution of the pandemic, the assessment of natural elements is now wrong, which makes it impossible to anticipate the expansion of the e-commerce business (Hadasik, 2020).

The lack of trust and consumers' loyalty toward not just individual enterprises, but the whole e-commerce market is also observed as a development hurdle for the whole sector (IAB Polska, 2013). For example, enterprises with a favorable worldwide image, such as Nike or Apple, profit from their popularity and frequently do not need additional steps to increase consumer loyalty and trust, as is the case with smaller entrepreneurships (Kalinić, 2014). Trust is described in the literature not simply as a hindrance to the entrepreneurship's processes yet as a risk that ought to be considered in the company's growth plans (Almousa, 2013). Olender-Skorek, Czarnecki, and Bartoszewska (2011) suggest that, besides the lack of confidence in Internet merchants, cybercrime presents as one of the key obstacles to the expansion of e-commerce, as it raises the dread of e-shopping among internet users.

Apart from social reasons, the literature also contains simply technological hurdles to the entry and operation on the e-commerce industry. Zaied (2012), when defining the impediments to adapting the e-commerce sector to six areas (social and cultural; technological; economical; political; organizational; legal and regulatory, technical), recognized technical difficulties as the most critical ones. Makowiec (2008) mentions technological restrictions such as inadequate bandwidth, expensive server maintenance costs, and challenges with the integration of e-commerce services with the current ICT solutions established in the company as the most critical impediments to the growth of electronic commerce. Chitura et al. (2008) demonstrate that practically all the challenges, including hurdles to entry, that an e-commerce firm in its early stage of growth had to contend with, also remain today. They are also concerned about a lack of technical talent and IT knowledge among employees, absence of adequate e-commerce software solutions for SMEs, undersupply of technology awareness or restricted dissemination of computers.

Among the factors impeding the start, expansion, and effective functioning of an e-commerce enterprise, academic circles also mention a lack of understanding of business approaches and techniques but also the costly nature of development and implementation of computer systems and environments tailored to the sector (Kapurubandara & Lawson, 2006). At this point, it should be underlined that the costs of launching an electronic company are typically lower than in the traditional model overall. It happens that apart from solely financial factors preventing the rise of e-commerce business, there are other limitations, such as the reluctance of top managers to employ cutting-edge techniques (Duan et al., 2012), the absence of long-range company plans (Arendt, 2008) or the lack of skills and knowledge (Abid et al., 2011).

An essential problem that has to be taken into consideration by any e-commerce organization is the local law surrounding the handling and safeguarding of personal data. In the European Union, all legal concerns to which any business (not just within the e-commerce sector) must react is controlled by the General Data Protection Regulation (GDPR), which came into force on May 25, 2018. Any firm in this industry, desiring to stay on the market and be competitive, categories personal data according to the type and amount of data handled. For this reason, sets of personal data with a suitable structure are utilized internally, which means that they are grouped in such a way that makes it feasible to search for particular data according to a specified criterion (Hadasik, 2019; Polish General Inspectorate of Personal Data[6]). It should be mentioned that the GDPR pertains solely to the following two scenarios (European Union):

- the company processes personal data and is based in the EU, regardless of where the data is actually processed;
- the company is based outside the EU but processes personal data in relation to offering goods or services to natural persons in the EU or monitors the behavior of natural persons in the EU.

Non-EU enterprises handling EU citizens' data must designate a representation in the EU. Further local data privacy regulations may apply to further overseas sales.

The above-mentioned hurdles and impediments may be fluidly separated into two categories: barriers for entrepreneurs and barriers for customers, nonetheless in many situations these impediments overlap or have points of interaction with each other (Kool et al., 2011; Kalinić, 2014). The first category contains all legal difficulties to which an e-commerce corporation must adjust, while the second group includes customer biases and confidence in internet commerce.

Szpringer (2005) brings emphasis on the fact that in the scope of operating a business, particularly in the e-commerce sector, one should behave in line with the "philosophy of balance", where both the entrepreneur and the customer are offered protection of the same degree. The customer, who is in an economic connection with an entrepreneur, serves as an actor in a disadvantaged position due to information asymmetry compromising him the most. It is vital to maintain an equilibrium between the two aims of consumer legislation, i.e., proper information and openness, and protection against circumstances of pressure and surprise (Hadasik, 2019).

At this stage, it is important to note that studies on the field of e-commerce in established and emerging economies demonstrated variances in the kind of variables impacting firms based on their type: business-to-business (B2B) and business-to-customer (B2C). While B2B enterprises are largely driven by global factors, B2C are impacted by local occurrences (Gibbs et al., 2003).

[6] After the entry into force of the GDPR, the Polish General Inspectorate of Personal Data changed its name to the Polish Office for Personal Data Protection.

Bearing in mind the obvious and widely described advantages of e-commerce, which were even more pronounced during the COVID-19 pandemic, the above-mentioned obstacles, barriers, or more broadly: the challenges, that the entire industry must face can be reduced after applying the actions proposed by the author to further expand the sector.

Firstly of all, for the purpose of eradicating the fundamental barrier, which is the restricted access to the Internet in insufficiently developed and emerging areas, initiatives geared toward reducing the information (internet) exclusion on both the national and local level alongside those implemented by international organizations and campaigns should be enacted. Transnational programs such as Project Loon by Alphabet LLC (which is the parent company of Google) and Internet.org, which was launched by Facebook (now changed into the "Facebook Connectivity" initiative), aim to build a "global Internet". Such a formulation may sound rather pleonastic but, in simple words, these notions imply the spread of internet connectivity even in poor and hard-to-reach locations. By decreasing the phenomena of online exclusion, both the worldwide and local growth of the e-commerce sector will be more active.

It is also worth enhancing citizens' understanding of conducting purchases via the Internet, particularly in emerging economies, through broad educational campaigns from an early to older age, including by presenting suitable talks at schools as well as retirement homes. Highlighting the advantages (including flexibility, accessibility, and convenience of use) as well as rejecting fake news in the area of online shopping will have a beneficial influence on the perception of e-commerce and will boost customer trust in the sector.

In order to decrease the absence or restricted knowledge of employees about business models, entrepreneurs should engage in the ongoing growth of their labor force by providing them with the chance to study in courses or attend specialized conferences. Employing highly skilled workers at top management levels is one of the keys to the success of an e-commerce organization, because the foresight of the managerial team and the implementation of long term plans may assure appropriate and continuous development of the company (Arendt, 2008).

An e-commerce firm in the first stage of development, intending to minimize additional investment expenses in the case of IT services, may choose for management, HR, and payroll applications, etc., which are located and run in the cloud. Without investing in servers, technical supervisors or server administrators in the initial phases of company's expansion, it should be kept into consideration that it is difficult to avoid fixed expenses in the form of subscriptions for cloud services or outsourcing. In following phases of development, the transformation of IT solutions into “local” (i.e., with a server with software stationed and controlled locally, generally at the company’s premises) may show to be a more appropriate option. Always be guided by the optimum options for the welfare of the firm.

When planning their development, e-commerce firms, when building a software, should incorporate an agile management system at every employee level. The Agile Manifesto (Beck et al., 2001), which was announced in 2001, had the following pillars in its original form:

- people and interactions over processes and tools,
- working software over detailed documentation,
- cooperation with the client over contract negotiations,
- reacting to changes beyond the implementation of the assumed plan.

Existing organizations with traditional business strategies might explore a move to agile. Scientific studies have demonstrated that agile management is the ideal route of an enterprise’s development, which is defined by flexibility, efficiency, and a suitable solution to issues with the modern setting (such as the concepts of *software as a service* or *open-source software*) (Ågerfalk et al., 2009). It should be highlighted however, that the enhancements to the agile concept are also essential: the present directions regarding development concentrate on the increasing significance of empirical research, bringing together knowledgeable leadership teams, providing greater prominence to management oriented approaches, and finally obtaining the essence from the source ideas of agile software development rather than emphasizing the enhancement of their understanding (Dingsøyr et al., 2008). Such a commercial approach to software development may assure the success of a firm, but e-commerce enterprises employing external software (e.g., in the form of outsourcing), without necessarily being its creators, can also profit from the foundations underpinning The Agile Manifesto. It is mostly about the human-oriented approach, interaction with the customer, and the need to execute changes.

Naturally, as noted in Subchapter 1.1, the asymmetry of information is a substantial barrier for the e-commerce industry and the broader service sector, or even more broadly: the worldwide economy. In order to lower it, it is critical to aim for an information balance between the information issuer and the recipient. An advertisement portal, which is an IT tool used in e-commerce, might show to be a suitable medium to prevent this discomfort, which may impact sub-optimal decision making by customers.

1.3. Advertisement portal as an e-commerce tool

Prior to making the ultimate choice about the purchase of a specific commodity (or not buying it), the customer seeks for information and collects it (Tillström, 2012). The search for information about a certain product takes occur with the engagement of the customer in the very category of that product (van Rijnsoever et al., 2009). This commitment pertains to the relation of the consumer's requirements, interests, values, and priorities toward a certain product, and is also significantly tied to the customer's experience and empirical knowledge about the product category.

The advertisement portal (or: classified ads website)[7] is a form of website that allows us to publish adverts and to read them, and therefore to associate the parties to the transaction (buyer and seller). Advertisers frequently include things in relevant categories, which makes it easier for consumers to discover the item. Auction websites are a specific form of advertising services, which, however, are not the topic of the work. The classifieds portal is a sort of a web tool that brings together buyers and sellers and gives information about a certain product that the buyer is interested in. A potential consumer, i.e., a user of the advertising site, will receive only as much information as the advertiser wishes to disclose. Advertising portals are fairly similar in their specificity – regardless of whether it is a site with modest advertisements, real estate or automotive. It should be highlighted that specialist websites with adverts, such as aforementioned real estate or automotive, are extra loaded with attributes (parameters) particular to a given product category, which may most often be filtered.

Advertising portals have the largest reach in terms of advertising the sales offer (possibly: rental/lease) on the Internet (Martyniak, 2015). On advert portals, housing developers or auto dealers regularly market themselves in tandem to their corporate websites as an alternative advertising channel. Intermediaries utilize classifieds websites as they provide them with an opportunity of endless exposure, a place to publish photographs, access to perpetual modification of the offer and, most crucially, a place to put direct connections to their own websites (Rodzeń, 2011, p. 215). The major advertisements services, in addition

[7] Later in the work, the terms *advertisement portal/website/service*, *advert portal/website/service*, *ad portal/website*, *classified ads portal/service/website*, and *classifieds portal/service/website* will be used interchangeably.

to their existence on the Internet, regularly promote themselves through other channels, such as radio, television or press, that possess a wider audience than local, and this implies an increased likelihood of getting to know the offer there by an interested party (Martyniak, 2015).

The quantity of offerings listed on the top classifieds websites approaches hundreds of thousands of advertisements[8]. Such an immense repository of parametrically structured items allows greater access to data on current offerings and pricing on the market to potential clients and, thus, a larger possibility of discovering the suitable product. The enormous number of adverts on these portals indicates the considerable interest among organizations putting the advertisement and people accessing them. Yet it should be observed that this produces greater competition; thus, it is important to emphasize the offers (often for a charge) using the tools given by the website, so that the commercial is not overlooked in the sea of other offers and is effective (Martyniak, 2015).

Ad portals have a pretty basic structure, which comprises the following components: search engine, category tree, filters, and a sorting tool. The search engine can be coupled with the category tree and the filtering tool (as is the case with the OTOMOTO web), which narrows the search results to certain parameters. In general, the search results may be ranked by price, date added or name (ascending or descending) using the sort function. Frequently, advertising portals utilize an algorithm for prioritizing search results in connection to the usage of highlighting – those ads that are “featured” are displayed before those “not featured” in the search results. There are studies that indicate a favorable association between consumer behavior and the prediction of the usefulness of tailored search results for the end user (Liu et al., 2012; Mao et al., 2017; Vakkari et al., 2019). Specialized advertising portals – such as a site with automobile advertisements – have precisely specified criteria that pertain to a single sector with which a specialized portal is related[9]. Regarding the instance of the automotive portal, these filters concern, among others: car brand, model, engine capacity, fuel type, equipment, etc. There are other filters that apply no matter what the expertise of the classifieds page is. Such a filter is, for

[8] Number of advertisements on the largest Polish portal with automotive advertisements OTOMOTO: 231,109 (as of October 22, 2023; source: otomoto.pl). Number of advertisements on the largest Polish portal with advertisements of various categories OLX.pl: 24,630,218 (as of October 22, 2023, source: olx.pl).

[9] In the case of advertising portals with various/minor advertisements, strictly defined filters usually apply to subcategories – for example, some filters apply to the furniture category, others – the fashion category, and others – the sports category.

example, a location filter with which we may narrow down the search results to a specified city, county or province. A localization filter frequently has a radius that may be used to calculate the distance from a certain spot. Thanks to it, we may, for example, filter our search results to these from Warsaw and 10 kilometers distant from the location.

The foregoing properties of the advertising platform suggest that it is efficiently minimizing information asymmetry. In particular, the potential buyer may pick an item that suits him utilizing in filtering and sorting capabilities. Customers regularly utilize these tools, so a failure to fill in the value in the property fields (e.g., of a car – in the case of an automotive classifieds website) would omit the listing's position from the filtered search results. The vendors, particularly in particular auto dealers, may be motivated to enhance the criteria of the marketing. It also is highly useful for potential buyers since the advertisement with the filled features of the offered thing gives them with more comprehensive knowledge about the goods before the purchase, and so the imbalance between the seller and the client is minimized.

The world nowadays is fraught with problems and hence of consequent uncertainty. In the face of the coronavirus pandemic, numerous realms of operation of states, organizations (including, and possibly in especially, from the e-commerce industry) or homes had to be accommodated to the backdrop of uncertainty. This uncertainty which permeates numerous sectors of life, frequently pushes judgments surrounded by it and thus generates the phenomena of information asymmetry. It can be efficiently decreased by economic instruments and strategies, such as the abovementioned advertising portal. By filling in the qualities of the advertising, the potential buyer acquires the required information that will assist him make the final purchase choice. In order to decrease the asymmetry of information even further, web scraping turns out to be beneficial, which will be detailed in more depth in Chapter 2. The following Chapter will also show the necessity for employing data extraction technologies, their qualities, as well as the potential and hazards (particularly in terms of ethics and legality). All these characteristics will also pertain to the elimination of information asymmetry.

Chapter 2. Web scraping

Data extraction tools and procedures show to be valuable for many reasons, such as data analysis or decrease of information asymmetry. While web scraping offers huge advantages, it also has notable repercussions that should always be addressed while gathering data. In this Chapter, the necessity to employ web harvesting tools for commercial and scientific reasons will be given, as well as all features of tools and techniques for acquiring data will be considered. Moreover, their advantages and hazards will be contrasted in the ethical and legal settings.

2.1. The need to implement automated data acquisition tools

Tools for obtaining data from the Internet directly connect to the notions of data and information, as well as the links between them. The next subsection 2.1.1 gives a theoretical framework of these notations, serving as the basis for future exploration of the notion of web scraping.

2.1.1. Data and information as a basis of the notion of web scraping

At this point, it is worth developing the concepts of *data* and *information*, and the associations between them, as an extension of the themes discussed in Chapter 1.

Our universe encompasses a variety of data since it surrounds us. Distance, weight, color, volume – all these may be data, as data can be anything that we can detect by our senses. A particular aspect of digital data is that it may be labeled as binary (Hilbert, 2016, pp. 70–71). Answering the question: "What is *data* and what role does it perform?" may prove to be a reference point to the core of web scraping as its own.

As *data* has various definitions depending on the context, it can be beneficial to provide the term used in the Knowledge Sciences that "[*data* is] unprocessed information" (Hey, 2004, p. 5). In information theory, *data* is defined as "symbols without any meaning or value" (Hilbert, 2016, p. 70). Definitions used elsewhere refer to *data* as a "representation of objective facts" (Hey, 2004, p. 5). Nonetheless, it is worth returning to the first definition and focusing on the reference to *information*. At this juncture, the notion of the DIKW hierarchy (which is an acronym of data-information-knowledge-wisdom) deserves to be mentioned. This pyramidal hierarchy was introduced by Sharma (2004), who provided the framework for additional discussions in the domain of Knowledge and Information Management sciences. It ought to be underscored that Sharma (2004) just consolidated and structured the already available information, referring to the works of Zeleny (1987) and Ackoff (1989) in this subject. Figure 2.1 represents the so-called "Knowledge Pyramid", which demonstrates the links between the discussed ideas.

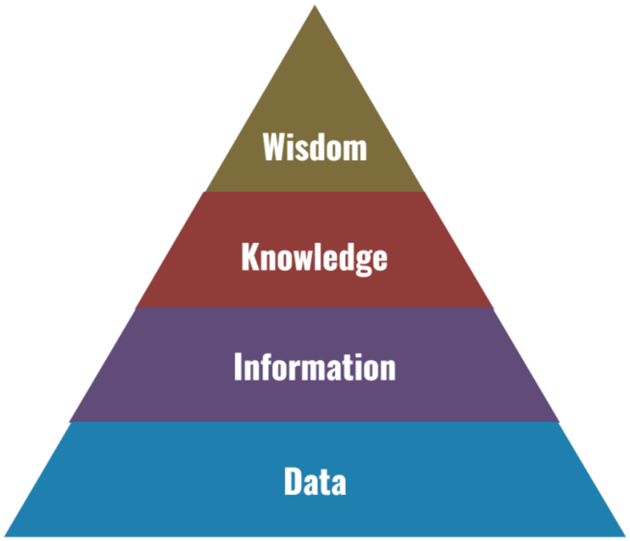


Figure 2.1. The Knowledge Pyramid (DIKW hierarchy)

Source: (Hey, 2004, p. 3).

The Internet, being a special and indivisible aspect of the current digital economy and, thus, a worldwide instrument with broad availability, has endless quantities of data that everyone of us (under certain conditions) may have access to. The World Wide Web, the latest version of the original information management system, is no longer merely a document network but a data network (Berners-Lee & Fischetti, 1999). Kinne & Axenbeck (2018, p. 1) take an even wider look at the WWW and characterize it as "a ubiquitous medium for communicating and disseminating information". In view of the fact that involvement in the online world is vital, it is advised that firms engaged in technological transformation empower a rising number of data consumers engaged in processing of data (Labadie et al., 2020). Data in its raw form, i.e., objective facts, as indicated in the definition supplied by Hey (2004, p. 5), is regrettably sometimes hard to comprehend by the analyst. It has to be digested in order to retrieve information which is positioned one level higher in a pyramidal notion, as represented in the aforementioned Figure 2.1.

Information as a concept is hard to describe, although multiple attempts have been made to develop this idea. Depending on the context, the definition of *information* may vary. Kullback and Leibler (1951) have managed to describe a mathematical expression of *information*. *Oxford English Dictionary* (1989, pp. 944–946) made the following distinction between the types of information: *information-as-process*, *information-as-thing*, and *information-as-knowledge*. This exhibits two sorts of information: tangible and intangible, as mentioned by Buckland (1991, p. 351). In fact, Buckland (1991) specifies four types of information because, apart from the above-mentioned distinction between tangible and intangible information, he stresses out that information may be both an entity and a process. This separation of information formats, together with instances of their occurrence in practice, is described in Table 2.1.

Table 2.1. Four aspects of information

	INTANGIBLE	TANGIBLE
ENTITY	*information-as-knowledge* (knowledge)	*information-as-thing* (data, document)
PROCESS	*information-as-process* (being informed)	*information processing* (data processing)

Source: (Buckland, 1991, p. 352).

An intriguing diligence to put it into words *information* was made by Bateson (1972, p. 272), who describes it as "[a] difference which makes a difference". The essential distinction is that information lessens uncertainty. Hilty (2010) in his lecture on information management exquisitely referenced Peter Drucker, a noted management guru, who said that "*information* is data empowered with relevance and purpose". Simply put: "*information* is *data* that has a meaning"; it is processed data that, after adding context and meaning, turned into facts (Cambridge International Examinations, 2015, p. 5).

To distinguish what *data* is and what *information* is, it is worth paying attention to whether the given binary symbol represents a "surprise". If not, it does not reduce the uncertainty and, therefore, it is not *information* but just redundant *data* (Hilbert, 2016, p. 71). *Information* can, therefore, be described as an antagonism of uncertainty. To extract *information* from the *data*, a "compression" is needed, which discards extraneous data, leaving just the differences that successfully reduce the impression of doubt (Hilbert, 2016, p. 71). The leftover amount of symbols is called "information source entropy", which was found by Claude E. Shannon and currently commonly referred to as "Shannon's source coding theorem" (Shannon, 1948; Cover & Thomas, 2006). The process of allegedly "converting" data to information was staged by Hilty (2010) as represented in Figure 2.2.

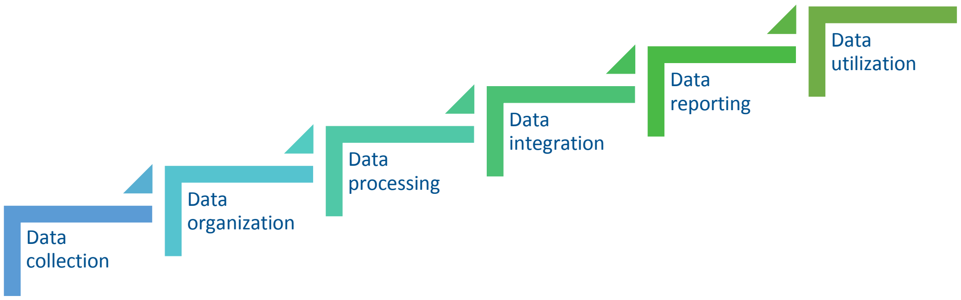


Figure 2.2. Six steps of conversion of data into information

Source: Own study adapted from (Hilty, 2010, p. 6).

The ultimate objective is to transform *information* into *knowledge* and later even into *wisdom*. *Knowledge*, in tandem with business intelligence tools, shows out to be effective in supporting activities connected to the quality of the decision-making process (Mach & Owoc, 2010; Wieder & Ossimitz, 2015). Still, additional processing of information is not the aim of this study.

2.1.2. Web scraping as a medium for obtaining information in a big data world

Why did the author decide to expand the idea of the relationship between *data* and *information* at the beginning of this Chapter, and not the previous one? Because the link between the presented theory and the idea of web scraping is of the utmost importance and underlying the data extraction as such. The mere word "data" is even entrenched in the term of "data extraction". Web scraping is recognized as one of the major strategies for collecting data (Fedak, 2018). The objective of web scraping is to go through all the phases of transformation of data into information that has been stated in the previously mentioned Figure 2.2.

In scientific publications, there are many equivalent phrases for *web scraping* (used interchangeably), such as: *web harvesting*, *web data mining*, *web content mining*, *web data acquisition* or *web (data) extraction* (Kosala & Blockeel, 2000; Fernández-Villamor et al., 2011, p. 451; Vargiu & Urru, 2013, p. 44; Mitchell, 2015, p. 10; Zhao, 2017, p. 1; vanden Broucke & Baesens, 2018, p. 3; Zamora, 2019). Hillen (2019, p. 3351) also points out *information scraping* and *screen scraping* as synonyms for *web scraping*. In the literature, however, the concepts of *data scraping* and *web scraping* are distinguished: the former denotes data extraction from a file using a computer program, and the latter explains the extraction of data from Internet sources (HTML files), also with the help of a computer program (Haddaway, 2015, p. 187). Moreover, the terms *web scraping* and *web crawling* are treated as separate and should not be used interchangeably (Kaspa et al., 2018, p. 50). *Web crawling*[10] merely refers to the process of mechanically visiting multiple web sites via indexing with hyperlinks, but no data is extracted in the process (Ceri et al., 2013, pp. 75-76; Krijnen et al., 2014, p. 1; Kaspa et al., 2018, p. 50; vanden Broucke & Baesens, 2018, p. 155). In general, automated programs that mine and search Internet resources are called *bots* (Mitchell, 2015, p. 10).

In the literature, we can find both definitions of *web scraping* as the method of obtaining either *data* or *information* from websites (Schrenk, 2007; Fernández-Villamor et al., 2011, p. 451; Haddaway, 2015, p. 187; Pereira & Vanitha, 2015; Xu et al., 2017, p. 489; Goldfein & Keyte, 2017; vanden Broucke & Baesens, 2018, p. 3). It cannot be stated that one of these definitions is wrong. Web harvesting is a procedure that first collects data, and then finally processes it in order to eliminate ambiguity, and therefore gain information. Slamet et al. (2018, p. 1) formed (on the basis of available literature sources) a definition of web scraping, with the inclusion of both the terms *data* and *information*, which reads as follows: "a process of automatic data and information collection from the internet, commonly in website pages using markup languages such as HTML or XHTML whose data analyzed for certain needs and purposes". Leaving aside the discussion of the relationship between *information* and *data*, which may slightly edge on philosophical, Glez-Peña et al. (2013, p. 789), in turn, aptly replaced the terms *information* and *data* with *content*. Hence, they defined the notion of web data scraping as "the process of extracting and combining con-

[10] Krijnen et al. (2014, p. 1) also provide synonyms for *web crawling*: *web spidering* and *web indexing*.

tents of interest from the Web in a systematic way". It should be emphasized here that online content mining is intimately connected to the notion of online automation (Bolin et al., 2005). It resembles natural human behavior when browsing Internet resources (Mitchell, 2013, p. 8; Vargiu & Urru, 2013), because it has been acknowledged that the World Wide Web was initially created to provide material to humans, not computers (Thomsen et al., 2012).

From all the data in the Internet area, the following subsets may be distinguished: structured, semi-structured, and unstructured (qualitative and quantitative), which originate from all Internet sources (Watson, 2014). Structured data is a set of data that has a definite and obvious organization and which, for instance, can be in a database (Arasu & Garcia Molina, 2003; Boronat, 2008, p. 8). Web scraping refers to obtaining unstructured or semi-structured data (that is, one that is readily provided on online sites to an ordinary web user) and then parsing it into a structured form that may even be relational (Kushmerick, 1997; Arasu & Garcia Molina, 2003; Eriksson, 2016, p. 5). Most choices made at the managerial level are usually made with the aid of structured and well-organized data (Himmi et al., 2017). Analyzes are also made on the obtained data (or during the collecting process), which has been correctly aggregated (Himmi et al., 2017). That is why it is so vital to leverage the potential of non-structured data, which may be turned into information utilizing web scraping technologies, as described earlier.

The procedure of collecting data via web harvesting tools, which was proposed by de S Sirisuriya (2015, p. 136), is illustrated in Figure 2.3.

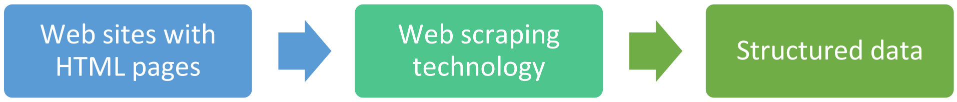


Figure 2.3. Simplified structure of web scraping – a process of obtaining data

Source: Own study adapted from (de S Sirisuriya, 2015, p. 136).

A similarly looking process was proposed by Krotov and Tennyson (2018). However, they state that web page analysis is a preliminary step to collecting data.

Due to the sheer volume of heterogeneous data available and regularly established on the Internet, web scraping is regarded a feasible, helpful, and robust medium in big data processing and analysis (Zhao, 2017, p. 1)[11]. This is since data extraction is regarded as an initial phase before an investigation or

[11] Based on: Mooney et al., 2015; Bar-Ilan, 2001.

development employing data science (Chaulagain et al., 2017). Big data is a relatively fresh solution under ongoing development that may be utilized to create a competitive advantage in a sustainable manner (Olszak & Mach-Król, 2018, p. 2). There are also studies that explore the use of big data in firms with the purpose of becoming more sustainable and, consequently, competitive (Mach-Król, 2017). As a consequence, web scraping may turn out to be one of the important ways to attain sustainable development goals by companies. As it has been acknowledged that organizations have to tackle growing difficulties and possibilities almost in real time, big data is proving to be beneficial in accomplishing such objectives (Yang & Meyer, 2015; Bartuś et al., 2017; Mach-Król, 2017).

In today's age of digital information, the major challenge is not the extraction of data itself, but the extraction of the most relevant and hence non-redundant information from such data (Kozak et al., 2020). It has been recognized that the appropriate and extensive use of big data in businesses may be considered as a medium for producing their value, termed as "significant value hidden in data" (Olszak & Zurada, 2020). Moreover, Khalil and Fakir (2017) observe that in order to boost the accuracy of web harvesting outcomes, only important data should be extracted. Value is considered as an extra pillar in the LaValle et al. (2011) 4V big data descriptive paradigm, alongside Velocity, Veracity, Volume, and Variety (Manyika et al., 2011; Erl et al., 2015; Himmi et al., 2017). The suggested 5V model, featuring Value as one of the five axes, is presented in the Figure 2.4.

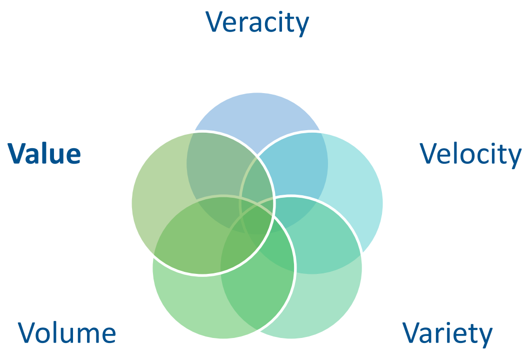


Figure 2.4. The 5V of big data

Source: Own study adapted from (Himmi et al., 2017, p. 1054).

It should be emphasized here that Erl et al. (2015) as well as Manyika et al. (2011) differentiated two more pillars of the "V" big data idea, which coexist with the abovementioned: Variability and Visualization. Acquiring data via the Internet is connected to dealing with technological challenges associated to the "V" qualities of big data (Goes, 2014; IBM, 2018)[12].

Big data, along with the overall data collecting process, finds a variety of practical uses from diverse scientific domains. In the course of the Web 2.0 revolution, which took place in the second half of the first decade of the twenty--first century, when instruments employing user-generated content (UGC) started showing up in everyday use, such as social media platforms like Facebook or Twitter, the significance of big data analysis has grown into one of the topconcerns and challenges for companies (Fernández Villamor et al., 2014; Chong et al., 2016, p. 359; Xu et al., 2017). Nowadays, while the growth of social networks and other portals is exceedingly evolving, it is vital to adopt complete and unorthodox analytical solutions, therefore web scraping is equally applicable with social media analysis. It may also be shown that online scraping finds practical application in e-commerce: for example, among airline comparison websites (Poggi et al., 2007), or when corporations scrape pricing from their competitors' websites (Stiving, 2017). Furthermore, there are investigations that use obtained data to construct telematics tools to study driver behavior (Andria et al., 2016). Another work illustrates the practical use of data acquiring techniques to improve the neutrino telescope (Aguilar et al., 2007). The adoption of the data collecting system also turned out to be beneficial in the development of power quality monitoring (Chen et al., 2009). Lastly, in the midst of a sudden and ever-shifting pandemic situation triggered by the rapid spread of the SARS-CoV-2 coronavirus, web scraping (in conjunction with expert systems) turned out to be helpful in forecasting the course of the disease which in turn aided in the ongoing determination of preventative measures at the government level (Mufid et al., 2020). Vargiu and Urru (2013, p. 45) also observe the usage of data collecting tools (with specific emphasis on web scraping technologies) in such fields as: online pricing comparison, weather data monitoring, website change detection, Web research, Web mashup, and Web data integration. Other instances of practical application may be expanded, as the use of technologies and the creation of data processing and acquisition procedures turn out to be of actual benefit in all sorts of scientific sectors as well as in business.

As previously noted, the network is loaded with an inconceivable quantity of data: it can be publicly accessible information, but also data that is accessible

[12] It is worth noting here that Goes (2014) as well as IBM (2018) distinguish only four most important features (pillars) of the basic descriptive big data model ("4V") that web scraping, as a method, has to face: Velocity, Veracity, Volume, and Variety.

to an extremely restricted audience. However, disclosing data to the public, even if it is voluntary, does not lessen information asymmetry, which is proven by empirical research undertaken in numerous sectors of science (Petersen & Plenborg, 2006; Brown & Hillegeist, 2007; Gajewski & Li, 2015). The key problem not just for data analysts, but for entire organizations and other stakeholders as well, is the utilization of the potential of unstructured data, the great majority of which is located in Internet resources. It has been noted that web scraping focuses on converting unstructured data that remains on the web (most typically in HTML format) into structured data that can be transferred to spreadsheets or databases for further examination (Vargiu & Urru, 2013, p. 44; Dewi et al., 2019). It is, thus, congruent with the idea provided by de S Sirisuriya (2015) and other aforementioned definitions. It also illustrates the significant potential of web collecting techniques in scientific and economic application.

Once online scraping data has been gathered, validation is deemed to be a critical stage in the complete gathering process. It is the last phase that needs to be completed shortly before further processing or evaluating the acquired data. Validated data may be understood more effectively and hence greater quality information can be gained (Dewi et al., 2019). Furthermore, validation is seen as an inherent and vital aspect of the complete knowledge management process, which the high-quality information comprises of (Bhatt, 2001; Mach & Owoc, 2001; Stal & Paliwoda-Pękosz, 2017).

Mitchell (2015, p. 10) demonstrates that accessing the Internet simply with the use of a web browser drastically limits the possibilities and spectrum of cognition. He also notes that the browser is just used to show relatively limited material and objects integrated on the website in an orderly, legible, and welcoming fashion for the ordinary e-user. Additionally, he points out that when using a search engine (such as Google), after typing a particular phrase, internet advertisements and relevant search results will be shown because search engines (which incidentally also use bots) only check a particular block on websites, not their entire content and exact values. Internet bots enable us to explore beyond what is visible on a small screen since they can search databases comprising hundreds or even millions of web pages at the same time, as Mitchell (2015, p. 10) points out. The rising tendency of integrating automatic IE technologies, including data gathering tools, is also noted in scientific research (Chang et al., 2006). This survey also reveals that online harvesting techniques are also good at managing vast volumes of semi-structured data, minimizing not just programming but also labeling efforts. It should be emphasized that not all IT system jobs can be done by automated technologies.

2.2. Properties of data acquisition tools

In order to start discussing the qualities of online scraping tools, it is vital to know the basic operating concepts utilized in the technology. The correct application of data extraction techniques is linked to knowledge of HTML in order to properly read the code of websites, as well as the method of data aggregation on the Internet (e.g., DOM) and knowledge of programming languages (such as Python) in order to develop and modify a web scraping tool (Krijnen et al., 2014). Fernández-Villamor et al. (2011) outline three fundamental strategies (levels) that are extensively employed in online scraping:

- *syntactic web scraping* – this technique extracts data from website structures by parsing HTML, CSS, and other typical web designing languages (Krijnen et al., 2014, p. 2);
- *semantic web scraping* – this technique makes it easier to map semantic web source materials with specialized frameworks[13]; however, surprisingly, the failure of the semantic web has made this technique not preferred by most applications (Krijnen et al., 2014, p. 3);
- *computer-vision webpage analyzing* – the technique that comes closest to simulating human browsing and human interpretation of websites; utilizes machine learning, artificial intelligence (AI), and computer vision techniques (Zhou & Mashuq, 2014).

As an extension of the previous principles, it is important describing what *syntax* and *semantics* are, and why these notions are distinct. The *syntax* of a language is its linguistic representation, that is, the systematic assertions of formal guidelines which, coupled with the formation of the implications of these principles, attempt to regulate that language (Carnap, 1934/1937, p. 1). *Syntax* refers to the manner in which symbols construct comprehensible and intelligible statements (including programs) in a particular language (Slonneger, 1995, p. 1). Hereof, in connection with the invoked appellation with Slonneger, it should also be mentioned that this pertains to programming languages used in creating computer programs and hence web scraping tools production as well. The same applies to *semantics*, which specifies the meaning of valid word sequences (strings) that are syntactically specified (Slonneger, 1995, p. 1). Slonneger's definition is also extended strictly to programming languages: *semantics* explains the behavior of a computer when running a certain program written in a specified language.

[13] Examples of these specialized frameworks are presented in Table 2.2.

Table 2.2 displays the sorts of web scraping strategies that are employed in the three forementioned fundamental levels of web scraping proposed by Fernández-Villamor et al. (2011):

Table 2.2. Types of web scraping techniques

syntactic web scraping	semantic web scraping	computer-vision webpage analyzing
• CSS • XPATH • Tress • RegExps • HTML/XHTML	• RDF • OWL • Triple Stores • SPARQL	• Artificial Intelligence (AI) • Machine Learning

Source: Own study based on (Krijnen et al., 2014, p. 3).

On the other hand, Dogucu and Çetinkaya-Rundel (2020) provide a slightly distinct differentiation of web scraping depending on the manner of collecting data:

- extracting data with the use of website's source code with an HTML parser or regular expression matching;
- extracting data with the use of APIs[14] provided by websites (where a given website offers a set of structured HTTP requests[15] that return JSON or XML files[16]).

Employing the API, i.e., an interface that may be utilized to acquire data, made accessible by a particular website, and therefore employing web scraping techniques, is regarded as part of the fundamental data science competencies (Hicks & Irizarry, 2018).

Data content mining tool is a computer software, hence it has its own qualities. In regard to syntax, a program is a text that follows a well-defined set of grammatical rules (Pair, 1990, p. 10; Détienne, 2002, p. 13). It is, thus, compatible with the aforementioned Carnap's (1934/1937, p. 1) and Slonneger's (1995, p. 1) definitions of syntax. Viewed from the standpoint of semantics, a program is an expression of a computation sequence (Pair, 1990, p. 10; Détienne, 2002, p. 13). By uniting these two points of view, a program is a lan-

[14] API is an abbreviation of *Application Programming Interface*.

[15] Such HTTP commands can be, for example, GET or POST – they can be completed, in example, with the use of Apache or Java libraries (Eriksson, 2016, p. 5).

[16] Sometimes they can also be files with comma separated values, which is CSV format, as well as even Microsoft Word files (with DOC/DOCX formats) or PDFs (Mitchell, 2015, pp. 90-91, 114-121).

guage that describes a specific function that a computer (machine) can understand to calculate[17] it (Pair, 1990, p. 10). Wirth (1976) also observes that a program has two components: an algorithm and – incidentally – the structure of data (objects). Web scraping apps, like any other software, might have a user-friendly graphical interface (a so-called GUI) or can be a program that uses the command prompt for its operation. It can either be installed locally on the computer's hard disk or incorporated in the Internet cloud. Almost always we require an Internet connection to employ data extraction tools, since we can only access an external website via WWW services.

Screen scraping tools, like other IT products, feature a life cycle consisting of three phases: generation, execution, and maintenance (Eriksson, 2016, p. 7). Hence, in the generation phase, the web scraping tool is established (created), and in the execution phase, it is in the process of its working (i.e., then it extracts data from the website). Due to the dynamic nature of online applications and continually changing Internet sources, the risk of faults in the operation of the scraper rises over time, which involves the necessity to provide adjustments (Eriksson, 2016, p. 7). Therefore, it also entails continuous maintenance (typically manual) of a particular tool, in order to "secure the robustness of the implementation" (Eriksson, 2016, pp. 7-9).

Ferrara et al. (2014), in their thorough examination, recognized four techniques to designing a data extraction tool: *regular-expressions-based*, *logic--based*, *tree-based,* and *machine-learning-based*. *Regular-expressions-based* approach is most typically used in data mining, which seeks to detect strings of text (or patterns) matching certain criteria. Scrapers constructed with this method employ selectors in a bid to shell out certain tags in HTML code of a semi-structured page, tabular structure or other criteria that are explicitly stated in terms of syntactics (Krijnen et al., 2014, pp. 2-3). Tools developed with *logic-based* approach are based on certain programming languages that were employed to develop the software and do not consider websites as a volatile text but rather as a "semi-structured tree document". The core of this technique is the DOM, which retrieves the required and various nodes from the HTML/XHTML code that specifies the features of the attributes and content of the document. *Tree-based* approach attempts to partition the document tree and, consequently, separate the objects (executed in adjacent portions of the

[17] Pair (1990, p. 10) notes that the word "calculate" should be taken in the broadest sense. "Calculating" can be "printing, drawing graphs, interpreting data transmitted by sensors, giving orders to a robot, consulting a dictionary or a file" or whatever fits that sense.

document, termed data record regions) for subsequent extraction (Zhai & Liu, 2006; Eriksson, 2016, p. 7). The purpose is to detect and identify these sections, and then the algorithm, based on the DOM and graphical representation, segments the page with the aim to find gaps between the records. *Machine-learning* approach, by definition, is training models to independently acquire data and even relevant knowledge contained in the data (Murdoch et al., 2019).

Boronat (2008, pp. 11-13) addresses the typical challenges that may be faced when an analyst wishes to retrieve data from an HTML page using web content mining tools:

- a website with a ‘chaotic’ data structure or a lack of it (then the website usually has an eye-catching appearance which is preferable by the end user),
- a badly structured HTML website file (also with improper use of CSS style sheets that incorrectly format the data) that does not conform to the W3C[18] web writing standards,
- a website with deeply nested data (also with embedded data inside Flash, JavaScript or AJAX scripts), which makes it difficult to obtain.

Additionally, it should be mentioned that the Internet comprises of numerous superfluous components. Such redundant material on webpages (the so-called “noisy information”) can greatly degrade the performance of the data extraction process (Ahmad Sabri et al., 2019). The key is to isolate and divest unnecessary information so that the collected data is as full as feasible in quality (Bartuś et al., 2017; Olszak & Zurada, 2020). Yi et al. (2003) as well as Yi and Liu (2003) suggested an effective taxonomy of noisy information, splitting it into two categories: *local noise* and *global noise*. *Local noise* (also called *intra-web noise*) is all the redundant content which is integrated directly on a specific website (such unneeded graphics, adverts, links, and so forth). *Global noise* (also called *inter-web noise*), on the other hand, is characterized as online sounds with great granularity (typically not smaller than individual webpages). They can be for example duplicated pages (it is unimportant if legal or unlawful), obsolete websites entailed for deletion (but still published in the web) or mirror sites.

[18] The World Wide Web Consortium (W3C) is “an international community where Member organizations, a full-time staff, and the public work together to develop Web standards” (W3C, 2019a). Its mission is “to lead the World Wide Web to its full potential by developing protocols and guidelines that ensure the long-term growth of the Web” (W3C, 2019b).

Addressing these challenges could be viewed as a troublesome undertaking, because the analyst utilizing data extraction techniques has no impact on the structure of the website in terms of technological backdrop. However, when such possible predicaments can be avoided, data collecting becomes substantially easier.

2.3. Use of web scraping in practice

The usage of data extraction techniques, including web scraping, gives various options and an extensive array of possibilities. As with any instrument or method, we need to be mindful of the external limits and operating frameworks. This portion of the study highlights the benefits of employing data content mining (in Subchapter 2.3.1) and focuses on the ethical and legal aspects of using Internet data extraction tools (in Subchapter 2.3.2).

2.3.1. Strengths and opportunities of data content mining

The mere possibility of acquiring data and putting it into a structured format (both through a syntactic and semantic approach) is regarded a benefit of the overall web scraping technique (Krijnen et al., p. 3). Also, a further advantage that can be rendered by web scraping is the modification of the data structure to the demands of a particular user who is interested in investigating the data. It may really be claimed that the data itself is a vital opportunity to be taken with the application of content mining. An important possibility for web harvesting is to merge statistical and IT themes not by creating proof hypotheses but by tackling problems in a precise sense (Dogucu & Çetinkaya-Rundel, 2020, p. 9). Web scraping approaches are not only suited to acquiring data from diverse sources but also facilitate effective interaction with data sets of varied sizes, formats, and forms (Dogucu & Çetinkaya-Rundel, 2020, p. 9). Hence, Horton et al. (2015) recognized that the interaction and processing of big, complicated, and even embroiled data sets that are not contained in rectangular matrices is particularly significant for statistical reasons. Through its flexibility and robustness, web scraping makes it feasible.

As the want to have the most up-to-date data should be an objective for a data scientist, the usage of screen scraping tools is frequently more desirable than the API supplied by the website administrator (Krijnen et al., 2014, p. 3). The API is generally updated less often than the dynamic page content shown to the end user. Due to the fact that web scraping by its definition captures data viewed by the Internet user, functioning like the user himself, it turns out to be a beneficial way of data collecting. The practical usage of data content mining in numerous scientific disciplines has been shown in Subchapter 2.1; the diversity of solutions that have been made available by data extraction technologies are an incontrovertible benefit of the subject as a whole.

Web scraping technologies, which are extensively utilized in every area of academia and business, have many of the indisputable benefits and prospects listed above. The method of gathering data, however, has a large potential jeopardy, which may be classified as ethical and legal concerns, and which may disrupt its overall operation. This threat is considered as complicated and widespread, thus the author of this work has opted to devote a separate section to discussing it. Notwithstanding, proper handling of web harvesting tools or approaches and bearing in mind the vital considerations outlined in Subchapter 2.3.2 may effectively mitigate the hazard, while not unreasonably constraining the analytical flexibility.

2.3.2. Legality and ethical use of web scraping

Web scraping, as a data collecting method, in terms of the legality of its usage, regrettably belongs to the so-called "gray area" (Mitchell, 2013, pp. 8-9; Snell & Menaldo, 2016; Zamora, 2019; Krotov et al., 2020). In actuality, it is mostly permissible to acquire data that is for private use solely, but when the data collected in this method is reprinted, the kind of scraped data plays a significant legal function (Lawson, 2015, p. 2). Improper management of screen scraping might even lead to a potential harm to the hosted website (Zamora, 2019). As the collecting of data from websites is related with increased traffic on them owing to the quantity of questions submitted in a short period, the person responsible for overloading or ruining the website may obtain claims on the basis of the "trespass to chattels" legislation (Merrell, 2002; Dreyer & Stockon, 2013)[19].

Thankfully, there is a wicket that permits us to employ automated data extraction techniques lawfully. Generally, as indicated previously, online harvesting tools are meant to simulate human traffic on the website, therefore

[19] A "trespass to chattels" law is "an intentional interference with another person's lawful possession of a personal property" (FindLaw, 2018). Although the very term "chattel" refers to any personal property, moving or unmoving, it should be noted that "trespass to chattels" law does not necessarily apply to any kind of a real property or any interest in land.

they cannot be directly compared to viruses invading internet resources. The most acceptable technique to receive data from the Internet (in the context of legality) is to utilize the API, (Krijnen et al., 2014, p. 2; Mitchell, 2015, pp. 10–11). Due to the fact that interfaces for getting data are established by website developers and it is they who manage the conditions of the API use, the danger of breaking the website's privacy is low. Often, however, the use of the API supplied by the website, despite the preferences arising from legal difficulties, is inadequate when juxtaposed with the building of an internet bot. Mitchell (2015, p. 11), within this scenario, emphasizes primarily concerns connected to data size constraints, the regularity of sending inquiries for their gathering, as well as the data format (type). He additionally stresses out that the application of the API will not fulfill its role when receiving data from a vast selection of websites that lack a consistent API interface. It also commonly happens that websites do not have an API, especially when data is rare, unique or controversial (Mitchell, 2015, p. 11).

Nevertheless, there is no regulation that directly tackles web scraping itself (Krotov & Silva, 2018, p. 3). Hence, it is beneficial to obey the conditions of use and copyright of the website from where the data originated (Mitchell, 2013, p. 8). At the moment, the use of online scraping tools is rigorously restricted by higher-level legal acts and theories, such as those dealing to "copyright infringement", "illegal access and use of data", "breach of contract", the Computer Fraud and Abuse Act (CFAA)[20], and the aforementioned "trespass to chattels" law (Dreyer & Stockton, 2013; Snell & Menaldo, 2016; Goldfein & Keyte, 2017). Various countries or regions throughout the world might not have a consistent designation for legislative acts that deal with the aforementioned issues, only the CFAA is a statute that applies in the United States territory. In European Union, General Data Protection Regulation (GDPR) pertains merely to safeguard individuals' privacy, hence it may be relevant only in terms of web scraping usage, when the scraped information comprises of personal data and thus breaches them (Hadasik, 2019; Davies, 2020).

Krotov et al. (2020) explicitly enlarged violative examples connected to the usage of online scraping techniques. First and foremost of all, in disputable

[20] The Computer Fraud and Abuse Act (CFAA) is a United States cybersecurity bill which was enacted in 1986 as an amendment to an existing computer fraud law (18 U.S.C. § 1030). It has also been included in the Comprehensive Crime Control Act of 1984 (adapted from: Conrad et al., 2014; Andress & Winterfeld, 2011). Due to the fact that the CFAA is a federal law, i.e., it is valid throughout the United States, comparable state laws have also been created based on it.

cases, the courts tend to analyze whether the "Terms of Use" or the "Terms of Service" rules have not been explicitly abused. These documents might specifically utilize the statement that the use of data extraction tools is outlawed. Thereby, it is examined whether access to the website is "unauthorized". At any point, the owner of the website that controls the data may withdraw permission to data gathering by sending a cease-and-desist-letter[21] to the data extractor. From this moment on, any usage of web content mining tools will be regarded "unauthorized" and may be subject to lawsuit. Additionally, when, aside from illegal access, the "Terms of Use" are damaged in an overt and purposeful manner, and in particular when the data controller suffers damage, this may be regarded a "breach of contract" violation (Dreyer & Stockon, 2013). However, it is under question whether the website owner may successfully restrict access to the site for content mining tools by expressing it expressly in "Terms of Use" (Krotov et al., 2020). The extraction and republication of expressly copyrighted or proprietary material might lead to "copyright infringement" violation (Dreyer & Stockon, 2013). However, it should be recognized that, for example, concepts are not governed by copyright law but only their specific form or expression (Krotov & Silva, 2018). For example, the production of summaries based on copyrighted data, and the use of such material under the "fair use" principle[22] is permitted (Krotov & Silva, 2018).

The greatest hurdle to the release of a clear verdict are objections from the group of the so-called "kitchen-sink"[23], which include the aforementioned violations (Zamora, 2019). These sorts of infringements linked to the usage of screen scraping technologies are detailed in depth and examined on particular cases in the paper by Snell and Care (2013). It should be acknowledged that the local legislation of the nation or region, where an infraction has occurred, may differ. International law may also be utilized in many circumstances, for example, if the infringement or corruption has been perpetrated in one country and the servers of the breached website are situated in another.

[21] A cease-and-desist-letter is "a cautionary letter sent to an alleged wrongdoer describing the alleged misconduct and demanding that the alleged misconduct be stopped" (Legal Information Institute). The intention of sending such letters is often to stop alleged or actual infringement of intellectual property rights, such as copyrights, trademarks, and patents.

[22] "Fair use" principle is explained as "any copying of copyrighted material done for a limited and 'transformative' purpose, such as to comment upon, criticize, or parody a copyrighted work" (Stanford University Libraries).

[23] "Kitchen-sink" in this case is explained as "being or made up of a hodgepodge of disparate elements or ingredients" (Merriam-Webster).

Another challenge to consistent verdicts is the fact that there are numerous aims throughout the range of societal approval of the business model with the use of internet scraping (Zamora, 2019)[24]. For example, the world's most prevalent indexing bot in the world, Googlebot, serves one of the important functions for users of the Google search engine: without this instrument, Google would not be able to index, sort or rank search results (Google Developers, 2020). On the other side, there is another plainly destructive extreme: there are imposter web bots meant to purposely damage certain targeted websites by launching DDoS assaults[25] (Zamora, 2019).

There are circumstances where the employment of data extraction techniques is inherently questionable and even harmful because to sensitive legal areas, irrespective of how data is handled or used. Searching for sensitive, personal or valuable data (e.g., searching address or contact info) is sometimes forbidden, which generally derives directly from the conditions of use of the website (Mitchell, 2013, p. 9). For example, the social networking site Twitter (now: X) specifically specifies that the use of web scraping tools for scanning real tweet data is banned, unless mutual approval for one-off usage has been granted[26]. Mitchell (2013, p. 9) notes that if the conditions of use of the website do not specifically restrict the use of robotic data collecting tools, it does not matter whether the website (and hence the information on it) is accessed using a browser or an automated script. It should be highlighted that local and international infringement or copyright laws may also apply and be relevant in the case of a disagreement.

Snell and Menaldo (2016), on the foundation of situations covered in their study, advise the following verification efforts (both by website owners and analysts employing data extraction technologies) to optimize the protection against potential legal conflicts:

a. checking the language clarity of the terms of use or service;
b. checking the enforceability of the terms of use;
c. providing the list of used technological tools to limit unwanted activity;

[24] Based on: Maheedharan (2016).

[25] DDoS (abbreviation of *distributed denial-of-service*) attack is a "malicious attempt to disrupt the normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic" (CloudFlare).

[26] This results directly from the Twitter "Terms of Service". Section 8 of the Twitter's Terms of Service ("Restrictions on Content and Use of the Services") emphatically states that: "crawling the Services is permissible if done in accordance with the provisions of the *robots.txt* file, however, scraping the Services without the prior consent of Twitter is expressly prohibited" (Twitter, 2012).

d. checking whether access to the website is protected such that a claim under different legal acts protecting privacy may be alleged;
e. checking whether data on the website content is protected by copyright;
f. checking whether the website owner will license or authorize uses of content.

As law and ethics are independent, yet complimentary, scientific domains (Mingers & Walsham, 2010), the author of this work determined that the ethical implications of online scraping will also be covered at this point. As science offers ethics from numerous angles, the definition of ethics by Donaldsson et al. (1983) was selected for the purposes of this work, which indicates that ethics is "the study of whatever is right and good for humans". Before commencing the process of a screen scraping tool construction, it is vital to consider if the tool and its usage will be both legal and ethical, which was also highlighted by Krotov et al. (2020).

The consequences underlying reckless utilization of web harvesting tools may include prejudice or bias. Inadequately handled data gathered by online scraping may lead to skewed actions and enhance the present or even develop new prejudices (Krotov et al., 2020). By irresponsible use of this data, new kinds of socio-financial discrimination or incorrect profiling of persons based on biases can be generated (Wigan & Clarke, 2013; Someh et al., 2019). Enterprises can particularly target services or products on the basis of prior behavior and, thus, may charge various pricing for their goods, based on attributes of an individual or group (Newell & Marabelli, 2015)[27].

Zamora (2019), after analyzing various situations reported in depth, has given four key requirements that should be followed to prevent screen scraping from needless disagreements:

- an automated tool that obtains data from the web should behave like a 'good e-citizen' that is not meant to overload the website;
- the copied data should be public, obtained in a way that does not circumvent applied security protocols (such as cracking passwords);
- the obtained data should be initially factual, which does not infringe anyone's rights (including copyrights);
- the data was acquired to create a transformation only, not to capture market shares by siphoning users or to create a substantial competitive product.

[27] For example: young drivers have to pay more for car insurance, which is solely based on the age of the individual. Based on: Krotov et al. (2020; cited by Newell & Marabelli, 2015) and Kubiczek (2019b).

Furthermore, Zamora (2019) advised to distinguish between actors driven by good and negative intentions. She also stated that limits are needed to avoid mimicking web scrapers from being utilized intrusively. Thus, such limits can successfully safeguard good intended actors by giving them priority in data collecting and therefore in having the opportunity to develop new information.

Krotov et al. (2020) constructed a good framework with questions regarding the online scraping process that encompass ethical and legal considerations. Answers to these questions assist to establish whether the data gathering procedure is appropriate in terms of compliance with law and ethical requirements. The author of this work decided to adapt these questions into statements that clearly explain the circumstances of non-compliance with the ethical and legal norms of the complete data content mining process. Here is what follows:

- web scraping or web crawling is expressly prohibited under "terms of use";
- data posted on the website is copyrighted;
- the use of web scraping techniques may cause a defect or harm the website;
- the website administrator has restricted (or disabled) access to it or to the data contained therein (e.g., by sending a "cease and desist letter" or by excluding the user's IP address from the access list);
- the instructions in the robots.txt protocol prevent or significantly block the use of data extraction tools;
- the obtained data will violate the individual privacy and rights of the entity from which the data is acquired or will be in antagonism to the antidiscrimination order;
- the data obtained may contain confidential information about the organizations associated with the website;
- the purpose of obtaining data is to diminish the value of the service provided by the website;
- the quality of the data obtained from the website can lead to decisions based on inadequate information.

If any of the aforementioned requirements is satisfied, it signifies that in a specific circumstance this particular procedure of acquiring data from a certain website is not entirely consistent with legal and ethical norms. As a result, the data gathering process is delayed and so data gathered in this process may lose its usefulness, becoming redundant.

***

Web scraping is an effective means of acquiring data that is frequently utilized in the scientific and economic areas. It successfully strives to eliminate information asymmetry and offers various opportunities, the potential of which may be leveraged not only by a data analyst but also by an average user. Its usage, however, is connected to the knowledge of basic IT concerns, like HTML instructions or the ideas of semantics and syntax. Despite its clear advantages, there are ethical and legal considerations that may limit the usage of data extraction techniques. However, when a user of online scraping tools is diligent while using these wisely and bearing in mind the potential ethical or legal dangers, the extraction of data from the web will not only be viable, but above all will provide a major additional value to the study. Besides, it will not hinder mutual rights or freedoms either. The following Chapter 3 is a practical portion of the thesis in which the analysis of the OTOMOTO advertising portal, which is the topic of data collecting, will be undertaken. This Chapter also describes the process of designing a web scraping tool, as well as its actual application and validation in the context of appropriate legal requirements.

Chapter 3. Designing a web scraping tool

Having explained the concept of web scraping in the context of information mining, its ethical and legal aspects, and highlighting the economic idea of reducing information asymmetry, the following Chapter strives to offer a practical approach to the construction of a data extraction tool for an automotive advertising portal. It will utilize the largest website with automobile classifieds in Poland, i.e. OTOMOTO as an example. The webpage extractor tool was built utilizing the Python programming language. The purpose of the practical component of this work is not only to construct a specialized medium for receiving data from a defined (yet dynamic) list of websites, but also to preserve this data in a pleasant manner for subsequent analysis. A trial analysis of the acquired data will also be given, accompanied by proper validation in order to achieve the aforementioned "significant value hidden in data" (Olszak & Zurada, 2020).

The description and structure of the OTOMOTO portal will be presented in this Chapter (in Subchapter 3.1), as will the scraping web program, along with the specification of the newly-created functions and variables, libraries used, and scraping strategies or approaches implemented. Apart from the ways of getting and presenting the acquired data, the application in question also contains actions targeted at storing them to a Microsoft Excel spreadsheet format (i.e., an XLSX file). Then, on the basis of the produced file, a demonstration data analysis will be carried out, following previous confirmation. The ethical elements of the obtained data will also be checked.

3.1. OTOMOTO – the largest Polish automotive classifieds portal

OTOMOTO is an automobile platform belonging to the Dutch OLX company, which enables us to sell and search for vehicles in Poland. With its support, on the one hand, sellers (private and corporations) may publish motoring adverts, supplementing the price, car qualities, and other factors, and on the other: buyers can search for these vehicles using filters based on data added by sellers. OTOMOTO is both the largest website with automobile announcements in Poland[28] and the most visited one (WirtualneMedia.pl, 2020)[29]. The structure of the discussed portal in terms of the way pages are placed, HTML code properties, and the link setups is presented in Subchapter 3.1.1.

3.1.1. Structure of the portal

The homepage of this portal, which can be accessed through the https://otomoto.pl address, in addition to the published advertisement, features a streamlined form for finding automobiles, as well as adverts that have been highlighted by the seller for an extra price. It is also possible to use the search engine for parts, bikes, vans, trucks, buses, etc., utilizing the tabs above the streamlined search form. With the portal's help, we can also access the user's panel (the link is in the top right corner of the "sticky bar") and travel to individual subpages of the website through links in the bottom. A screenshot of the OTOMOTO homepage[30] is presented in Figure 3.1 (see p. 50).

The pages of the OTOMOTO advertising portal can be, in simplification, divided into two categories: a collective list of classifieds and a page of a specific ad. It is worth mentioning that this simplified division does not include strictly informational subpages, such as contact pages, those related to the regulations or the press office vortal.

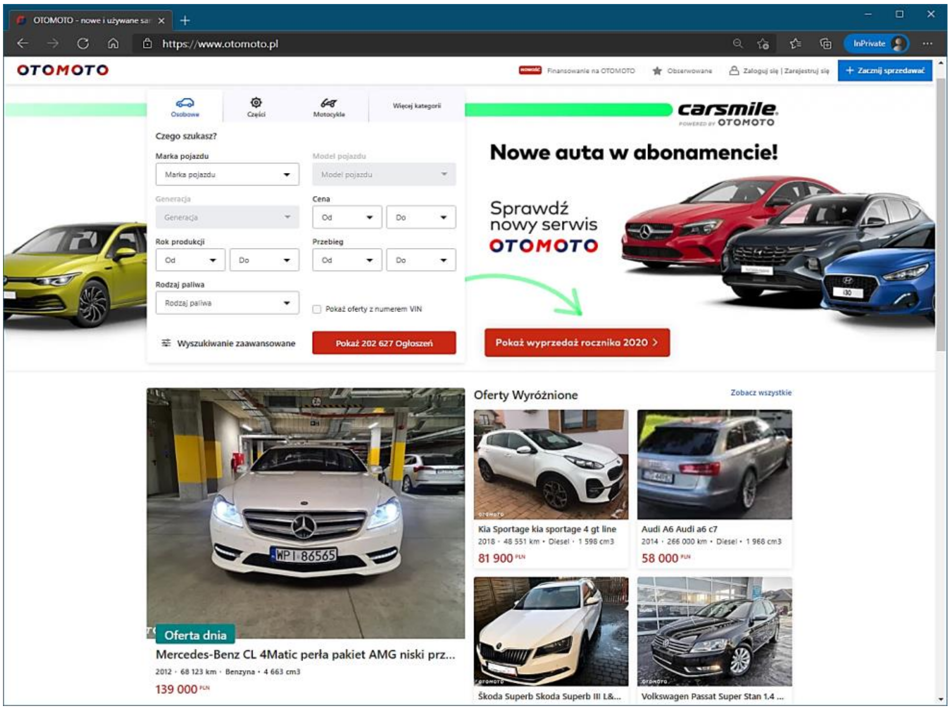
The motoring listings page consists of two main elements: the filters section at the top of the page, and the list of individual ads below. Tiles with links to ads, in addition to the brand and model name, contain a photo of the vehicle, price, location, and several individual technical data of the vehicle. Using the filter section, it is possible to select individual ads, dividing them according to the year of production, price, equipment, fuel type, engine, location, and many other filters. Brand new car (from dealers) ads as well as classifieds with aftermarket cars can also be selected. There is also a sorting tool with which

[28] Based on the number of advertisements in the "passenger cars" category. The number of ads in the three largest portals of this type in Poland as of October 22, 2023 was: OTOMOTO – 231,109 advertisements, Sprzedajemy.pl (automotive section) – 64,898 advertisements, gratka.pl/motoryzacja – 90,045 advertisements. As OTOMOTO is part of the OLX group, the OLX.pl advertising portal was not taken into account.

[29] In June 2020, the OTOMOTO website was visited by 6.8 million users, i.e., 25 percent of all Internet users in Poland. They accounted for 345.43 million views on the website, each an average of 50.8 (based on the Gemius/PBI survey prepared by WirtualneMedia.pl). For comparison, the second largest automotive classifieds website in Poland – Sprzedajemy.pl (automotive section) – recorded in the same period only about 1.1 million users with less than 8 million views (based on the same study).

[30] All the screenshots of the OTOMOTO website were used in the work under the right of quote. Further explanation is included in the Appendix, along with Figure A.1.

there is a possibility to order ads by date or price. On the page with the list of ads, there are also the so-called “breadcrumbs”, which indicate the user’s location on a website. The issue of the link structure of a page with filtered advertisements will be discussed later in this Chapter. The example page with the listed classified ads is presented in Figure 3.2 (see p. 51).

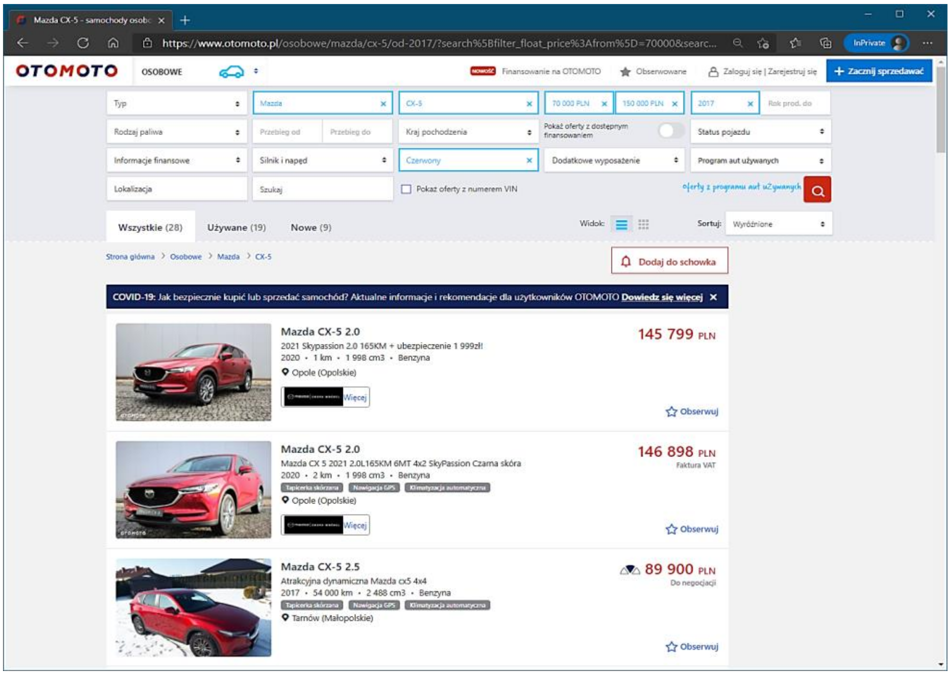


OTOMOTO - nowe i używane sa
https://www.otomoto.pl
OTOMOTO
Finansowanie na OTOMOTO
Obserwowane
Zaloguj się | Zarejestruj się
+ Zacznij sprzedawać
Osobowe
Części
Motocykle
Więcej kategorii
Czego szukasz?
Marka pojazdu
Model pojazdu
Generacja
Cena
Od
Do
Rok produkcji
Przebieg
Rodzaj paliwa
Pokaż oferty z numerem VIN
Wyszukiwanie zaawansowane
Pokaż 202 627 Ogłoszeń
carsmile
POWERED BY OTOMOTO
Nowe auta w abonamencie!
Sprawdź nowy serwis OTOMOTO
Pokaż wyprzedaż rocznika 2020 >
Oferty Wyróżnione
Zobacz wszystkie
Oferta dnia
Mercedes-Benz CL 4Matic perła pakiet AMG niski prz...
2012 · 68 123 km · Benzyna · 4 663 cm3
139 000 PLN
Kia Sportage kia sportage 4 gt line
2018 · 48 551 km · Diesel · 1 598 cm3
81 900 PLN
Audi A6 Audi a6 c7
2014 · 266 000 km · Diesel · 1 968 cm3
58 000 PLN
Škoda Superb Skoda Superb III L&...
Volkswagen Passat Super Stan 1.4 ...

Figure 3.1. Homepage of OTOMOTO

Source: https://otomoto.pl (accessed March 1, 2021).

The structure of a page of a specific vehicle specimen that is an advertisement also consists of several basic elements. The most exposed is the photo gallery of the car, but directly next to it (on the right), there is the make and model of the vehicle, the price, as well as links to contact the seller, or the location provided. It is worth noting that, like the list of all ads, the classifieds page also has breadcrumbs which are above the photo gallery. Below the photo gallery, in the central part of the page, there are advertisement parameters, such as the date of addition, ID or type of offer (from a private person/company), as well as technical specifications of the vehicle, such as engine, power, types of fuel, gearbox, vehicle, year of production, etc. These parameters are used not only to describe the advertisement itself but also in the previously discussed filters to narrow down the search results. A screenshot with an example of an advertisement page on the OTOMOTO portal is shown in Figure 3.3 (see p. 51).



Mazda CX-5 - samochody osob

https://www.otomoto.pl/osobowe/mazda/cx-5/od-2017/?search%5Bfilter_float_price%3Afrom%5D=70000&searc...

OTOMOTO OSOBOWE

Finansowanie na OTOMOTO | Obserwowane | Zaloguj się | Zarejestruj się | Zacznij sprzedawać

Typ	Mazda	CX-5	70 000 PLN	150 000 PLN	2017	Rok prod. do
Rodzaj paliwa	Przebieg od	Przebieg do	Kraj pochodzenia	Pokaż oferty z dostępnym finansowaniem		Status pojazdu
Informacje finansowe	Silnik i napęd		Czerwony	Dodatkowe wyposażenie		Program aut używanych
Lokalizacja	Szukaj		Pokaż oferty z numerem VIN		oferty z programu aut używanych	

Wszystkie (28) | Używane (19) | Nowe (9) | Widok: | Sortuj: Wyróżnione

Strona główna > Osobowe > Mazda > CX-5

Dodaj do schowka

COVID-19: Jak bezpiecznie kupić lub sprzedać samochód? Aktualne informacje i rekomendacje dla użytkowników OTOMOTO **Dowiedz się więcej**

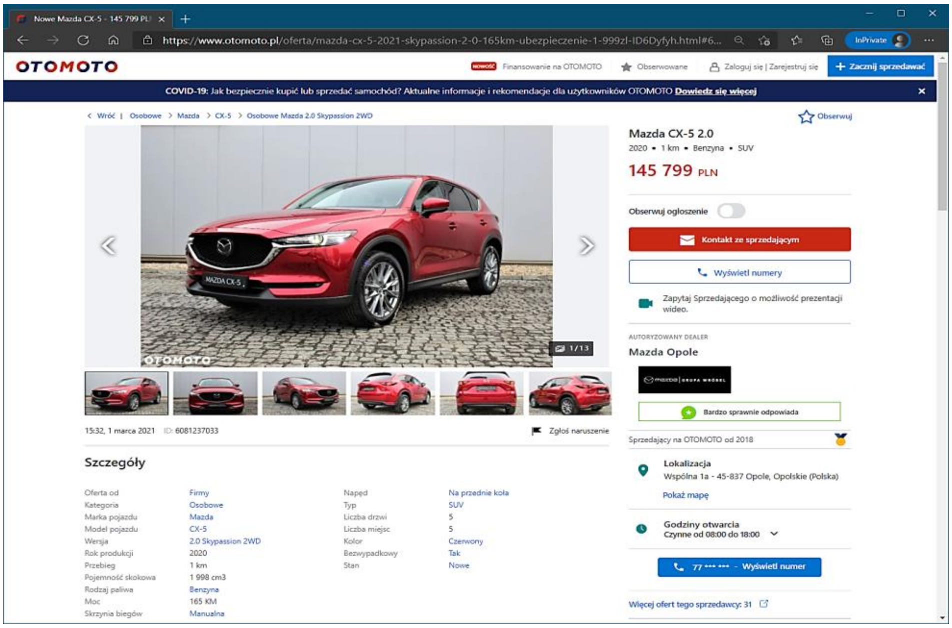
Mazda CX-5 2.0 — 145 799 PLN
2021 Skypassion 2.0 165KM + ubezpieczenie 1 999zł!
2020 • 1 km • 1 998 cm3 • Benzyna
Opole (Opolskie)
Więcej | Obserwuj

Mazda CX-5 2.0 — 146 898 PLN, Faktura VAT
Mazda CX 5 2021 2.0L165KM 6MT 4x2 SkyPassion Czarna skóra
2020 • 2 km • 1 998 cm3 • Benzyna
Tapicerka skórzana | Nawigacja GPS | Klimatyzacja automatyczna
Opole (Opolskie)
Więcej | Obserwuj

Mazda CX-5 2.5 — 89 900 PLN, Do negocjacji
Atrakcyjna dynamiczna Mazda cx5 4x4
2017 • 54 000 km • 2 488 cm3 • Benzyna
Tapicerka skórzana | Nawigacja GPS | Klimatyzacja automatyczna
Tarnów (Małopolskie)
Obserwuj

Figure 3.2. Page with the list of advertisements on the OTOMOTO portal with applied filters

Source: https://otomoto.pl (accessed March 1, 2021).



Nowe Mazda CX-5 - 145 799 PL

https://www.otomoto.pl/oferta/mazda-cx-5-2021-skypassion-2-0-165km-ubezpieczenie-1-999zl-ID6Dyfyh.html#6...

OTOMOTO

Finansowanie na OTOMOTO | Obserwowane | Zaloguj się | Zarejestruj się | Zacznij sprzedawać

COVID-19: Jak bezpiecznie kupić lub sprzedać samochód? Aktualne informacje i rekomendacje dla użytkowników OTOMOTO **Dowiedz się więcej**

< Wróć | Osobowe > Mazda > CX-5 > Osobowe Mazda 2.0 Skypassion 2WD

Obserwuj

Mazda CX-5 2.0
2020 • 1 km • Benzyna • SUV
145 799 PLN

Obserwuj ogłoszenie

Kontakt ze sprzedającym

Wyświetl numery

Zapytaj Sprzedającego o możliwość prezentacji wideo.

AUTORYZOWANY DEALER
Mazda Opole

Bardzo sprawnie odpowiada

Sprzedający na OTOMOTO od 2018

Lokalizacja
Wspólna 1a - 45-837 Opole, Opolskie (Polska)
Pokaż mapę

Godziny otwarcia
Czynne od 08:00 do 18:00

77 *** *** - Wyświetl numer

Więcej ofert tego sprzedawcy: 31

1/13

15:32, 1 marca 2021 ID: 6081237033 Zgłoś naruszenie

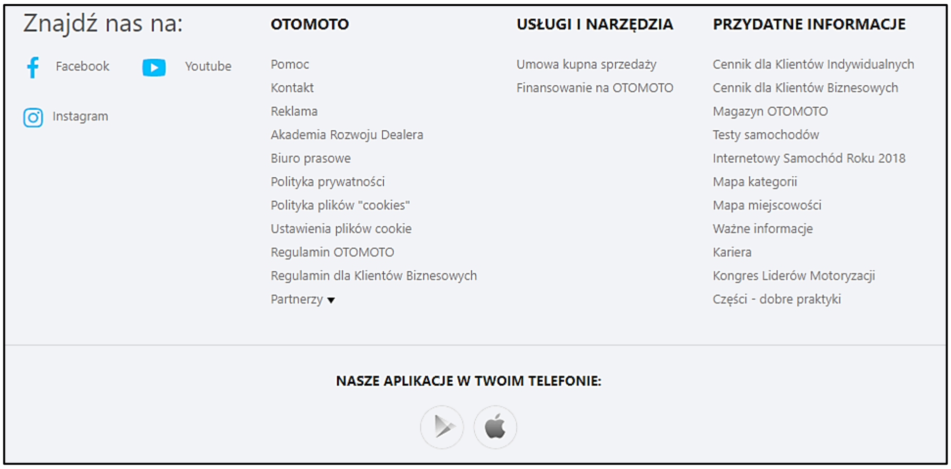
Szczegóły

Oferta od	Firmy	Napęd	Na przednie koła
Kategoria	Osobowe	Typ	SUV
Marka pojazdu	Mazda	Liczba drzwi	5
Model pojazdu	CX-5	Liczba miejsc	5
Wersja	2.0 Skypassion 2WD	Kolor	Czerwony
Rok produkcji	2020	Bezwypadkowy	Tak
Przebieg	1 km	Stan	Nowe
Pojemność skokowa	1 998 cm3		
Rodzaj paliwa	Benzyna		
Moc	165 KM		
Skrzynia biegów	Manualna		

Figure 3.3. Page of a specific advertisement on the OTOMOTO portal

Source: https://otomoto.pl (accessed March 1, 2021).

In the OTOMOTO website's footer, there are links to the help pages, contact page, press office, as well as hyperlinks to the regulations or privacy policy (along with the cookie policy). These pages are not the direct subject of the work but will prove useful for the verification of the ethical and legal aspects of data extraction from this portal, which will take place later in the following Subchapter 3.1.2. The footer with the links is shown in the screenshot below (Figure 3.4).


Znajdź nas na:

Facebook

Youtube

Instagram

OTOMOTO

Pomoc
Kontakt
Reklama
Akademia Rozwoju Dealera
Biuro prasowe
Polityka prywatności
Polityka plików "cookies"
Ustawienia plików cookie
Regulamin OTOMOTO
Regulamin dla Klientów Biznesowych
Partnerzy ▾

USŁUGI I NARZĘDZIA

Umowa kupna sprzedaży
Finansowanie na OTOMOTO

PRZYDATNE INFORMACJE

Cennik dla Klientów Indywidualnych
Cennik dla Klientów Biznesowych
Magazyn OTOMOTO
Testy samochodów
Internetowy Samochód Roku 2018
Mapa kategorii
Mapa miejscowości
Ważne informacje
Kariera
Kongres Liderów Motoryzacji
Części - dobre praktyki

NASZE APLIKACJE W TWOIM TELEFONIE:

Figure 3.4. Footer of the OTOMOTO portal

Source: https://otomoto.pl (accessed March 1, 2021).

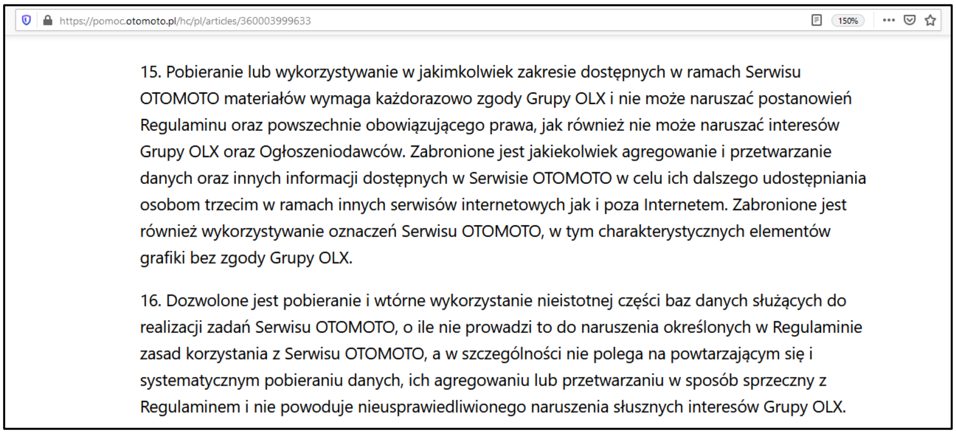
Subchapter 3.1.2 presents a short analysis of the regulations of the OTOMOTO portal in the context of the use and processing of data on this website. This analysis is the starting point for designing a web scraping tool and is needed for the final validation of the use of the designed tool.

3.1.2. A quick look at the OTOMOTO regulations

In order to maintain the ethical issues that are pointed out by Krotov and Silva (2018) and Mitchell (2015, p. 251-264), the regulations of the website, from which obtaining data occurs, should also be analyzed. The analysis of the regulations of the OTOMOTO portal as a whole is, however, not the subject of this work, but it should be checked if they contain clauses limiting or even prohibiting automated data collection by scraping robots. As Jaszewski (2018, p. 10) also pointed out, in the legal aspect, it is also important to check the compliance of the scraping program with the website regulations in accordance with the Polish Act on the protection of databases (of July 27, 2001). On the one hand, point 1 of Article 8 indicates that processing of the database for non-commercial research and scientific purposes is allowed, while point 2 of the same article says that “repeated and systematic data extraction”, which causes “an infringement of the legitimate interests of the [database] manufacturer” is not allowed (even if used for non-commercial purposes). Therefore, Jaszewski (2018, p. 10) strongly recommends to follow the provisions of the regulations of the specific website from which the data is collected.

The most important elements of the OTOMOTO regulations from the point of view of the use of web scraping tools are points 15 and 16 of Article 3, which are presented in Figure 3.5 (see p. 54). They state that it is not possible to use automated tools for “repeated and systematic” data collection and its subsequent aggregation. The regulations must be followed; however, in Article 16, there is a slight gateway to the use of “an insignificant part of databases” but in accordance with the regulations. In conjunction with Article 8 point 1 of the Act, which allows the use of databases for research and personal use, there is a certain possibility to use the database of the OTOMOTO portal. As the design of the web scraping tool serves only the purpose of downloading the results from the first page of ads only once (i.e., 33 ads out of the total number of 210,213 ads – as of March 17, 2021, which is 0.016% of the entire database), which may prove that only “an irrelevant part of the database” is downloaded, it is permissible, according to the regulations, to use this part for private or scientific purposes. However, in order to prevent possible violations of the legitimate interests of the OLX/OTOMOTO group, in accordance with the OTOMOTO website regulations, which state that it is not possible to forward these results to third parties, also for scientific purposes, it is not possible to present the operation of this program in this scientific work. Nevertheless, the description of the program’s operation is presented in a different way, described in Subchapter 3.3.7, which at the same time does not infringe upon anyone’s interests and is fully legal.

Knowing the structure of the website from which we want to obtain data, we can proceed to the process of designing a customized scraping web tool. However, wanting to start the programming process implies a necessity to perform several preliminary steps as a preparation for the whole project. These stages are presented and described in Subchapter 3.2.


https://pomoc.otomoto.pl/hc/pl/articles/360003999633

15. Pobieranie lub wykorzystywanie w jakimkolwiek zakresie dostępnych w ramach Serwisu OTOMOTO materiałów wymaga każdorazowo zgody Grupy OLX i nie może naruszać postanowień Regulaminu oraz powszechnie obowiązującego prawa, jak również nie może naruszać interesów Grupy OLX oraz Ogłoszeniodawców. Zabronione jest jakiekolwiek agregowanie i przetwarzanie danych oraz innych informacji dostępnych w Serwisie OTOMOTO w celu ich dalszego udostępniania osobom trzecim w ramach innych serwisów internetowych jak i poza Internetem. Zabronione jest również wykorzystywanie oznaczeń Serwisu OTOMOTO, w tym charakterystycznych elementów grafiki bez zgody Grupy OLX.

16. Dozwolone jest pobieranie i wtórne wykorzystanie nieistotnej części baz danych służących do realizacji zadań Serwisu OTOMOTO, o ile nie prowadzi to do naruszenia określonych w Regulaminie zasad korzystania z Serwisu OTOMOTO, a w szczególności nie polega na powtarzającym się i systematycznym pobieraniu danych, ich agregowaniu lub przetwarzaniu w sposób sprzeczny z Regulaminem i nie powoduje nieusprawiedliwionego naruszenia słusznych interesów Grupy OLX.

Figure 3.5. An extract from the OTOMOTO regulations, which specifies the use and processing of data on the website

Source: Centrum Pomocy OTOMOTO (2021).

3.2. Preparation for a web scraping tool project

Before proceeding to the proper design of a web harvesting tool (i.e., the programming part), it is important to prepare the environment in which the project will be created, with the prior creation of an algorithm of operation. Subchapter 3.2.1 focuses on the conceptual work that results in the created algorithm, visualized with a process diagram. Subchapter 3.2.2 describes the technical preparation for working on a programming project of a web scraping tool.

3.2.1. Conceptual work. Description of the algorithm's operation. Process diagram

Before approaching programming work in the strict sense, it is necessary (in accordance with the Project Implementation Profile (PIP)) to carry out an implementation plan, including designing an algorithm for the operation of the program itself (Pinto & Prescott, 1988). Therefore, the first step of this phase is a descriptive presentation of the operating steps of the program for data extraction from the OTOMOTO portal, which is finished with a summarized process diagram.

The first step of the program's operation is a welcome message indicating its correct initiation. It is also presented here that the user should follow the commands issued by the script, so that the program can function properly. The first command issued by the script is a general question as to whether the user would like to narrow down the search results to a car brand (in other words: apply a brand filter). As this is a general question, it can be answered “yes” or “no”. In order to simplify communication with the user, the program asks us to enter one letter: “y” (which is equivalent to “yes”) and “n” (which is equivalent to “no”):

- If the user answers “yes”, the program first downloads the list of brands available on the OTOMOTO website, then asks the user to enter the brand, and if it is available, it goes to the next step after saving the brand selection to a variable. If the brand was entered incorrectly (i.e., it is not in the downloaded brand list), the program is interrupted, informing the user about it with an appropriate prompt.
- If the user answers “no”, the program leaves the above-mentioned actions and goes to the next step.
- For both “yes” and “no” answers, the user’s decision is saved to an auxiliary Boolean variable, which will be useful in further stages of the algorithm’s execution.

When the user wants to narrow down the search results according to the vehicle make, he can also select a specific model. The narrowing steps are analogous to those for the brand case. It should be noted, however, that a list of models for a specific vehicle brand is downloaded, which is created dynamically based on the car make selected by the user. If the user does not wish to narrow down the vehicle brand, the stage is skipped.

The next step is to use additional filters: narrowing down the price and the year of production. This stage works irrespective of the user’s decision to narrow down the car make or model. First, the program asks the user about the desire to limit the price, and then asks about the year of production[31]. As in the previous steps, the program asks the user if he wants to narrow down the price (or year of manufacture) and writes his answer to an auxiliary Boolean variable. Again, in this step, the program simplifies communication with the user by narrowing down the answer to “y” or “n”:

- If the user answers “yes”, the program asks the user for a “from” value and then for a “to” value of the price range. The user can use just a one-way narrowing down – if the user wants to do so, he can leave the answer blank, and then a default value[32] will be assigned.

[31] It should be mentioned that the sequence of these steps can be reversed, referring to the conceptual stage (i.e., asking about narrowing down the year of production before the price).

[32] The default values for the price range are: “from” – 0 PLN, “to” – 9,999,999 PLN. The default values for the production year are: “from” – 1900, “to” – *current year*, dynamically created.

- If the user answers “no”, the program leaves the above-mentioned actions and goes to the next step.
- Similarly to the previous stages, for both “yes” and “no” answers, the user’s decision is saved to an auxiliary Boolean variable, which will be useful in further stages of the algorithm’s execution.

The next step, independent of the answers to the questions from the previous stages, is the question about the desire to narrow down the year of production of the car. It is performed analogously to the step filtering the price range. As in the previous steps, the user’s answer to this question is written to an auxiliary Boolean variable.

Completing the above stages, which can be included in the information gathering phase, leads the user to the actual scraping phase. At this point, a link to the list of advertisements meeting the criteria imposed by the program user is generated. The link is generated using the auxiliary binary variables and thereupon saved values of the proper variables (i.e., concerning the names of the make and model of the vehicle, as well as the values of price and year of production ranges).

As it is a scraping program, this is where the imitation of the behavior of a real website visitor begins. The program enters the page with the list of offers (using the link generated from the previous step), and then retrieves links to the appropriate vehicle offers from it, which are consistent with the previously specified narrowing parameters. The program only examines the front page of the offers, which is in line with the ethical principles of limiting the use of the target server (Mitchell, 2015, p. 44, 255). The program works as long as all data from the assumed offer attributes is extracted and as long as all offer pages to which the created links direct are analyzed and scraped. Then the program starts collecting data from the linked pages. The data is taken from a predetermined scheme by the developer who sets the attributes of the offer[33]. Data is searched for on the website according to them and assigned to the variable of a specific variable and a specific ad. This creates a two-dimensional matrix with attributes as columns and specific models in rows. The distinguishing feature (key) of each offer is the offer ID (set by OTOMOTO) because it stays unchangeable and is different for each advertisement.

[33] It was assumed that the user would want to narrow down the attributes (the list of variables) needed for him only at the stage of analyzing the data that has already been collected. This way, when the user wants to perform several analyses of the same model list on the basis of a different variable list, he will not have to restart the program.

The final phase of the program's operation is to display all matrix results in a user-friendly way in the Python shell, and then to export the created matrix to a Microsoft Excel (XLSX) file. In order to facilitate the further analysis already taking place in Microsoft Excel, it was considered more appropriate to add the results of the data extraction operation to a new sheet of the same file rather than creating a new separate file. After the correct data export to the spreadsheet, the program finishes its operation.

Having described the functioning of the algorithm, its operation should also be illustrated. A process diagram, which has been presented in Figure 3.6, may be considered a suitable measure for this purpose. The diagram, which was constructed using the *draw.io* tool[34], shows a simplified operation of the program, where the individual elements correspond to parts of the previously introduced description as well as the actual application.

The shapes of individual elements in the process diagram, which were introduced in the aforementioned Figure 3.6 (see pp. 58-60), have been established in accordance with the accepted standards in the construction of UML diagrams (Miles & Hamilton, 2006). It was decided to use color highlights that will group elements of the algorithm with a similar origin. The following assumptions in terms of color distinctions were made:

- green/red – the beginning and the end of the algorithm's operation,
- orange – conditional elements:
  - light orange – dependent on user's decision (preference),
  - dark orange – independent on user's decision (preference);
- turquoise – setting values,
- purple – saving to variables,
- blue – activities on an external site,
- light blue – input,
- lime – connectors,
- dark grey – operations on matrices,
- light grey – displaying values.

[34] The tool is available for free at https://app.diagrams.net.

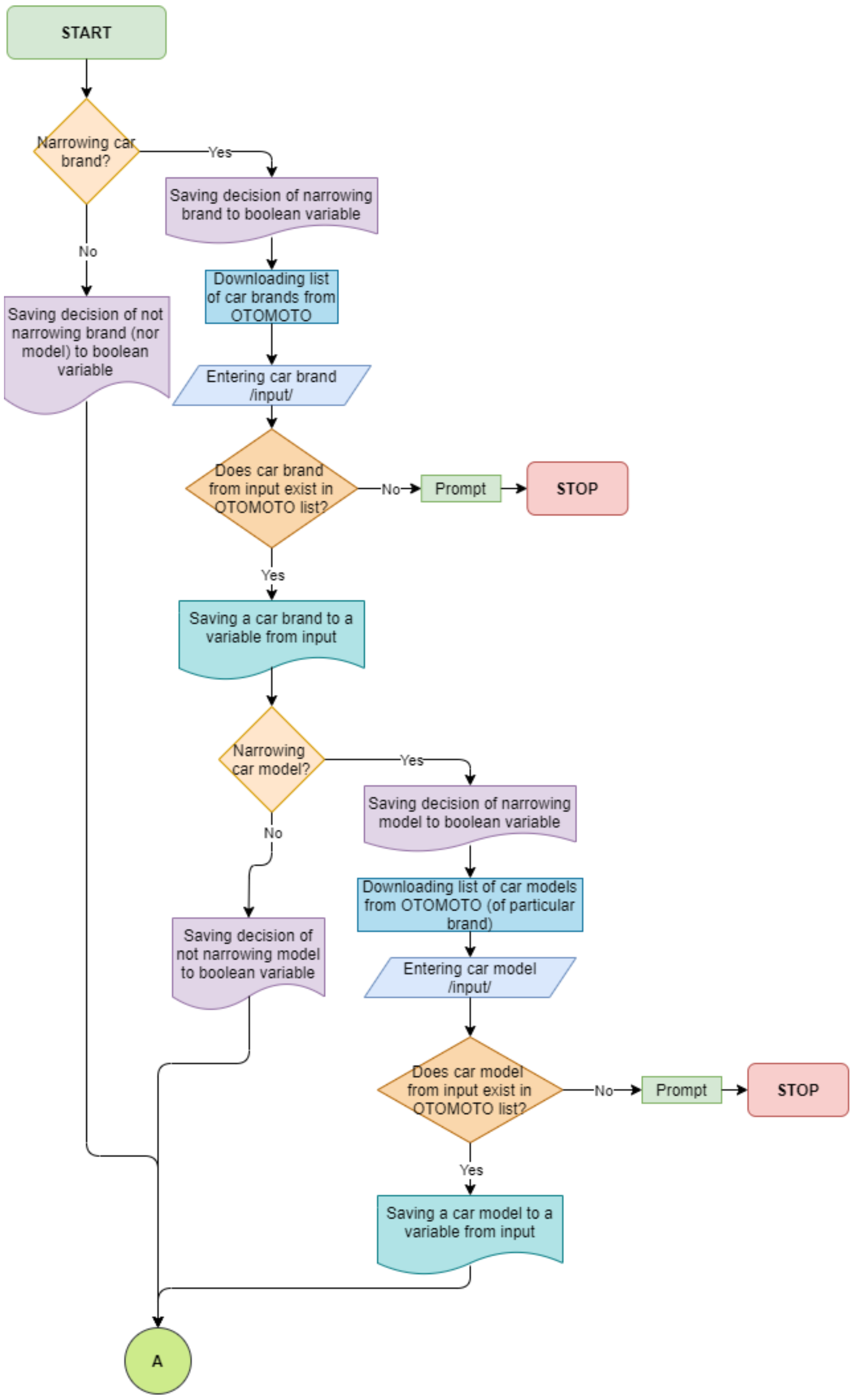

START
Narrowing car brand?
Yes
Saving decision of narrowing brand to boolean variable
No
Saving decision of not narrowing brand (nor model) to boolean variable
Downloading list of car brands from OTOMOTO
Entering car brand /input/
Does car brand from input exist in OTOMOTO list?
No
Prompt
STOP
Yes
Saving a car brand to a variable from input
Narrowing car model?
Yes
Saving decision of narrowing model to boolean variable
No
Downloading list of car models from OTOMOTO (of particular brand)
Saving decision of not narrowing model to boolean variable
Entering car model /input/
Does car model from input exist in OTOMOTO list?
No
Prompt
STOP
Yes
Saving a car model to a variable from input
A

Figure 3.6. The algorithm of the scraping web program visualized by a process diagram

Source: Own study.

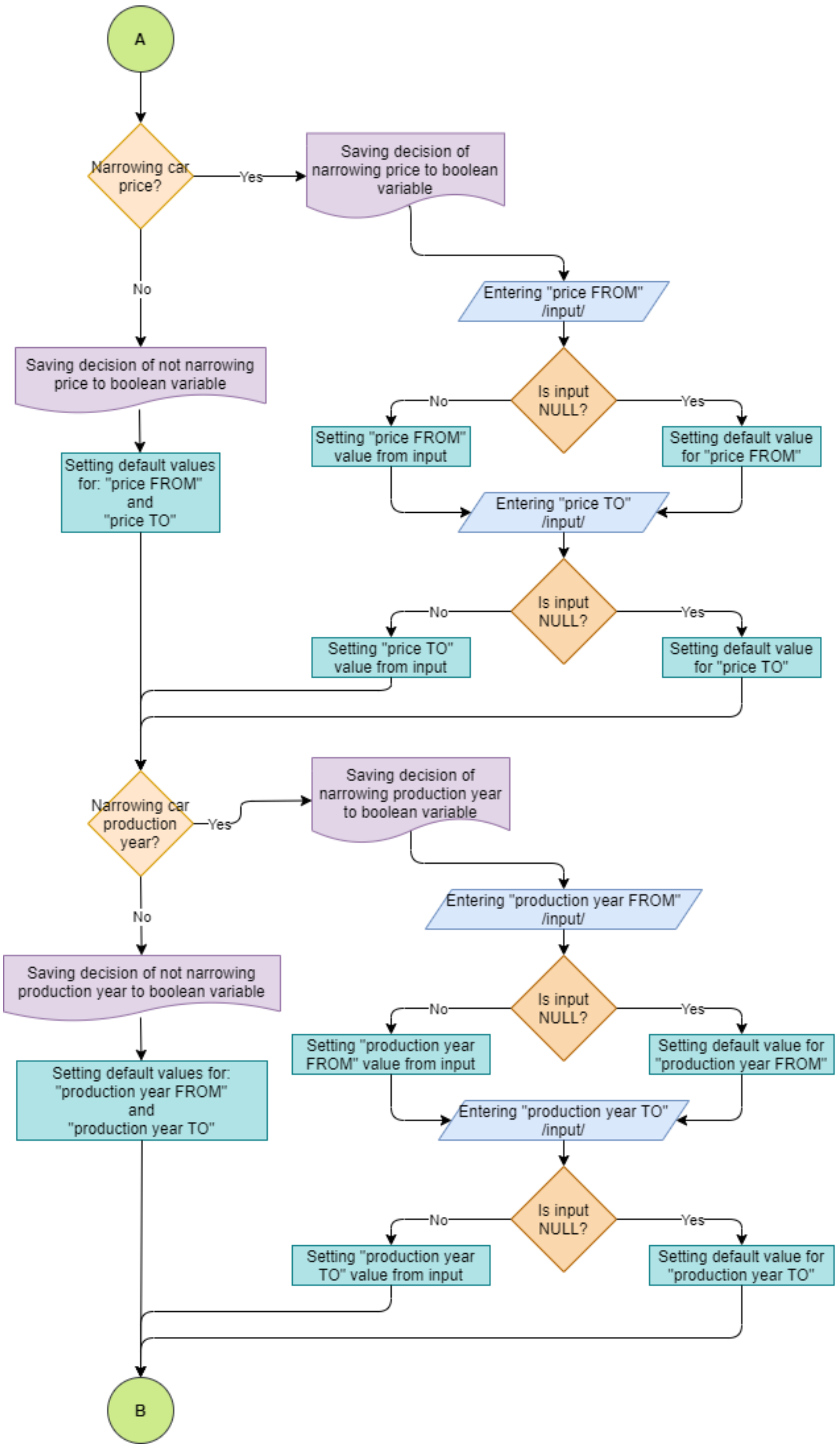

A
Narrowing car price?
Yes
Saving decision of narrowing price to boolean variable
Entering "price FROM" /input/
Is input NULL?
No
Yes
Setting "price FROM" value from input
Setting default value for "price FROM"
Entering "price TO" /input/
Is input NULL?
No
Yes
Setting "price TO" value from input
Setting default value for "price TO"
No
Saving decision of not narrowing price to boolean variable
Setting default values for: "price FROM" and "price TO"
Narrowing car production year?
Yes
Saving decision of narrowing production year to boolean variable
Entering "production year FROM" /input/
Is input NULL?
No
Yes
Setting "production year FROM" value from input
Setting default value for "production year FROM"
Entering "production year TO" /input/
Is input NULL?
No
Yes
Setting "production year TO" value from input
Setting default value for "production year TO"
No
Saving decision of not narrowing production year to boolean variable
Setting default values for: "production year FROM" and "production year TO"
B

Figure 3.6 (cont.).

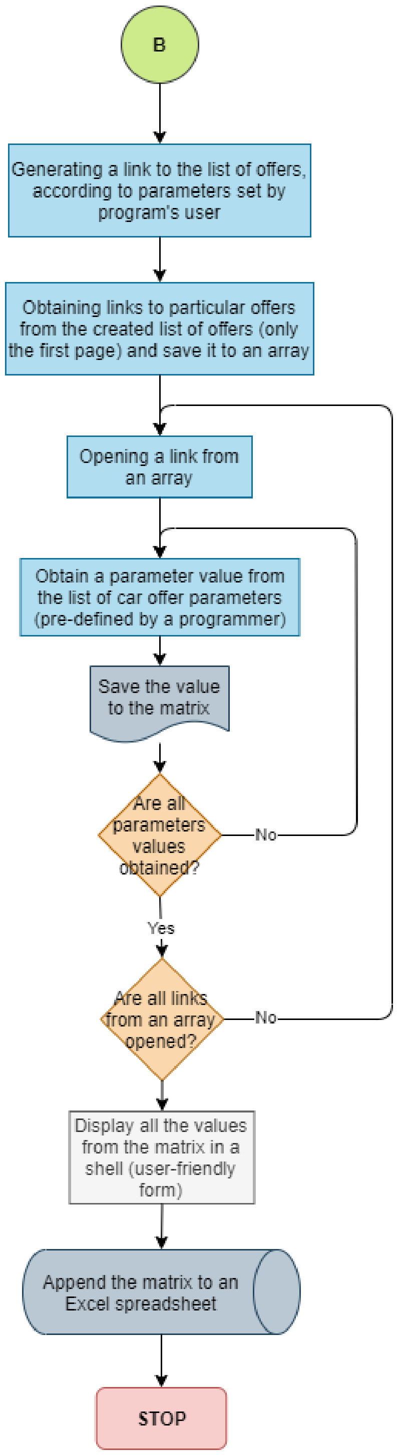

B
Generating a link to the list of offers, according to parameters set by program's user
Obtaining links to particular offers from the created list of offers (only the first page) and save it to an array
Opening a link from an array
Obtain a parameter value from the list of car offer parameters (pre-defined by a programmer)
Save the value to the matrix
Are all parameters values obtained?
No
Yes
Are all links from an array opened?
No
Display all the values from the matrix in a shell (user-friendly form)
Append the matrix to an Excel spreadsheet
STOP

Figure 3.6 (cont.)

Having the algorithm described in this way, proceeding to the phase of proper programming of a scraping web application using the Python language can be the next step. Its code in this paper was written using the PyCharm development environment. The following Subchapter 3.2.2 presents programming effects along with the analysis of the target website (OTOMOTO) and explanatory statements when discussing a particular part of the code.

3.2.2. Preparation of the technical environment

Wanting to write any program to be executed on a computer, it should first be allowed to be interpreted correctly not only by a human but also by a machine. Computers read the code with the help of interpreters of a specific programming language, but the important issue is choosing the appropriate language that would meet the requirements of the idea of web scraping. Such requirements include, among others, high scalability and high availability of various libraries, mainly network libraries. There are many programming languages that can meet such conditions, but along with the great "exquisiteness of coding chic" Python is considered the most appropriate language for extracting data from specific websites (Thomas & Mathur, 2019). In addition, Python, as an interpretable high-order programming language, has a dynamic type system and automated memory management tools. These make it a suitable language for programming scraping tools from both static and dynamic content pages (Brown, 2018). This language was also chosen to make a data extraction tool on the example of the OTOMOTO advertising portal in order to achieve the work's objectives.

The program was designed and developed on a computer running the Windows 10 operating system. As the Python language is not installed by default in Windows (nor in Mac OS as well as Linux/UNIX systems), it must be downloaded manually. It is best to download the program from the official source, i.e., the website managed by the Python Software Foundation[35], https://python.org.

Striving to get the most out of writing a program in a particular programming language (along with the latest security patches), it is recommended to download the latest version of it. As of March 6, 2021, the latest version of Python is 3.9.2, but due to the lack of full compatibility of the language version with the external libraries used in the development of the web scraping tool, it

[35] Python Software Foundation (PSF) is a non-profit organization that owns and manages the intellectual property of the Python programming language. PSF also aims to "promote, protect, and advance the Python programming language (…)" (Python Software Foundation).

was decided to write a program based on Python version 3.8 (with the latest security patches)[36]. On the official *python.org* page, it can be read that version 3.8 is officially supported.

In order to write the program as efficiently as possible, it was decided to write it in an integrated development environment (IDE) with support for projects, multiple files, structures and syntax highlighting. Any environment that supports Python interpretation would be appropriate; however, PyCharm was arbitrarily considered the most appropriate solution, due to its high scalability in the context of Python and “smart” solutions that beautify the code and boost its quality. PyCharm, which was created by JetBrains, was downloaded in the free Community version, which provides a full-fledged PyCharm environment for individual and non-commercial purposes. The download of the suite took place from the official site https://www.jetbrains.com/pycharm. As of January 15, 2021, when the tool design process began, the most recent version of PyCharm was the 2020.3.2 version that was used to write the program.

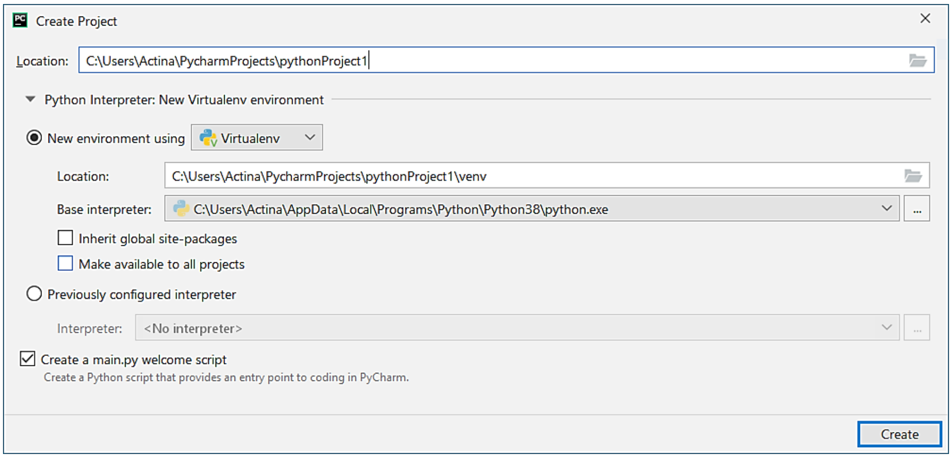
After the correct installation of the PyCharm environment, a new project should be initiated there, along with the selection of a name or interpreter. Since previously a manual version of Python was installed, the IDE detects it automatically and suggests we use an internal interpreter. For the purposes of this work, the name of the project is *otoMotoExtract*. Figure 3.7 (see p. 63) shows the project initiation dialog in the PyCharm environment[37].

The *otoMotoExtract* project is created in a specific space called “virtual environment” (*virtualenv* or *venv*), which offloads the project itself by reducing its size and isolating it from system directories and making it more susceptible to changes and the implementation of external libraries. When creating a project, a list of files needed for the correct operation of the program is also created. The most important of the newly-created files is the *main.py* file[38], which is a specific executable initiator of the code. The full list of files (including those added by the author) will be described in Subchapter 3.3.

[36] The *BeautifulSoup* library, which is described in Chapter 3.3, at the moment of starting the development of the tool in the development environment, was not compatible with the Python language version 3.9, which resulted in the inability to install it correctly to the project.

[37] As the name *otoMotoExtract* is already taken, the name *pythonProject* was adopted for the purpose of showing the project initiation dialog in the PyCharm environment (as seen in Figure 3.6).

[38] The .py extension indicates a file written in the Python programming language that can be interpreted with Python interpreter.


Create Project

Location: C:\Users\Actina\PycharmProjects\pythonProject1

Python Interpreter: New Virtualenv environment

● New environment using Virtualenv

Location: C:\Users\Actina\PycharmProjects\pythonProject1\venv

Base interpreter: C:\Users\Actina\AppData\Local\Programs\Python\Python38\python.exe

☐ Inherit global site-packages

☐ Make available to all projects

○ Previously configured interpreter

Interpreter: <No interpreter>

☑ Create a main.py welcome script

Create a Python script that provides an entry point to coding in PyCharm.

Figure 3.7. PyCharm environment dialog box initiating the creation of the project

Source: Own study.

In order to achieve the goals of the work and create a fully functional web scraping tool, Python alone with all its basic functionalities is insufficient. However, due to the previously mentioned possibility of extending the Python environment with additional libraries, it is possible to design such a tool with the help of these extensions. The Beautiful Soup version 4 (*bs4*) library was chosen as the basis for creating the data extractor, which, as it can be read in the official documentation, is a library that extends the Python language with tools that retrieve data from HTML and XML files (Crummy.com). In addition, as the project was developed and subsequent stages were implemented, the *Selenium* library, which is applied to automate web operations in browsers, was also exploited (Selenium.dev). Finally, once the data display was accessed, the Pandas library, which is used to analyze data in Python, was used to properly represent the retrieved information to export the data to a structured table in Microsoft Excel format (Pydata.org). This form of data presentation can be considered friendly for the purpose of further data analysis.

Installing additional libraries to the Python environment is most conveniently done through the PIP package manager. From version 3.4, PIP is installed by default into Python, so installing extensions can take place immediately after Python is properly installed on our computer. Libraries are added via `pip install <library_name>` from the Python console entry (also installed by default), but with the help of the PyCharm environment, there is a “dialog” way to add extensions. Namely, PyCharm has an integrated PIP package manager, which can be run from the “Settings → Project → Python Interpreter” dialog. In this section of the window, a list of installed libraries can be found, and the installation of additional ones is carried out using the "+" button. There, a library to be installed can be selected from the drop-down list or via entering its name in the search field. After pressing the "Install" button, the same operation is executed as for the discussed `pip install` command, with the difference, however, that it is performed without the participation of the Python shell[39].

The above steps, after their correct implementation, are enough to move to the tool design stage, primarily at the programming level but also at the level of conceptual work. The following Subchapter 3.3 focuses on the implementation of the concept into the programming part with the use of the previously prepared environment. As the created web scraping tool will be used to extract data from a specific set of pages, each design stage will relate to the analysis of the pages in terms of their technical structure (mainly related to the analysis of HTML and CSS scripts). The following Chapter also presents programming effects along with the technical analysis of the target website (OTOMOTO) and explanatory statements when discussing a particular part of the code.

3.3. Web extractor – design and programmatic implementation

Wanting to design a data extraction tool, as well as any other computer application, a few important issues that constitute the axis of this subsection should be remembered. These are, first of all, the proper structure of the program, code cleanliness, accessible communication with the user, correct implementation of algorithmic solutions, as well as testing the constructed tool with validation.

At this point, it is also important to mention a few stylistic procedures used later in this Chapter and work. Any reference to a file name (including its extension) as well as to a library name or notation used in this development environment is in *italics*. On the other hand, code fragments, such as operators, aliases or names of functions, objects or variables, are written in the `Courier New` font.

[39] Figures showing the process of installing external libraries using the PyCharm environment can be found in Appendix – Figures A.2 and A.3.

3.3.1. Structure of the project in the PyCharm environment

Having a prepared and properly configured development environment, it shall be proceeded to activities aimed at structuring and systematizing individual code fragments. It was decided that the entire program, as an *otoMotoExtract* project, will consist of three separate files, making up the whole. These components are:

- executable part – located in the *main.py* file[40],
- part of the code responsible for web scraping – located in the *otoMoto Scraper.py* file,
- definitions of executable functions – located in the *otoMotoFunctions.py* file.

In the programming project folder, there is also a Microsoft Excel spreadsheet format file, i.e., *otoMotoList.xlsx*, which is a structured output of the results being obtained. The project resources also include files of imported external libraries that supplement the code, as well as auxiliary files responsible for the correct creation of a virtual environment (*venv*). It is also worth paying attention to the *geckodriver.log*, which is a working file for the Selenium library used in writing a certain part of the code. However, it does not constitute an essential element of this work. The list of *otoMotoExtract* project files is shown in Figure 3.8.

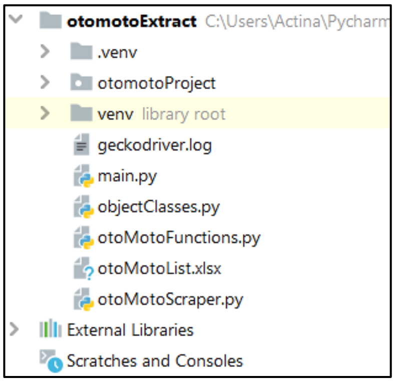

```
otomotoExtract C:\Users\Actina\Pycharm
  .venv
  otomotoProject
  venv library root
  geckodriver.log
  main.py
  objectClasses.py
  otoMotoFunctions.py
  otoMotoList.xlsx
  otoMotoScraper.py
External Libraries
Scratches and Consoles
```

Figure 3.8. List of files in the otoMotoExtract project

Source: Own study.

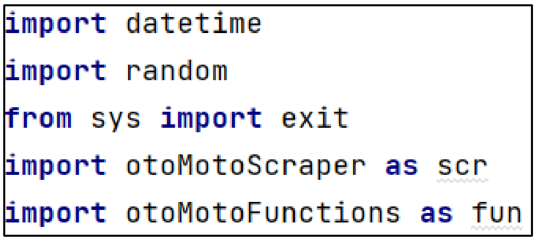
[40] The main.py file is a file created directly by the IDE when the project was created. It contains a main() function that is automatically run when the program is started.

3.3.2. Program communication with the user – collecting information

Since a file (especially its functions) written in the Python programming language, as in most other languages, is executed line by line, from top to bottom, code exploitation in this study is also performed this way. When starting the analysis of the main executable file (*main.py*), it is important to pay attention to the fact that the code fragments responsible for importing external libraries should be included in the initial lines of the code for their correct implementation and references to them later. In the *main.py* executable, the libraries listed in Figure 3.8 are imported via the `import` operant. It also shows a reference to the rest of the program files with the help of friendly and short aliases:

- *otoMotoScraper.py* file is referenced by the `scr` alias,
- *otoMotoFunctions.py* file is referenced by the `fun` alias,

Creating aliases is optional, but they do make the code look neater and tidier. Referencing via aliases is done via the `as` operant (Figure 3.9).

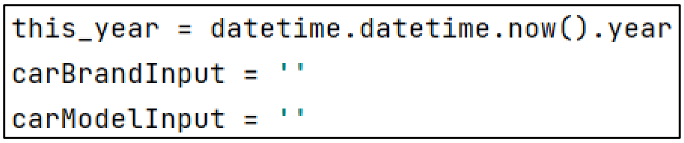


```
import datetime
import random
from sys import exit
import otoMotoScraper as scr
import otoMotoFunctions as fun
```

Figure 3.9. Lines of code responsible for importing specific libraries and project files

Source: Own study.

At the beginning, variables were prepared that are useful in the later stages of coding. The variable named `this_year`, which uses the `datetime` library functions, points to the current year. It was created to avoid imposing a specific year by the programmer and to be created dynamically. It will be useful with the function of narrowing down the year of production. It is not necessary to initialize the variables at this point, as Python allows variables to be created dynamically, but in this case the goal was to make the code clearer and avoid potential referencing problems. The `carBrandInput` and `carModelInput` variables are designed to store what the user enters (car make and model). The initiation of the above-discussed variables is presented in Figure 3.10.

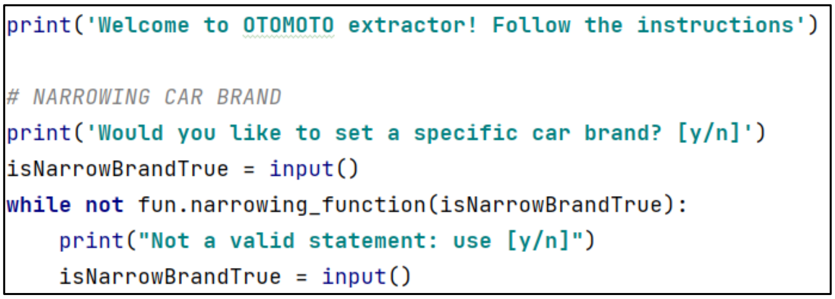


```
this_year = datetime.datetime.now().year
carBrandInput = ''
carModelInput = ''
```

Figure 3.10. Variable initialization

Source: Own study.

Initialization of these variables is followed by the first message that appears in the Python shell – the welcome prompt, asking the user to follow the instructions, displayed using the `print` operator. Immediately following this message is the first question directed at the user about the desire to narrow down the brand. This is where the newly created `narrowing_function` is called (referenced by the alias `fun` and initiated with the `def` operator[41]), which returns a binary variable. It is true only if the argument is "y" or "n", otherwise it is false. This function is looped around in a `while` loop, which means that this function will run as long as the argument is not "y" or "n". A helper variable holding the letters "y" or "n", indicating user's decision about narrowing the brand, is hidden under the name `isNarrowBrandTrue`. The variable takes the value from what the user enters in the input field. The definitions of the present code snippets are shown in Figures 3.11 and 3.12.

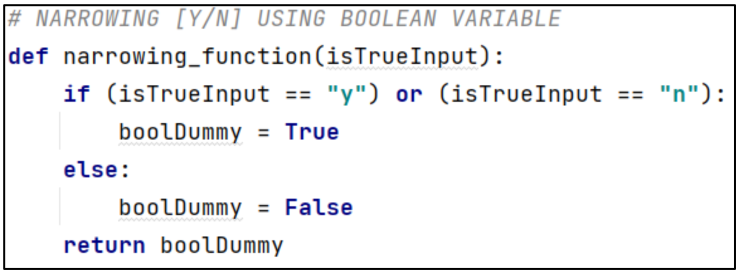


```
print('Welcome to OTOMOTO extractor! Follow the instructions')

# NARROWING CAR BRAND
print('Would you like to set a specific car brand? [y/n]')
isNarrowBrandTrue = input()
while not fun.narrowing_function(isNarrowBrandTrue):
    print("Not a valid statement: use [y/n]")
    isNarrowBrandTrue = input()
```

Figure 3.11. The code fragment responsible for displaying the welcome message and for the functioning of the quasi-binary loop.

Source: Own study.

[41] Creating a function in the Python programming language is always done with the def operator. It is followed by the name of the function, and in parentheses any arguments that the function takes.



```python
# NARROWING [Y/N] USING BOOLEAN VARIABLE
def narrowing_function(isTrueInput):
    if (isTrueInput == "y") or (isTrueInput == "n"):
        boolDummy = True
    else:
        boolDummy = False
    return boolDummy
```

Figure 3.12. Definition of a quasi-binary loop

Source: Own study.

Then a condition occurs that affects the further operation of the algorithm and calling specific functions. This is a condition for checking whether the user wants to filter the search results by vehicle brand. It is invoked in accordance with the aforementioned process diagram, as presented in Figure 3.6. If the user enters the letter "y", which expresses the desire to apply the filter to the vehicle brand, then after the appropriate message is displayed for the user, the first thing the program does is to download all vehicle brands from the OTOMOTO website. Such action should be done for the sake of preventing the misspelling of the car brand or entering one that does not exist. The `getOto-MotoBrands` function (accepting no arguments), which was defined in the *otoMotoFunctions.py* file, is called (similarly to a narrowing function), the alias fun. This is the first function to be called to retrieve data from a website, and its core is the *BeautifulSoup* library, which is dedicated to retrieving data from internet sources. The library should be imported into the *otoMotoFunctions.py* file, similar to the *main.py* file. The imported libraries in the file are shown in Figure A.4. To properly initialize the scraper, a `BeautifulSoup` object is created that takes two arguments:

- `urlopen` object (which, in turn, takes a link as an argument) – it is used to indicate the website from which data is to be retrieved;
- a string that defines the parsing method (and thus indicates the language used on the target page).

For the sake of code transparency, it was decided to break this initialization into two steps: in the first, an object of the `urlopen` type is created, and in the next – `BeautifulSoup`. The OTOMOTO classifieds website, which starts with the common core (https://otomoto.pl/osobowe) is a starting point in many situations, so it was decided to assign this part of the URL to the `oto-MotoStartUrl` variable. As the entire process involves acquiring data from an external repository and thus the possibility of exceptions, it was decided to include the entire procedure in a `try-except` clause, preparing the code for random situations and avoiding critical errors. Such potentially occurring errors can be connected with server communication (HTTP) or URL link. If the connection with the website has been successfully initiated, an appropriate prompt is displayed.

Then, when the robot knows which page to open, it runs a loop to write all brands to an array. With the bs object (of the `BeautifulSoup` type), one specific element on the page is found using the `find` function, and then all elements that meet the specified condition are found inside it (using the `find_all` function). A specific element on the page is found using a condition that we have to define ourselves when analyzing the website. Hence, on the website with the list of OTOMOTO ads (at https://otomoto.pl/osobowe), in the filters section, there is a drop-down list with car brands, as shown in Figure 3.13. After inspecting the website (e.g., through the Inspector tool in Google Chrome), we will find that the drop-down list is a `select`-type object that has its own identifier (`id`) called "`param571`". The HTML code designation of this element in the Google Chrome Inspector is presented in Figure 3.14. From the discussed drop-down list, all the items that have an assigned value (literally: the value is not `None`, i.e., it cannot be empty) are being obtained. This can be read from inspecting this element, and also that these elements are of the type option. These assumptions complete the condition for the loop to function. There is a conditional expression in the loop that adds values of the type value to the previously initialized array named `carBrands`, as presented in Figure 3.15. The function designed in Python is shown in Figure 3.16.

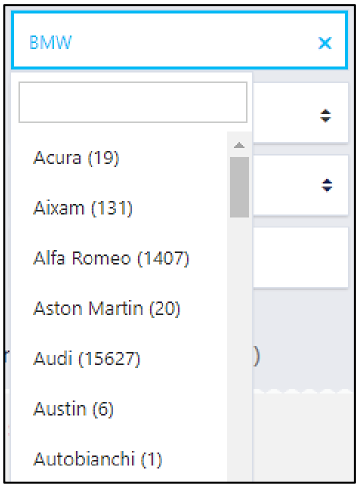
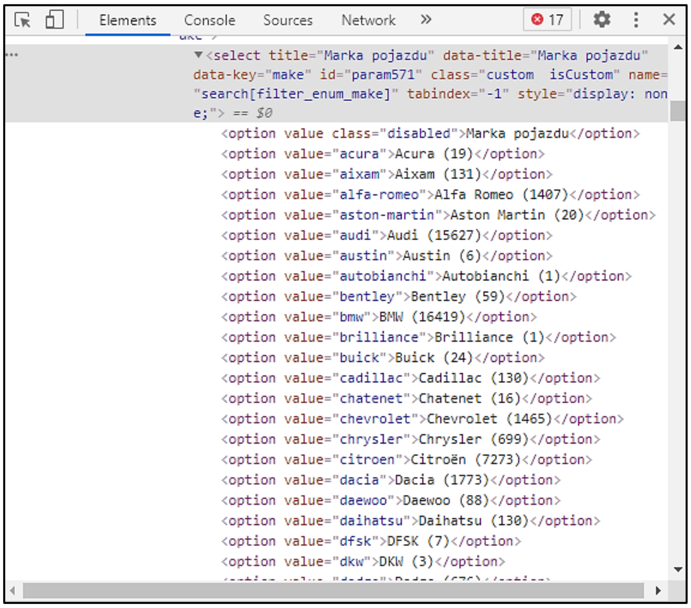


Figure 3.13. OTOMOTO drop-down list with car brands

Source: https://otomoto.pl/osobowe (accessed March 1, 2021).



```
▼<select title="Marka pojazdu" data-title="Marka pojazdu"
data-key="make" id="param571" class="custom  isCustom" name=
"search[filter_enum_make]" tabindex="-1" style="display: non
e;"> == $0
    <option value class="disabled">Marka pojazdu</option>
    <option value="acura">Acura (19)</option>
    <option value="aixam">Aixam (131)</option>
    <option value="alfa-romeo">Alfa Romeo (1407)</option>
    <option value="aston-martin">Aston Martin (20)</option>
    <option value="audi">Audi (15627)</option>
    <option value="austin">Austin (6)</option>
    <option value="autobianchi">Autobianchi (1)</option>
    <option value="bentley">Bentley (59)</option>
    <option value="bmw">BMW (16419)</option>
    <option value="brilliance">Brilliance (1)</option>
    <option value="buick">Buick (24)</option>
    <option value="cadillac">Cadillac (130)</option>
    <option value="chatenet">Chatenet (16)</option>
    <option value="chevrolet">Chevrolet (1465)</option>
    <option value="chrysler">Chrysler (699)</option>
    <option value="citroen">Citroën (7273)</option>
    <option value="dacia">Dacia (1773)</option>
    <option value="daewoo">Daewoo (88)</option>
    <option value="daihatsu">Daihatsu (130)</option>
    <option value="dfsk">DFSK (7)</option>
    <option value="dkw">DKW (3)</option>
```

Figure 3.14. HTML code designation of the OTOMOTO drop-down list with car brands in the Google Chrome Inspector

Source: https://otomoto.pl/osobowe (accessed March 1, 2021).

```
carBrands = []
carBrandModels = []
offerUrls = []
otoMotoStartUrl = 'https://otomoto.pl/osobowe/'
```

Figure 3.15. Initiation of array variables

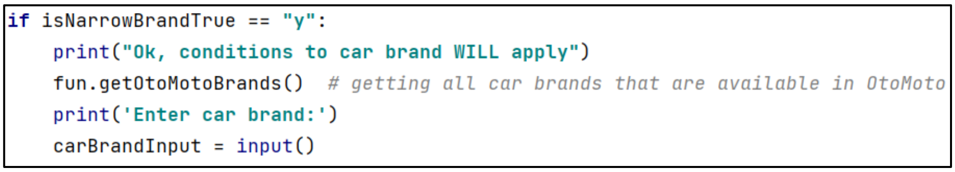
Source: Own study.

```
def getOtoMotoBrands():
    try:
        html = urlopen(otoMotoStartUrl)
        bs = BeautifulSoup(html, 'html.parser')
        for brand in bs.find('select', {'id': 'param571'}).find_all(
                'option', value=not None):
            if 'value' in brand.attrs:
                carBrands.append(brand.attrs['value'])
    except HTTPErrorProcessor as e:
        print(e)
    except URLopener as e:
        print('A problem with URL occurred: ', e)
    else:
        print('Connection established')
```

Figure 3.16. Definition of a function responsible for acquiring car brands from the OTOMOTO website

Source: Own study.

Going back to the *main.py* file, where the program is executed: after retrieving the brand list, the program asks the user to enter the brand by which he wants to filter. The user's choice is assigned to the variable of the type input, similarly to the quasi-binary narrowing function mentioned previously. This step is illustrated in Figure 3.17.



```
if isNarrowBrandTrue == "y":
    print("Ok, conditions to car brand WILL apply")
    fun.getOtoMotoBrands()  # getting all car brands that are available in OtoMoto
    print('Enter car brand:')
    carBrandInput = input()
```

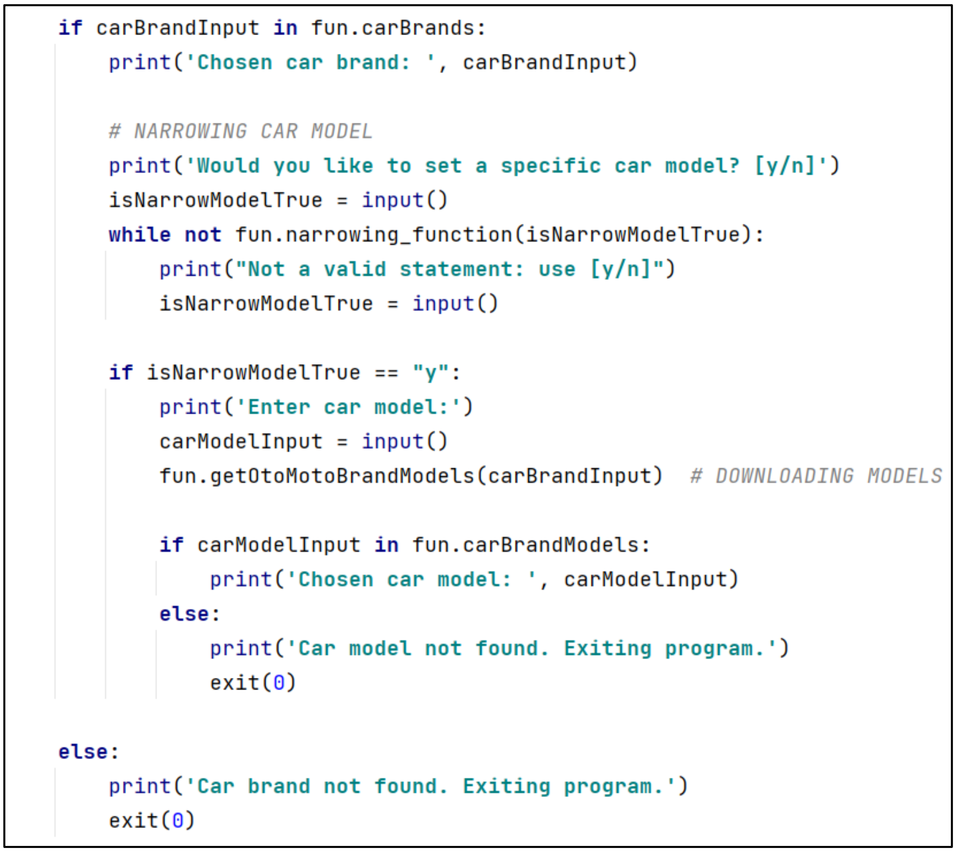
Figure 3.17. Assigning user's input to a variable

Source: Own study.

Then it is checked whether the brand entered by the user, i.e., his input, is on the list of previously downloaded models. If it is not present, the program is terminated, informing the user beforehand with an appropriate message. If the brand is on the list, a check is made to see if the user wants to narrow it down to the model (given brand). This condition is embedded one level lower, which means that it is only checked when the user also wants to filter the vehicle brand, which prevents reference to an empty item. Such action is also consistent with the logic: the vehicle model is always assigned to a specific brand (it is a "child" of the brand – because model inherits from it) and it is not possible to narrow down the model without prior filtering of the car make.

The procedures for checking the user's willingness to filter against the model, entering the model name as well as appending the values into the array[42] are performed in the same way as for the activities related to the vehicle brand. The only difference in this procedure is the process of retrieving car model names. This action is done with the use of the newly created `getOtoMotoBrandModels(string_input)` function which takes the brand name entered by the user as an argument and which is stored in the `carBrandInput` variable. The argument of this function is necessary in order to correctly generate a link to the classifieds page of a specific brand from which the list of models will be downloaded. The elements of the algorithm that trigger the procedures of narrowing down models and retrieving values from the user are shown in Figure 3.18, while the function retrieving a list of models of a given brand is shown in Figure 3.19.

[42] The array of car model names is stored in a variable named `carBrandModels`.

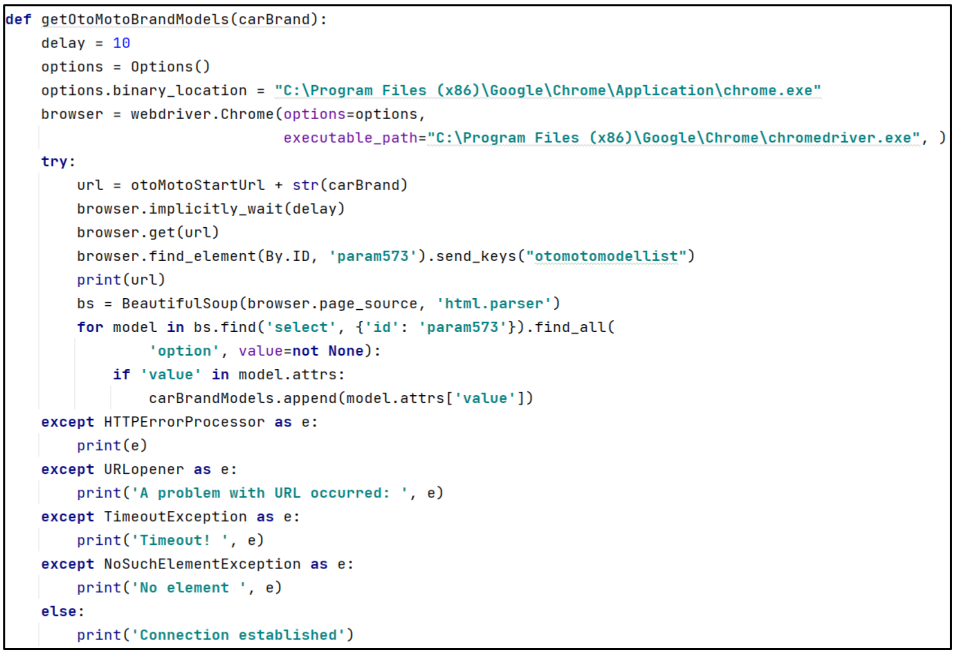


```
if carBrandInput in fun.carBrands:
    print('Chosen car brand: ', carBrandInput)

    # NARROWING CAR MODEL
    print('Would you like to set a specific car model? [y/n]')
    isNarrowModelTrue = input()
    while not fun.narrowing_function(isNarrowModelTrue):
        print("Not a valid statement: use [y/n]")
        isNarrowModelTrue = input()

    if isNarrowModelTrue == "y":
        print('Enter car model:')
        carModelInput = input()
        fun.getOtoMotoBrandModels(carBrandInput)  # DOWNLOADING MODELS

        if carModelInput in fun.carBrandModels:
            print('Chosen car model: ', carModelInput)
        else:
            print('Car model not found. Exiting program.')
            exit(0)

else:
    print('Car brand not found. Exiting program.')
    exit(0)
```

Figure 3.18. Procedures of narrowing down models and retrieving values from the user

Source: Own study.

As it can be seen in Figure 3.18, the `getOtoMotoBrandModels` function differs slightly from the `getOtoMotoBrands` function. In this case, apart from the *BeautifulSoup* library, used to download data from the website, there was a need to use the *Selenium* library in parallel, which is used to imitate browser behavior. This type of action had to be used in order to correctly download data from the OTOMOTO website. It has been noticed that loading the page with OTOMOTO advertisements takes place in stages. As previously mentioned, the model is brand-dependent, so in this case the filter element storing the model list (depending on the car make) is starting to load only after the element on the page storing the selected brand is correctly loaded. The *BeautifulSoup* library does not allow for manipulation of conditional page loading, so we have to use a library that allows us to wait on the page, in this case *Selenium*. The *Selenium* type object (which in the discussed program is called browser) is created with the help of objects inheriting from the *Options* and *webdriver* sub-libraries (responsible for the access path to the browser and for creating an automated browser driver, respectively).



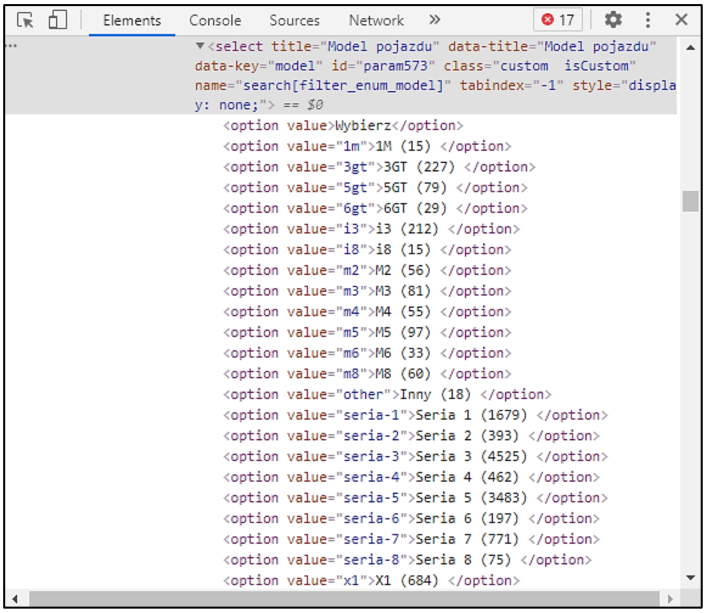
```
def getOtoMotoBrandModels(carBrand):
    delay = 10
    options = Options()
    options.binary_location = "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe"
    browser = webdriver.Chrome(options=options,
                               executable_path="C:\Program Files (x86)\Google\Chrome\chromedriver.exe", )
    try:
        url = otoMotoStartUrl + str(carBrand)
        browser.implicitly_wait(delay)
        browser.get(url)
        browser.find_element(By.ID, 'param573').send_keys("otomotomodellist")
        print(url)
        bs = BeautifulSoup(browser.page_source, 'html.parser')
        for model in bs.find('select', {'id': 'param573'}).find_all(
                'option', value=not None):
            if 'value' in model.attrs:
                carBrandModels.append(model.attrs['value'])
    except HTTPErrorProcessor as e:
        print(e)
    except URLopener as e:
        print('A problem with URL occurred: ', e)
    except TimeoutException as e:
        print('Timeout! ', e)
    except NoSuchElementException as e:
        print('No element ', e)
    else:
        print('Connection established')
```

Figure 3.19. Function retrieving a list of models of a given brand

Source: Own study.

At this point, it is necessary to raise and pay attention to two issues: the way of data transfer ("cooperation") between objects of two different types (coming from two different libraries) and the way of calling the function of waiting for the element to load. With the first point in mind, both the *BeautifulSoup* and Selenium library objects use their browser drivers when using Internet resources. This means that despite the fact that there were procedures on the website using the *Selenium* type object (such as waiting procedure), when the *BeautifulSoup* type object is created in the standard way, the actions made by the *Selenium* type object will not be detected. However, there is a way to pass the browser driver from a Selenium object to a *BeautifulSoup* object. Namely, when creating an object of the *BeautifulSoup* type instead of entering a website address as an argument, we can use the page_source attribute on the Selenium type object. Thus, all data retrieval activities by the *BeautifulSoup* type object take place on the *Selenium* object browser driver.

The second issue that needs to be raised at this point is the aforementioned waiting procedure. The `implicitly_wait(time)` function was used, which is part of the *Selenium* library in question, and which directly dictates the time that the browser has to wait (in seconds). Indeed, there is a "more elegant" way to load an element on the page, i.e., more generic and explicit, but in this particular case it cannot be used. It is the `wait.until` function, where the `visibility_of_element_located` and `element_to_be_clickable` selectors can be used as an argument that inherits from the `expected_conditions` sub-library. These methods allow for a generic treatment of the waiting procedure, which would allow waiting until the element is loaded (i.e., visible or clickable, respectively). Unfortunately, after analyzing the OTOMOTO page, it turns out that the `select`-type element that stores the list of models is not visible to automatic tools, although it is visible to the user (it has the "`display: none;`" parameter set for the `style` attribute), as shown in Figure 3.20. Likewise, this item is not clickable. There is a separate element that is visible (for the browser and the user) and, therefore, also clickable, but with its help it is impossible to access the list of models, because they are passed to the visible element on the page through hidden inheritance.



```
▼<select title="Model pojazdu" data-title="Model pojazdu"
data-key="model" id="param573" class="custom  isCustom"
name="search[filter_enum_model]" tabindex="-1" style="displa
y: none;"> == $0
    <option value>Wybierz</option>
    <option value="1m">1M (15) </option>
    <option value="3gt">3GT (227) </option>
    <option value="5gt">5GT (79) </option>
    <option value="6gt">6GT (29) </option>
    <option value="i3">i3 (212) </option>
    <option value="i8">i8 (15) </option>
    <option value="m2">M2 (56) </option>
    <option value="m3">M3 (81) </option>
    <option value="m4">M4 (55) </option>
    <option value="m5">M5 (97) </option>
    <option value="m6">M6 (33) </option>
    <option value="m8">M8 (60) </option>
    <option value="other">Inny (18) </option>
    <option value="seria-1">Seria 1 (1679) </option>
    <option value="seria-2">Seria 2 (393) </option>
    <option value="seria-3">Seria 3 (4525) </option>
    <option value="seria-4">Seria 4 (462) </option>
    <option value="seria-5">Seria 5 (3483) </option>
    <option value="seria-6">Seria 6 (197) </option>
    <option value="seria-7">Seria 7 (771) </option>
    <option value="seria-8">Seria 8 (75) </option>
    <option value="x1">X1 (684) </option>
```

Figure 3.20. Drop-down list of a particular car brand models on the OTOMOTO website represented in HTML code (in Google Chrome inspector)

Source: https://otomoto.pl/osobowe/bmw (accessed on March 1, 2021).

Because of such a situation, selectors that could be used in the `wait.until` function are not capable to be exploited in this case. To work around this problem, an unconventional solution has been made with the function `implicitly_wait(time)`. The wait time was set to 10 seconds – although the element usually loads after about 3-5 seconds, the time buffer is left for slower connections but not stretching it too much, in order to optimize the code still as much as possible.

At this point, it is also worth paying attention to the fact that in the *Selenium* library, referencing specific elements on the page is done in a different way than in the case of the *BeautifulSoup* library. This is done through the `find_element` function with the following arguments:

- the type of attribute that we want to find on the page (this should be done via the *By* sub-library),
- parameter value for this attribute.

A great part of this function has also been given a `try-except` clause with exception handling, again to avoid random events and, therefore, fatal errors. Apart from the HTTP nature of exceptions, mainly the handling of a timeout exception as well as a selector/pointer exception have been added.

After a possible narrowing of the car brand and model, regardless of the user's will expressed in relation to these elements, the values of the price range and the year of production range filters should be collected, if the user so wishes. Collecting that data in both cases is akin to narrowing down the make and model of the vehicle, i.e., using the input field, the newly developed quasi-binary narrowing function (with *y/n* responses), and saving the values provided by the user to variables:

- in the case of price range:
  - from: `priceRangeFromInput`
  - to: `priceRangeToInput`
- in the case of production year range:
  - from: `productionYearFromInput`
  - to: `productionYearToInput`

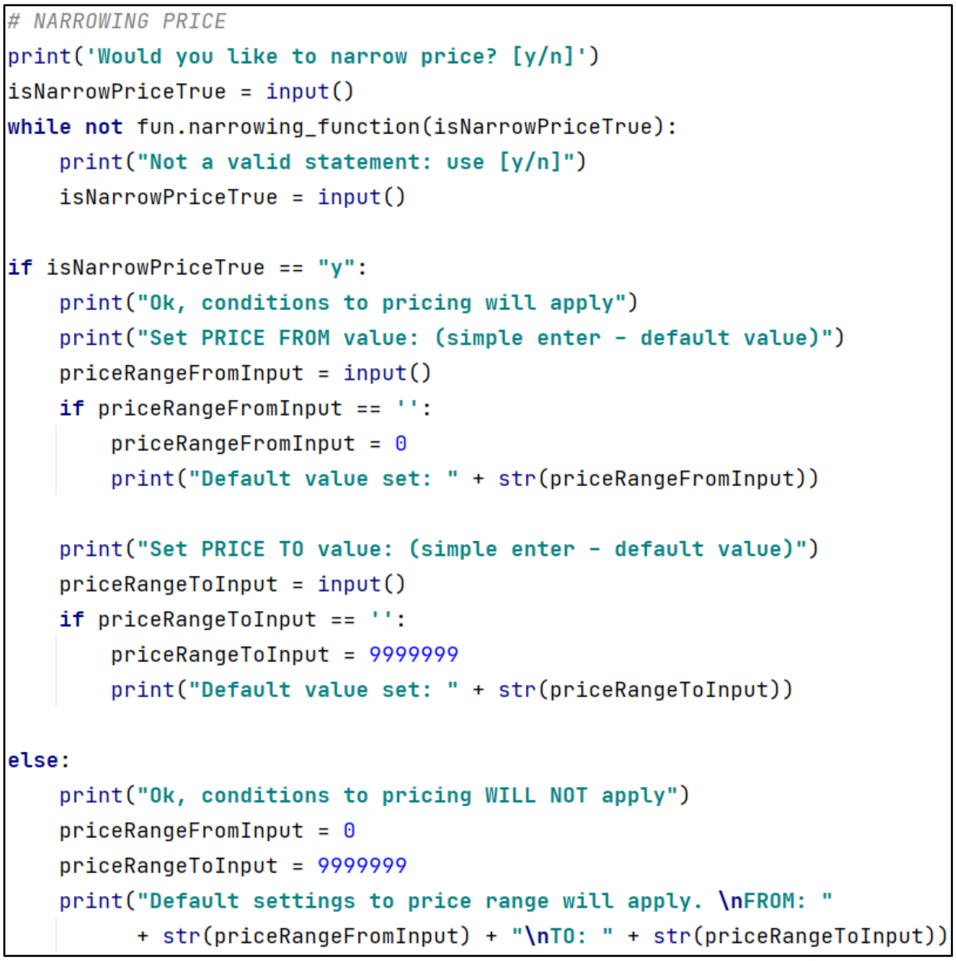
Each step and user's input is confirmed by a message displayed in the shell.

The only significant difference from the brand and model procedures is the handling of default values. It was assumed that the user may be willing to narrow down the price and year of production unilaterally, i.e., leaving one part of the range open. In order not to overcomplicate the code and to ensure its transparency and maximum communicativeness of the program with the user, it was decided that instead of asking more yes-no (lower-order) questions, it would be a better solution to handle default values. The user, wanting to leave one value of the interval blank, leaves the field blank by clicking enter, which is communicated by a prompt. In this case, the default values are assigned:

- in the case of the price range:
  - from: 0 PLN
  - to: 9,999,999 PLN
- in the case of the production year range:
  - from: year 1900
  - to: *current year*[43]

Also, if the user does not want to narrow down the search in relation to the price, the steps taking values from him are skipped and thus default values in beginning and end of the interval are automatically assigned, informing about this fact with an appropriate message. The same situation occurs when there is no desire to filter in relation to the year of production. The parts of the code narrowing the price and production year are shown in Figures 3.21 and 3.22, respectively.

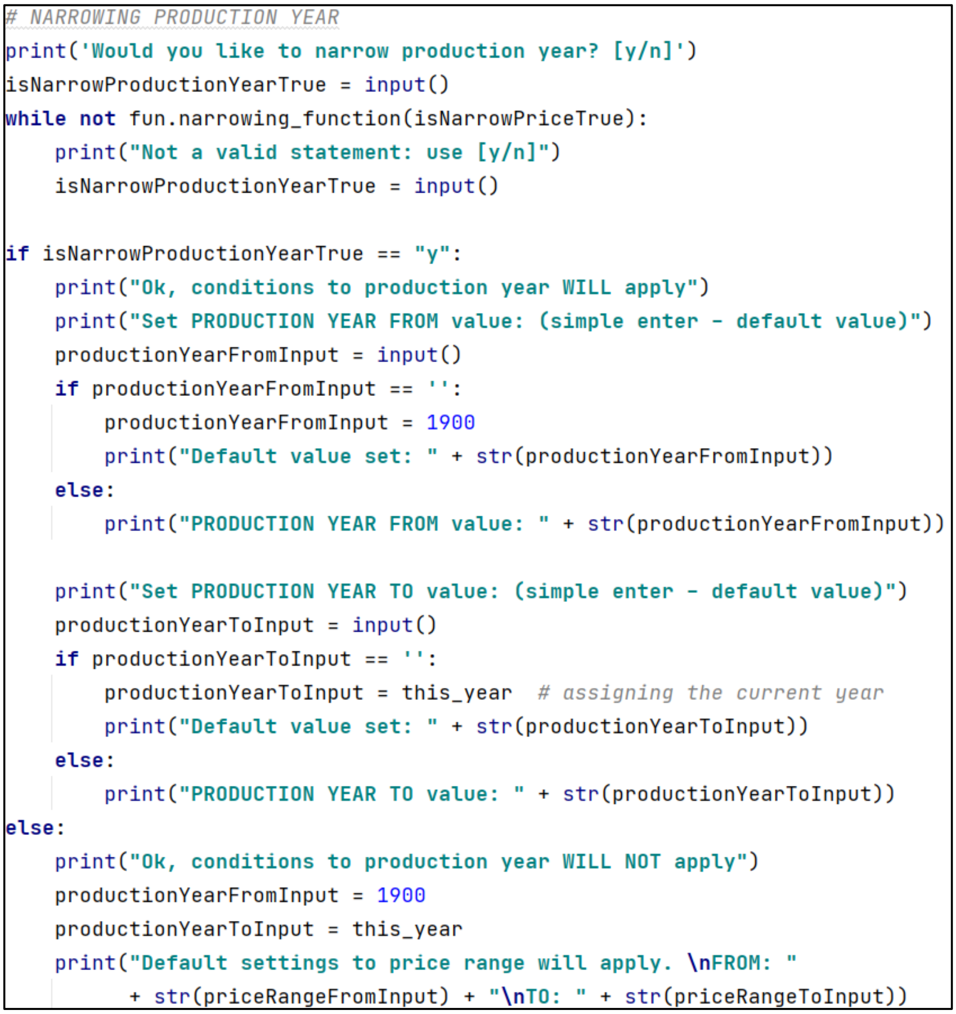
[43] The variable dynamically assigns the current year based on the previously created variable `this_year`, which obtains the current year through the function `datetime.datetime.now().year` with the use of the `datetime` library.



```python
# NARROWING PRICE
print('Would you like to narrow price? [y/n]')
isNarrowPriceTrue = input()
while not fun.narrowing_function(isNarrowPriceTrue):
    print("Not a valid statement: use [y/n]")
    isNarrowPriceTrue = input()

if isNarrowPriceTrue == "y":
    print("Ok, conditions to pricing will apply")
    print("Set PRICE FROM value: (simple enter - default value)")
    priceRangeFromInput = input()
    if priceRangeFromInput == '':
        priceRangeFromInput = 0
        print("Default value set: " + str(priceRangeFromInput))

    print("Set PRICE TO value: (simple enter - default value)")
    priceRangeToInput = input()
    if priceRangeToInput == '':
        priceRangeToInput = 9999999
        print("Default value set: " + str(priceRangeToInput))

else:
    print("Ok, conditions to pricing WILL NOT apply")
    priceRangeFromInput = 0
    priceRangeToInput = 9999999
    print("Default settings to price range will apply. \nFROM: "
          + str(priceRangeFromInput) + "\nTO: " + str(priceRangeToInput))
```

Figure 3.21. The part of the code responsible for imposing a filter that narrows down the price

Source: Own study.



```
# NARROWING PRODUCTION YEAR
print('Would you like to narrow production year? [y/n]')
isNarrowProductionYearTrue = input()
while not fun.narrowing_function(isNarrowPriceTrue):
    print("Not a valid statement: use [y/n]")
    isNarrowProductionYearTrue = input()

if isNarrowProductionYearTrue == "y":
    print("Ok, conditions to production year WILL apply")
    print("Set PRODUCTION YEAR FROM value: (simple enter - default value)")
    productionYearFromInput = input()
    if productionYearFromInput == '':
        productionYearFromInput = 1900
        print("Default value set: " + str(productionYearFromInput))
    else:
        print("PRODUCTION YEAR FROM value: " + str(productionYearFromInput))

    print("Set PRODUCTION YEAR TO value: (simple enter - default value)")
    productionYearToInput = input()
    if productionYearToInput == '':
        productionYearToInput = this_year  # assigning the current year
        print("Default value set: " + str(productionYearToInput))
    else:
        print("PRODUCTION YEAR TO value: " + str(productionYearToInput))
else:
    print("Ok, conditions to production year WILL NOT apply")
    productionYearFromInput = 1900
    productionYearToInput = this_year
    print("Default settings to price range will apply. \nFROM: "
          + str(priceRangeFromInput) + "\nTO: " + str(priceRangeToInput))
```

Figure 3.22. The part of the code responsible for imposing a filter that narrows down the production year

Source: Own study.

Having collected all the data from the user, including the car make and model, as well as the filters narrowing the price range and the year of production, it is possible to create a link to the page with offers that are of interest to the user and from which hyperlinks to specific ads will be obtained. The following Subchapter 3.3.3 focuses on these steps.

3.3.3. Creation of the list of offers' URL

Having collected the necessary information from the user about the type of offers he wants to search, regardless of whether the user wanted to narrow down the search results by brand or model, the program is able to create a complete URL which directs to the classifieds page. That particular address will be used when launching the web scraper. At this point, it is worth paying attention to the structure of the URL where the page with the list of advertisements on OTOMOTO is located (according to filters set by the user). An example link looks as follows[44]:

https://www.otomoto.pl/osobowe/**mazda**/**cx-5**/od-**2016**/?search%5Bfilter_float_price%3Afrom%5D=**75000**&search%5Bfilter_float_price%3Ato%5D=**120000**&search%5Bfilter_float_year%3Ato%5D=**2019**

As an example, the above link directs the user to the first page of the list of ads on the OTOMOTO website, where Mazda CX-5 classifieds from the production year from 2016 to 2019 are displayed, with the price ranging from PLN 70,000 to PLN 120,000. Some permanent elements of this link can be noticed, as well as the dynamically changing ones, depending on the user's preferences and thus the filters used. Those dynamically changing URL elements are shown in bold. Thereby, a mockup of the link can be created, with replaceable gaps regarding the filters applied, as shown below:

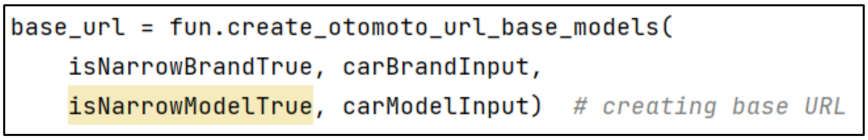
https://www.otomoto.pl/osobowe/**<car_brand>**/**<car_model>**/od-**<production_year_from>**/?search%5Bfilter_float_price%3Afrom%5D=**<price_from>**&search%5Bfilter_float_price%3Ato%5D=**<price_to>**&search%5Bfilter_float_year%3Ato%5D=**<production_year_to>**

By creating a link redirecting to the list of ads, based on the link analysis, two parts of it can be specified: the basic part and the one with additional filters. It was decided to make this kind of distinction for the following reason: the brand and model cannot be replaced with default values (in link creation, they must be removed if the user chose not taking them into account), unlike the price and year of manufacture. Thus, the following occur:

- the base part of the URL: https://www.otomoto.pl/osobowe/**<car_brand>**/**<car_model>**
- the part with additional filters: *the remainder of the address following the above*.

[44] This link to the list of ads has been created based on the use of filters: brand, model, price (from and to), and year of production (from and to). More filters applied make the link longer, and likewise: fewer filters shorten the URL accordingly.

A similar distinction has been made in the creation of URL-composing functions. First, the base part of the link is created by the function create_otomoto_url_base_models, which is assigned to the base_url variable. This function takes four arguments: the value of the quasi-binary function for the brand[45], the value of the quasi-binary function for the model[46], the value for the brand, and the value for the model, as shown in Figure 3.23. Naturally, it returns the base part of the URL, according to the scheme adopted above.



```
base_url = fun.create_otomoto_url_base_models(
    isNarrowBrandTrue, carBrandInput,
    isNarrowModelTrue, carModelInput)  # creating base URL
```

Figure 3.23. Calling the function creating a link to the list of offers on the OTOMOTO website

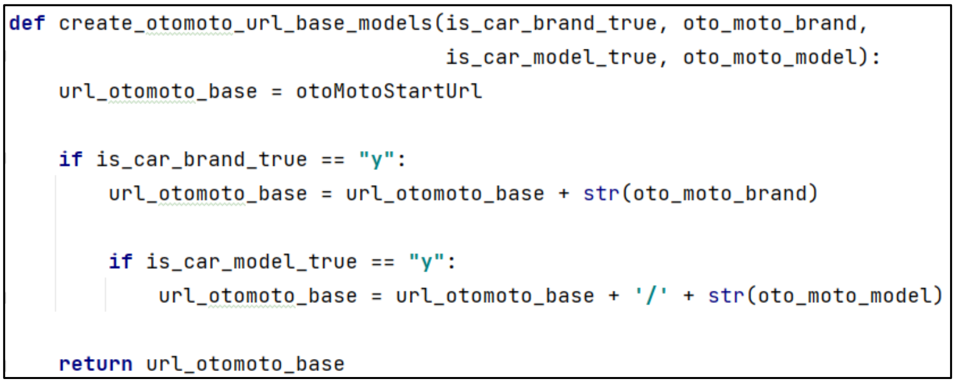
Source: Own study.

The function that creates the basic part of the URL (possibly including the brand or model) starts at the constant element, i.e., https://otomoto.pl/osobowe/ by passing the value from the previously created otoMotoStartUrl variable[47]. The remaining elements (i.e., the name of the brand and model) are appended ("glued") to this fixed element of the address depending on the user's wishes, according to the structure of the OTOMOTO link that is being created. It is worth noting that the conditional "if" procedure for the model is nested inside the parent conditional "if" procedure for the brand, which indicates that the model name cannot appear alone in the address but only accompanied by the brand name. This is in line with the previously mentioned logical rule (described on p. 62), which indicates a complete dependence of the model on the vehicle brand. The link creation also uses string parsing through the str function, which prevents a possible type-incompatibility phenomenon. The definition of the create_otomoto_url_base_models function is shown in Figure 3.24.

[45] The value for this parameter can be "y" or "n" as defined by the aforementioned function called narrowing_function, and therefore the values that the variables holding these responses can take.

[46] Same as above.

[47] The "www" element, which precedes the "otomoto.pl" domain address, is not necessary because of the same page indication.



```python
def create_otomoto_url_base_models(is_car_brand_true, oto_moto_brand,
                                   is_car_model_true, oto_moto_model):
    url_otomoto_base = otoMotoStartUrl

    if is_car_brand_true == "y":
        url_otomoto_base = url_otomoto_base + str(oto_moto_brand)

        if is_car_model_true == "y":
            url_otomoto_base = url_otomoto_base + '/' + str(oto_moto_model)

    return url_otomoto_base
```

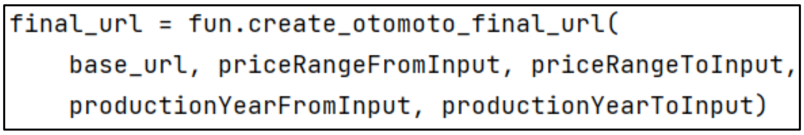
Figure 3.24. Definition of the function creating a part of the list of offers' URL with a car brand and model

Source: Own study.

Having the backbone of the link, consisting of the brand name and model name, we can proceed to the creation of a complete URL, additionally consisting of user-imposed filter parameters. To achieve this goal, the function `create_otomoto_final_url` is used, which takes the following arguments:

- base URL (passed from the `base_url` variable to which the value was assigned by the `return` of the function `create_otomoto_url_base_models`),
- price range (start and end value of the interval),
- production year range (start and end value of the interval).

As mentioned before, if the user does not want to filter by price or year of production, default values are passed to the values of these arguments. Figure 3.25 shows calling this function and assigning its return value to the `final_url` variable at the same time.



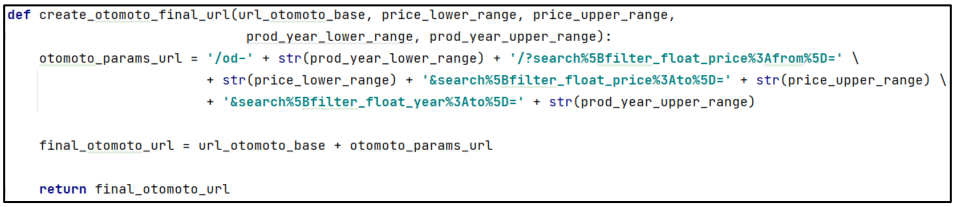
```python
final_url = fun.create_otomoto_final_url(
    base_url, priceRangeFromInput, priceRangeToInput,
    productionYearFromInput, productionYearToInput)
```

Figure 3.25. Calling the function that creates a full URL to the list of offers with ads based on user filters

Source: Own study.

The operation of this function consists of two stages: the first stage creates the appended part of the URL, taking into account the filters imposed by the user, while the second part creates the final version of the link to the

OTOMOTO website with advertisements with user-selected restrictions. As was the case with the base part of the address, the part with filters also consists of constant elements (which are a string – in apostrophes) and dynamic ones passed by function arguments, following the OTOMOTO link scheme. Also, to ensure type compatibility, dynamic elements passed by arguments have been additionally parsed to the string. Part of the link with filters has been assigned to the internal variable `otomoto_params_url`, used to create the full version of the address. The `final_otomoto_url` variable that is returned by the function is of the string type and is created by "sticking" the part of the link with the user's parameters to the base URL. The design of this function is shown in Figure 3.26.



```python
def create_otomoto_final_url(url_otomoto_base, price_lower_range, price_upper_range,
                             prod_year_lower_range, prod_year_upper_range):
    otomoto_params_url = '/od-' + str(prod_year_lower_range) + '/?search%5Bfilter_float_price%3Afrom%5D=' \
                         + str(price_lower_range) + '&search%5Bfilter_float_price%3Ato%5D=' + str(price_upper_range) \
                         + '&search%5Bfilter_float_year%3Ato%5D=' + str(prod_year_upper_range)

    final_otomoto_url = url_otomoto_base + otomoto_params_url

    return final_otomoto_url
```

Figure 3.26. Definition of a function that creates a full URL to the list of ads with ads based on user filters

Source: Own study.

Having a structured link to the list of automotive offers, it is possible to proceed to the stage of obtaining hyperlinks to specific ads from it. Subchapter 3.3.4 introduces the process of creating programming functions to retrieve hyperlinks to specific advertisements from the advertisements list page, to which the previously created link leads.

3.3.4. Retrieving hyperlinks to particular advertisements

The newly created function `addToURLTable` is used for retrieving links to specific offers, according to user's-imposed filters and a previously created link in Subchapter 3.3.3. The function takes as an argument a link to the OTOMOTO list of offers. As offer links are of a homogeneous type and need to be grouped, it has been found that the best solution is to store them in a one-dimensional array[48]. See Figure 3.27 for calling this function in the *main.py* executable.

[48] This array is stored in the previously created table variable `offerUrls`, which is located in the *otoMotoFunctions.py* file. As it is not a private variable and there is an import to the *otoMotoFunctions.py* file in the remaining project files, it can be referred to via the `fun` alias.

```
fun.addToURLTable(final_url)
```

Figure 3.27. Calling the function that adds links to specific offers to the table

Source: Own study.

Due to the fact that this function retrieves specific data from the website based on certain given conditions, the *BeautifulSoup* library is used in this case. The conditions that need to be set to the scraper (in the `find / find_all functions`) arise again from the analysis of the HTML code of the website from which the data is collected. The part of the code that is responsible for displaying the element with the link redirecting to the page of a specific offer is shown in Figure 3.28.

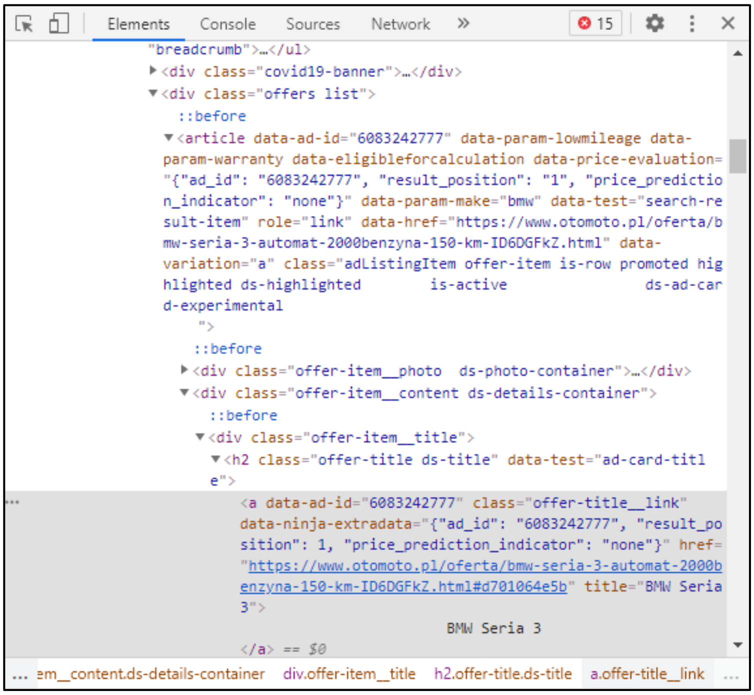

```
"breadcrumb">…</ul>
▶<div class="covid19-banner">…</div>
▼<div class="offers list">
   ::before
  ▼<article data-ad-id="6083242777" data-param-lowmileage data-
  param-warranty data-eligibleforcalculation data-price-evaluation=
  "{"ad_id": "6083242777", "result_position": "1", "price_predictio
  n_indicator": "none"}" data-param-make="bmw" data-test="search-re
  sult-item" role="link" data-href="https://www.otomoto.pl/oferta/b
  mw-seria-3-automat-2000benzyna-150-km-ID6DGFkZ.html" data-
  variation="a" class="adListingItem offer-item is-row promoted hig
  hlighted ds-highlighted        is-active               ds-ad-car
  d-experimental
      ">
    ::before
   ▶<div class="offer-item__photo  ds-photo-container">…</div>
   ▼<div class="offer-item__content ds-details-container">
      ::before
     ▼<div class="offer-item__title">
       ▼<h2 class="offer-title ds-title" data-test="ad-card-titl
        e">
            <a data-ad-id="6083242777" class="offer-title__link"
            data-ninja-extradata="{"ad_id": "6083242777", "result_po
            sition": 1, "price_prediction_indicator": "none"}" href=
            "https://www.otomoto.pl/oferta/bmw-seria-3-automat-2000b
            enzyna-150-km-ID6DGFkZ.html#d701064e5b" title="BMW Seria
            3">
                                        BMW Seria 3
            </a> == $0
… em__content.ds-details-container   div.offer-item__title   h2.offer-title.ds-title   a.offer-title__link …
```

Figure 3.28. Code part responsible for displaying the element with the link redirecting to the page of a specific OTOMOTO offer

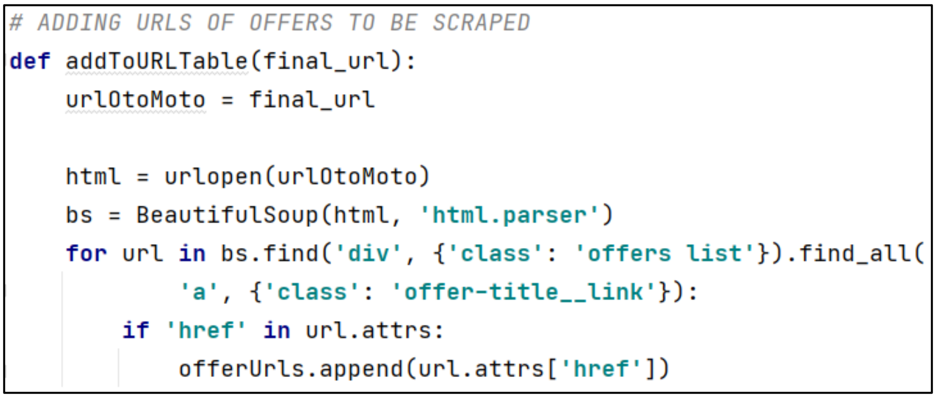
Source: https://otomoto.pl/bmw/seria-3 (accessed on March 1, 2021).

From the code presented in above Figure 3.28, it can be concluded that all offers on the list are in the section `(div)` with the class identifier "`offers list`". A link to a specific advertisement can be found in the `<a>` element of the "`offer-title__link`" class that is nested in the parent class "offers list". These are the boundary conditions that must be set to the extractor in the addToURLTable function discussed here.

After the object of the *BeautifulSoup* type is initialized, the loop begins, which works as follows:

- the boundary conditions that define the operation of the loop are set:
  - initially, the first <div> element identifiable by the "offers list" class via the find function is found;
  - then inside this class there is a need to obtain all <a> elements, which are identifiable by the class "offer-title__link" through the find_all function;
- the operation performed by the loop is checking if this element actually contains a link (i.e., has a href attribute), and if so, it is added to the offerUrls table.

After executing this function, the offerUrls table contains the necessary links to specific ads, so referring to it is legitimate from this moment on. It is worth paying attention to the fact that only links from the first page of advertisements are downloaded due to the fact that it prevents excessive data consumption on the server, which is also in line with the recommendations (Mitchell, 2015, p. 44). The operation of this function is presented in Figure 3.29.



```
# ADDING URLS OF OFFERS TO BE SCRAPED
def addToURLTable(final_url):
    urlOtoMoto = final_url

    html = urlopen(urlOtoMoto)
    bs = BeautifulSoup(html, 'html.parser')
    for url in bs.find('div', {'class': 'offers list'}).find_all(
            'a', {'class': 'offer-title__link'}):
        if 'href' in url.attrs:
            offerUrls.append(url.attrs['href'])
```

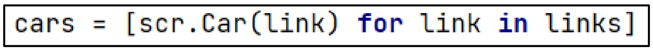
Figure 3.29. Function definition that adds the obtained offers' URLs to the array

Source: Own study.

After following all of the above steps, it can be proceeded to actual data extraction process from specific classifieds pages. Subchapter 3.3.5 shows the coding steps responsible for retrieving the information from the requested pages.

3.3.5. The process of the factual web scraping

This is where the process of proper data acquisition from specific OTOMOTO advertisements begins, the links of which are in the `offerUrls` table. The appropriate web scraper tool, which is defined in the *otoMotoScraper.py* file, is used for achieving this objective. The analysis of individual elements of this file will take place in the order to call specific functions or code fragments. The triggering of a loop creating a two-dimensional matrix with individual parameters of vehicles from offers starts the actual process of obtaining data from the websites. The matrix, which is assigned to a variable called “`cars`”, is created with the help of the Car class objects that are defined in the *otoMotoScraper.py* file. This process is illustrated in Figure 3.30.

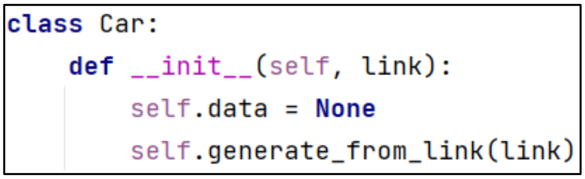


```
cars = [scr.Car(link) for link in links]
```

Figure 3.30. Adding Car objects to the new array by calling a loop activating the `Car` class constructors

Source: Own study.

When starting the process of creating an object of a given class, its constructor is called, i.e., the `__init__` function[49], which is run each time an object of a given class is created. It is also worth paying attention at this point to the fact that whenever a function is part of a class, it must refer to itself, or rather to an object that is created by itself. This is done through the “`self`” operator, which is only used to define the function, but is omitted when it is called. The constructor contains two elements: a data object, which will ultimately consist of the retrieved data from offers, and a call to the `generate_from_link` function, which is part of the Car class. The constructor of the newly created class `Car` looks as shown in Figure 3.31.



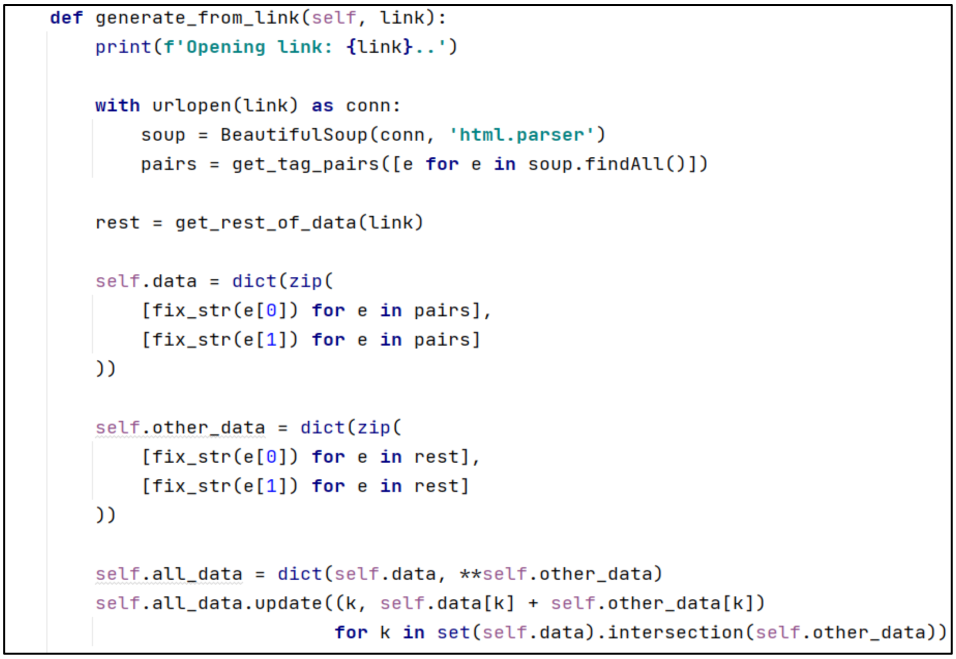
```
class Car:
    def __init__(self, link):
        self.data = None
        self.generate_from_link(link)
```

Figure 3.31. `Car` class constructor definition

Source: Own study.

[49] The constructor can take different names depending on the programming language. The name `__init__` is typical of the Python class constructors.

The `generate_from_link` function, which is called in the class constructor (as presented in Figure 3.31) and which belongs to the `Car` class[50], takes as argument the link passed when creating the `Car` object. Since the objects are created in a loop and assigned to the array, this function will be called multiple times – as many times as there are links in the `offerUrls` array, and thus how many pages with advertisements are to be analyzed. Hence, it was decided to inform the user about each rotation of the loop via the message "Opening link". The definition of the `generate_from_link` function is shown in Figure 3.32.



```python
def generate_from_link(self, link):
    print(f'Opening link: {link}..')

    with urlopen(link) as conn:
        soup = BeautifulSoup(conn, 'html.parser')
        pairs = get_tag_pairs([e for e in soup.findAll()])

    rest = get_rest_of_data(link)

    self.data = dict(zip(
        [fix_str(e[0]) for e in pairs],
        [fix_str(e[1]) for e in pairs]
    ))

    self.other_data = dict(zip(
        [fix_str(e[0]) for e in rest],
        [fix_str(e[1]) for e in rest]
    ))

    self.all_data = dict(self.data, **self.other_data)
    self.all_data.update((k, self.data[k] + self.other_data[k])
                         for k in set(self.data).intersection(self.other_data))
```

Figure 3.32. Definition of the `generate_from_link` function

Source: Own study.

As shown in Figure 3.32, several new functions have been invoked here, the operation of which will be demonstrated in the following paragraphs. Due to the fact that these are functions typical for the operation of a proper web harvester, it was decided to put their definitions into the *otoMotoScraper.py* file – not only for the ordering and aesthetics of the code, but also to avoid unnecessary references to an external file.

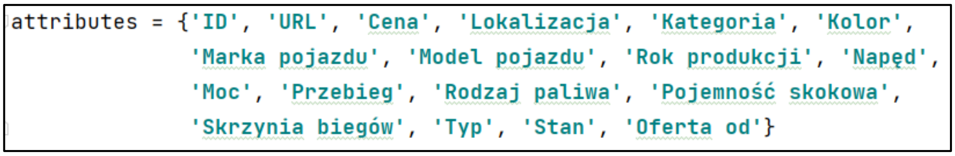
[50] This is visible through self-reference, that is, by using the "`self`" operator.

The *BeautifulSoup* library object is initialized as usual. This can be done using the “`with`” clause with an additional reference alias, as shown in Figure 3.32, however, the functionality is the same as for the initialization method presented earlier. In order to distinguish it (which has no effect on code performance and aesthetics), the name “`soup`” is used in this file for this object instead of “`bs`”.

The following attributes of the advertisement offer were assumed, which will be obtained from the website:

- offer ID,
- URL,
- price,
- location of the offer,
- category,
- color,
- car brand (make),
- car model,
- production year,
- drive type,
- engine power,
- mileage,
- type of fuel,
- engine displacement,
- type of gearbox,
- type of vehicle (car segment),
- the origin of the offer.

All of the above `attributes` have been assigned to a dictionary called attributes (in the order as shown above), which is defined by curly braces. Please note that the names of the attributes were named in Polish in the same format as it appears on the OTOMOTO website, due to the correct identification of the names later downloaded by the web scraper. The definition of the `attributes` dictionary in Python is shown in Figure 3.33.



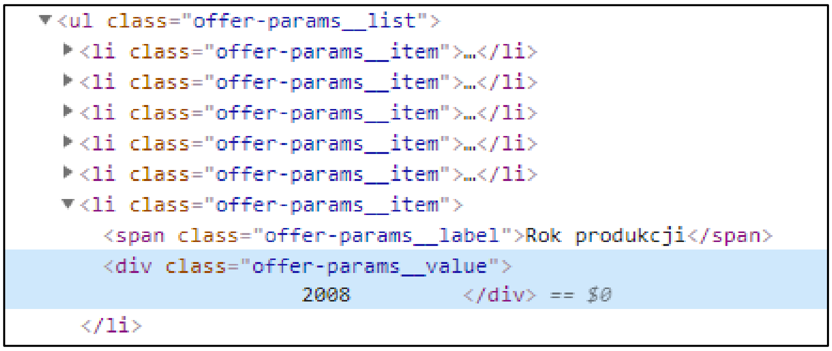
```
attributes = {'ID', 'URL', 'Cena', 'Lokalizacja', 'Kategoria', 'Kolor',
              'Marka pojazdu', 'Model pojazdu', 'Rok produkcji', 'Napęd',
              'Moc', 'Przebieg', 'Rodzaj paliwa', 'Pojemność skokowa',
              'Skrzynia biegów', 'Typ', 'Stan', 'Oferta od'}
```

Figure 3.33. Definition of the `attributes` dictionary

Source: Own study.

After analyzing the page of a specific vehicle advertisement on the OTOMOTO website, it was noticed that the necessary data is located in different places: the vehicle parameters (such as the year of production, engine capacity, etc.) are in the table under the photo gallery, while the rest of the data identifying the advertisement (such as ID or location) are in other parts of the site. Therefore, it was decided to create two functions that will work in different ways. Although the method of retrieving data in both methods uses the *BeautifulSoup* library, the method of finding and selecting data is slightly different.

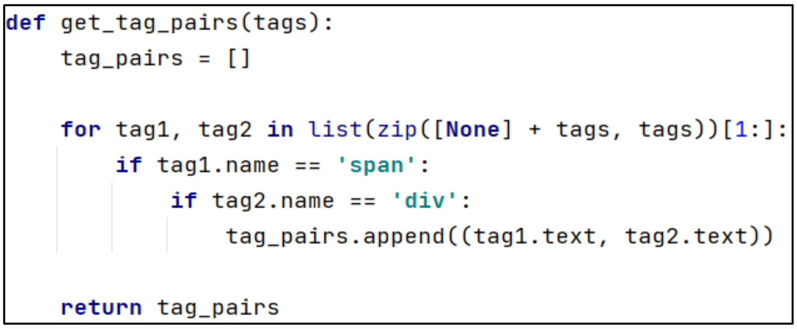
In the case of the vehicle data in the table, some regularity was noticed during the website analysis: the parameter name is placed directly before the value, which is reflected in the HTML code. Correspondingly: the name of the parameter is located in the `<span>` tag, while the value is located in the directly following (very importantly) `<div>` tag, as it is presented in Figure 3.34. It was decided to create a list of all pairs of consecutive tags: first `<span>`, then the following `<div>`. This was done with a loop running on a "list of pairs"[51], i.e., a quasi-two-dimensional matrix where the text values stored in the tags are joined together by a `zip` function. Pointing to these pairs of tags is done through a double and nested "`if`" condition, as shown in Figure 3.35. The text values are saved to the `tag_pairs` list returned by the function.



```
▼<ul class="offer-params__list">
  ▶<li class="offer-params__item">…</li>
  ▶<li class="offer-params__item">…</li>
  ▶<li class="offer-params__item">…</li>
  ▶<li class="offer-params__item">…</li>
  ▶<li class="offer-params__item">…</li>
  ▼<li class="offer-params__item">
      <span class="offer-params__label">Rok produkcji</span>
      <div class="offer-params__value">
                      2008          </div> == $0
    </li>
```

Figure 3.34. A fragment of the HTML code of the offer page responsible for the tabular display of the parameter list (below the photo gallery)

Source: https://otomoto.pl (accessed March 15, 2021).



```
def get_tag_pairs(tags):
    tag_pairs = []

    for tag1, tag2 in list(zip([None] + tags, tags))[1:]:
        if tag1.name == 'span':
            if tag2.name == 'div':
                tag_pairs.append((tag1.text, tag2.text))

    return tag_pairs
```

Figure 3.35. A function definition responsible for retrieving `<span>` and `<div>` tag pairs

Source: Own study.

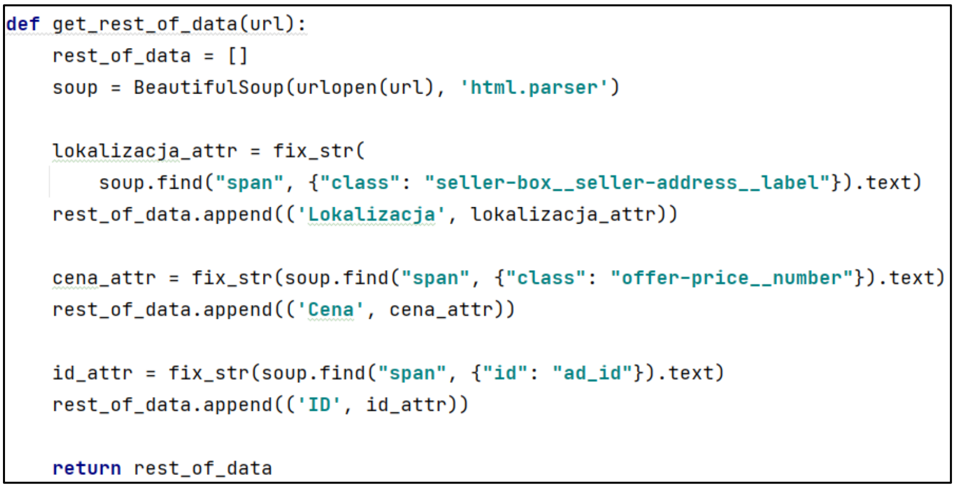
[51] As it is impossible to perform mathematical operations on the retrieved values (and we would not even need to), it was decided to use the list type, not the array type. In these iterations used, there is no difference in operation, but the list as a type is less cache-intensive.

Retrieving the rest of the data, i.e., ID, price, and location, is done in a more "conventional" way with the use of the `get_rest_of_data` function. It also uses the creation of the *BeautifulSoup* library object, which is assigned to the name "`soup`" as well. Finding objects on the page is done by the `find` function with imposed conditions that result from the analysis of the HTML code of the advertisement:

- for localization: text from the `<span>` tag with `class called` "`seller-box__seller-address__label`",
- `for price: text from the <span> tag with class called` "`offer-price__number`",
- for ID: text from the `<span>` tag with `id` called "`ad_id`".

It is preferable to refer to individual elements by an ID (as a tag attribute) that is unique to each element but need not be given. In case the ID is not available, reference should be made through other tag attributes. An analysis of the classifieds website showed that the class names that were used to identify the location and price are only present for these items, so indicating them via the class name does not cause errors and is correct.

The discussed function adds to the newly created quasi-two-dimensional list (called `rest_of_data`) the values collected in the process of scraping in this stage along with creating the field names. Then the function, the definition of which is shown in Figure 3.36, returns this list.

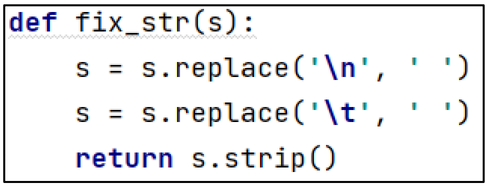


```python
def get_rest_of_data(url):
    rest_of_data = []
    soup = BeautifulSoup(urlopen(url), 'html.parser')

    lokalizacja_attr = fix_str(
        soup.find("span", {"class": "seller-box__seller-address__label"}).text)
    rest_of_data.append(('Lokalizacja', lokalizacja_attr))

    cena_attr = fix_str(soup.find("span", {"class": "offer-price__number"}).text)
    rest_of_data.append(('Cena', cena_attr))

    id_attr = fix_str(soup.find("span", {"id": "ad_id"}).text)
    rest_of_data.append(('ID', id_attr))

    return rest_of_data
```

Figure 3.36. Definition of a function that returns data from the advertisement page that are not in the parameter table

Source: Own study.

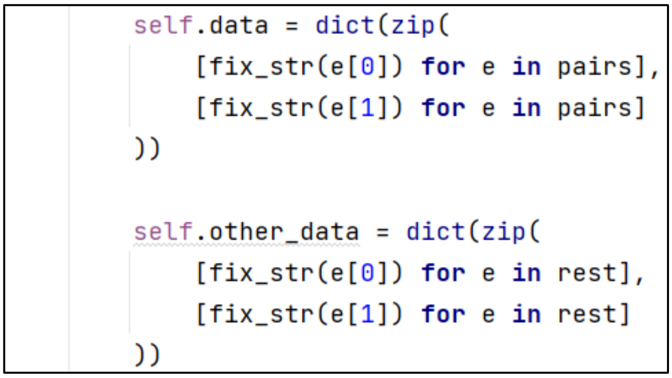
Accordingly, some of the data related to the announcement is in the `pairs` list (it has not yet been strained from the rest of the `<span> / <div> pairs`) and the rest is in the `rest` list. Then follows the definition of two dictionaries (`dict`) on which key operations will take place. When defining dictionaries, the data is initially "purged" by eliminating additional spaces or tabulations, because it has been noticed that they are embedded in original data. This cleansing of the information will allow for a more effective analysis of the collected data. This is done with the newly created `fix_str` function, which replaces unwanted strings with other strings with the `replace` function and removes redundant spaces with the `strip` function. The operation of the `fix_str` function is shown in Figure 3.37, while the creation of dictionaries of vehicle data (`self.data`) and other data (`self.other_data`), which take place in the `Car` constructor, is shown in Figure 3.38.



```
def fix_str(s):
    s = s.replace('\n', ' ')
    s = s.replace('\t', ' ')
    return s.strip()
```

Figure 3.37. Definition of the `fix_str` function

Source: Own study.

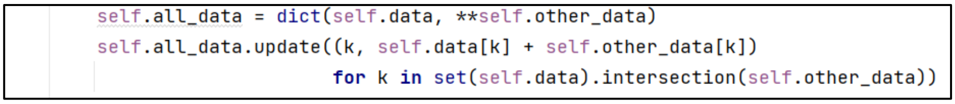


```
self.data = dict(zip(
    [fix_str(e[0]) for e in pairs],
    [fix_str(e[1]) for e in pairs]
))

self.other_data = dict(zip(
    [fix_str(e[0]) for e in rest],
    [fix_str(e[1]) for e in rest]
))
```

Figure 3.38. Creation of dictionaries in the `Car` constructor

Source: Own study.

At this point, there is a need to combine the above dictionaries into one that can be used for further operations. For this purpose, a `self.all_data` dictionary was created, consisting of all `<span> / <div>` pairs and joined parameters with values (i.e., ID, location, and price). The dictionary definition can be seen in the first line of Figure 3.39, while the actual process of combining two dictionaries into one is done using the update function (also seen in Figure 3.39). As an argument, a loop is given that combines the parameters and values of the dictionaries by a specific condition (namely: that the parameters should be assigned to the parameter section, and the values to the value section). This is done with the `intersection` function used on the `set`[52].



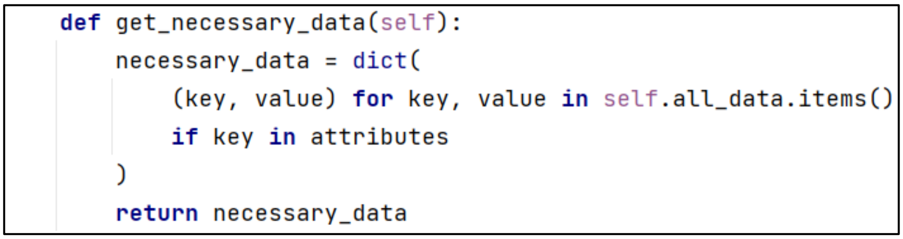
```
self.all_data = dict(self.data, **self.other_data)
self.all_data.update((k, self.data[k] + self.other_data[k])
                     for k in set(self.data).intersection(self.other_data))
```

Figure 3.39. Definition of the `all_data` dictionary along with entering the combined data into it

Source: Own study.

The constructor is finished with the above, so data dictionaries have been created for all links that are in the `offerUrls` array. It remains to extract the necessary data, i.e. that that has been defined in the attributes dictionary, from this dictionary (for each created object of the Car class). This is done using the `get_necessary_data` function that was defined in the `Car` class. Again, a dictionary (named `necessary_data`) is created, which is built from the items in `self.all_data`, which is next returned by the function. Such a multi-stage solution is a specific buffer for the further development of the program, if in the future there will be a desire to download other types of data from the OTOMOTO websites. Items are added to the dictionary (with the use of the "`if`" conditional) only when a key is in the `attributes` dictionary. The definition of this function is shown in Figure 3.40.

[52] Despite the slight differences between the dictionary and the set in Python language, it was decided to use the set here in order to be able to use the intersection function without any problems. However, `all_data` is still a dictionary by definition (`dict`), so type compatibility is preserved.

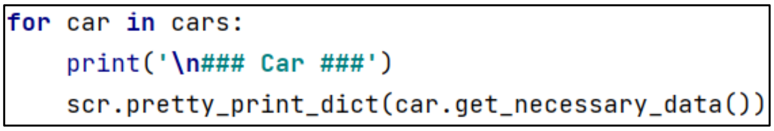


```python
def get_necessary_data(self):
    necessary_data = dict(
        (key, value) for key, value in self.all_data.items()
        if key in attributes
    )
    return necessary_data
```

Figure 3.40. Function creating a dictionary with only the necessary car data and offer

Source: Own study.

The `get_necessary_data` function is called in the *main.py* file immediately after creating an array with the `Car` objects. It is in a loop that displays the results to the user in a Python shell. The loop is responsible for displaying all parameters with values for a given offer, and individual offers are separated by a “`### Car ###`” delimiter, as seen in Figure 3.41.

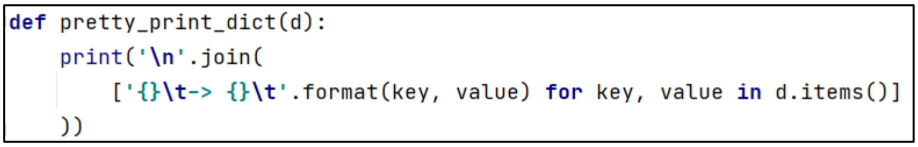


```python
for car in cars:
    print('\n### Car ###')
    scr.pretty_print_dict(car.get_necessary_data())
```

Figure 3.41. Loop displaying all offer data in the Python shell

Source: Own study.

In order to display the data more approachably (user-friendly), the newly created function `pretty_print_dict` was used, which takes the dictionary as an argument. It is responsible for adding a tab and an arrow separating the parameter name from the value, which translates into a more user-friendly look of the displayed results. The definition of this function is shown in Figure 3.42.



```python
def pretty_print_dict(d):
    print('\n'.join(
        ['{}\t-> {}\t'.format(key, value) for key, value in d.items()]
    ))
```

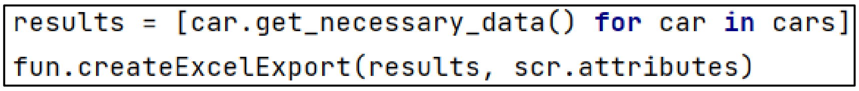
Figure 3.42. Function responsible for the accessible display of values

Source: Own study.

Having collected all the necessary parameters along with data values, we can proceed to the code software responsible for exporting to a spreadsheet. Subchapter 3.3.6 focuses on exporting acquired data to a Microsoft Excel file because of suitability of this form for carrying out further analyses.

3.3.6. Exporting scraped results to a Microsoft Excel spreadsheet

The last step of the program is to save the collected data to a Microsoft Excel spreadsheet. For this purpose, a function named `createExcelExport` designed for this project, which was defined in the *otoMotoFunctions.py* file, is called. As arguments this function takes a list of the `Car` objects with the collected data and a list (dictionary) of columns to be used in the export. The first argument is a filtered list of the `Car` type offers (created by a loop, filtered by the `get_necessary_data` function, and named "`results`"), while the dictionary of created attributes is used as the second argument. The `results` array creation as well as a call of the `createExcelExport function` are shown in Figure 3.43.



```
results = [car.get_necessary_data() for car in cars]
fun.createExcelExport(results, scr.attributes)
```

Figure 3.43. Creation of the `results` array and a call of the `createExcelExport` function

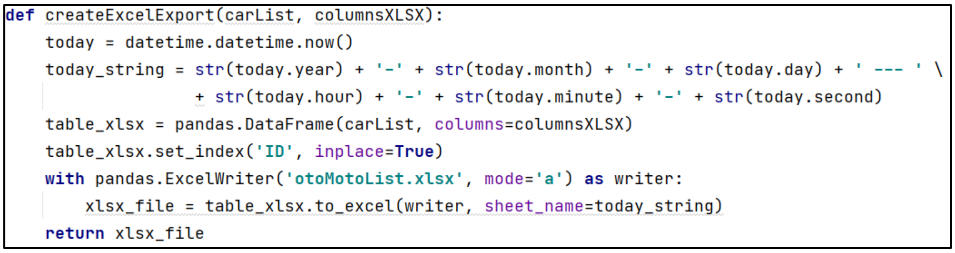
Source: Own study.

For a more convenient analysis of several searches by the user (by running the program several times), it was decided that the most appropriate solution would not be to create a new file each time but to add sheets in one file. This will allow us to more easily manipulate and switch between worksheets to reference the data in the whole spreadsheet. The Microsoft Excel file is named "*otoMotoList.xlsx*" and individual sheets will be appended to it.

In order to distinguish the sheets from one another, it was decided that each sheet would be named with the date and time of data collection. The date and time data is retrieved using the `datetime` library, and the variable `today` is assigned the called now function, which is responsible for the exact time when this function was run. The names of individual sheets will be created in the format "*YYYY-MM-DD --- HH-MM-SS*" and assigned to the `today_string` variable[53].

[53] The use of hyphens instead of periods or slashes in the entry of the sheet name results from the inability to use certain special characters in the names, which is imposed by the Microsoft Excel naming rules.

Creating sheets and adding data to an Excel file, as well as operating on XLSX files in Python, is most conveniently done using the `pandas` library. In order to start an operation with a Microsoft Excel file, we must create an object of the `DataFrame` class, which is part of the `pandas` library. The object is created by selecting a dataset (as a mandatory argument; in this case it is a car list) and defining the columns (via the so-called *kwargs*[54]; in this case it is an attribute list). The `DataFrame` object is stored in the `table_xlsx` variable. It is also important to set the index, i.e., the column identifying a given row in the table, which is done with the `set_index` function, which can be called on the DataFrame class objects[55]. In this case, the "*ID*" attribute that was used as an index of the table because of meeting all the criteria for the identifier of a given offer. The actual data entry into the spreadsheet is done with the use of the `ExcelWriter` type object (which is part of the `pandas` library), which in the constructor takes the name of the file on which the operation is performed as an argument. A suitable method of operating on this object is also the with clause along with a reference alias (in this case it is a `writer`). The `to_excel` function, which takes an `ExcelWriter` object as an argument, enters the data into the spreadsheet. Additionally, with `sheet_name` *kwarg*, a unique name is assigned to the sheet that will be newly created. Definition of the `createExcelExport` function is presented in Figure 3.44.



```python
def createExcelExport(carList, columnsXLSX):
    today = datetime.datetime.now()
    today_string = str(today.year) + '-' + str(today.month) + '-' + str(today.day) + ' --- ' \
                   + str(today.hour) + '-' + str(today.minute) + '-' + str(today.second)
    table_xlsx = pandas.DataFrame(carList, columns=columnsXLSX)
    table_xlsx.set_index('ID', inplace=True)
    with pandas.ExcelWriter('otoMotoList.xlsx', mode='a') as writer:
        xlsx_file = table_xlsx.to_excel(writer, sheet_name=today_string)
    return xlsx_file
```

Figure 3.44. Definition of the `createExcelExport` function

Source: Own study.

At this stage, the programming part is over, but the program remains to be tested, which is the subject of Subchapter 3.3.7. It also presents a sample data analysis based on the exported Microsoft Excel file. In this Chapter, the pro-

[54] *Kwargs* are in simplification optional arguments of a function that can be assigned some initial value when it is called.

[55] This can be simplified as using a primary key in database tables.

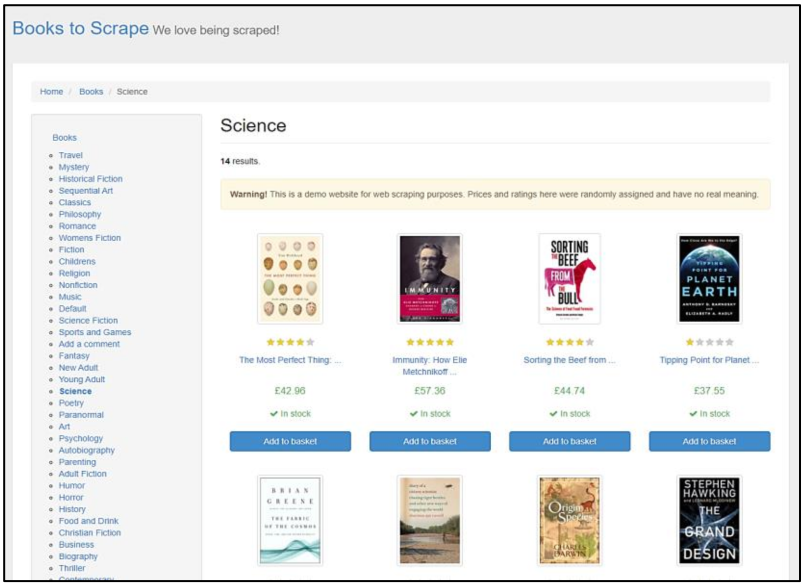
gram operation is also verified in the legal context, in accordance with the theoretical outline in Subchapter 2.3.2 and the OTOMOTO regulations quoted and analyzed in Chapter 3.1.2.

3.3.7. Program operation in practice with a sample analysis

In view of the legal doubts presented in Subchapter 3.1.2 and the position presented by Jaszewski (2018, p. 47), it was decided to present the results of the data extractor on the example of a website that was created for exercises with web scraping tools. The "Books to Scrape" website[56] is a non-profit "sandbox" website with a pseudo-online store, where it is possible to perform unlimited data downloading and further processing. This solution protects against legal consequences resulting from possible infringement of intellectual property or the legitimate interest of the OLX group, which owns OTOMOTO. Importantly, it is worth emphasizing at this point that the author of this study successfully tested this tool on the target OTOMOTO website, owing to a gateway resulting from the law of limited functionality of the tool on an insignificant part of the database in the private scope. The application tests were carried out with the greatest possible respect for the database administrator and with all ethical and legal considerations, as Krotov, Johnson, and Silva (2020) presented in their legal-ethical framework for web scraping (and which was also introduced in Subchapter 2.3.2).

There are many sites designed with the practice of data content mining in mind, however the choice was made for the "Books to Scrape" page, given its structure. Although it is not as advanced as the OTOMOTO website and the subject of the website (book) differs from the OTOMOTO website (cars), there are several common elements between them, bearing in mind the structure. There is a category tree on it, which can be simplified compared to the filters section of the OTOMOTO website. The pages of individual products also look similar: there is a general data section (including the title of the book and its price) next to the highlighted photo, while the detailed data section (including, among others, UPC identifier, availability or number of reviews) is also shown in a tabular manner. Such similarities in the structure made the presentation of the functioning of the presented program possible on this website, after adjusting the parts of the code responsible for indicating elements on the page. The page with the search results by category is shown in Figure 3.45, while the page for a specific book offer is in Figure 3.46.

[56] Website available publicly at: https://books.toscrape.com (accessed March 15, 2021)


Books to Scrape We love being scraped!

Home / Books / Science

Books

- Travel
- Mystery
- Historical Fiction
- Sequential Art
- Classics
- Philosophy
- Romance
- Womens Fiction
- Fiction
- Childrens
- Religion
- Nonfiction
- Music
- Default
- Science Fiction
- Sports and Games
- Add a comment
- Fantasy
- New Adult
- Young Adult
- **Science**
- Poetry
- Paranormal
- Art
- Psychology
- Autobiography
- Parenting
- Adult Fiction
- Humor
- Horror
- History
- Food and Drink
- Christian Fiction
- Business
- Biography
- Thriller

Science

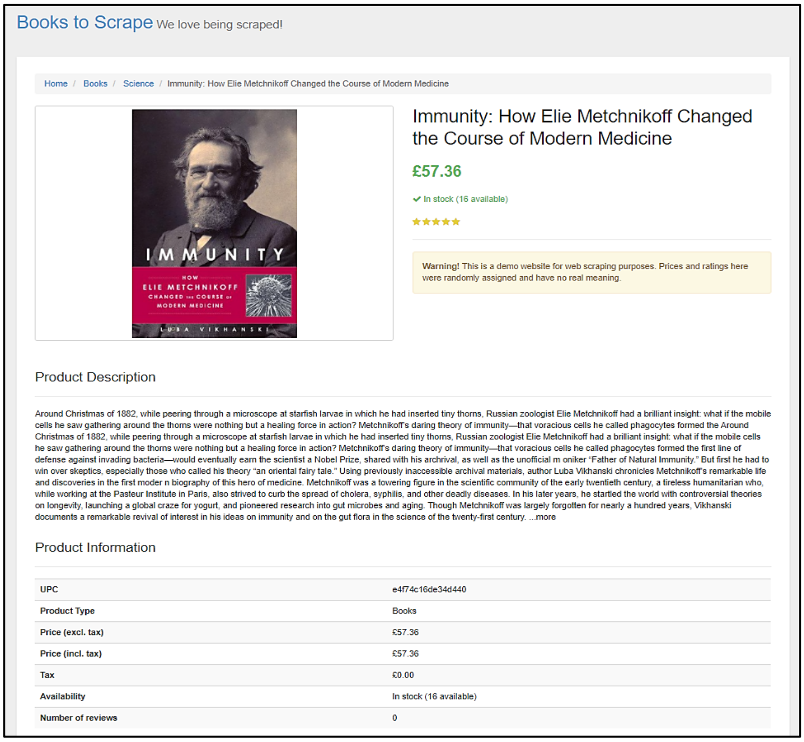
14 results.

Warning! This is a demo website for web scraping purposes. Prices and ratings here were randomly assigned and have no real meaning.

The Most Perfect Thing: ...	Immunity: How Elie Metchnikoff ...	Sorting the Beef from ...	Tipping Point for Planet ...
★★★★☆	★★★★★	★★★★☆	★☆☆☆☆
£42.96	£57.36	£44.74	£37.55
✓ In stock	✓ In stock	✓ In stock	✓ In stock
Add to basket	Add to basket	Add to basket	Add to basket

Figure 3.45. The Science category page of the "Books to Scrape" website with product listing

Source: Own study.


Books to Scrape We love being scraped!

Home / Books / Science / Immunity: How Elie Metchnikoff Changed the Course of Modern Medicine

Immunity: How Elie Metchnikoff Changed the Course of Modern Medicine

£57.36

✓ In stock (16 available)

★★★★★

Warning! This is a demo website for web scraping purposes. Prices and ratings here were randomly assigned and have no real meaning.

Product Description

Around Christmas of 1882, while peering through a microscope at starfish larvae in which he had inserted tiny thorns, Russian zoologist Elie Metchnikoff had a brilliant insight: what if the mobile cells he saw gathering around the thorns were nothing but a healing force in action? Metchnikoff's daring theory of immunity—that voracious cells he called phagocytes formed the Around Christmas of 1882, while peering through a microscope at starfish larvae in which he had inserted tiny thorns, Russian zoologist Elie Metchnikoff had a brilliant insight: what if the mobile cells he saw gathering around the thorns were nothing but a healing force in action? Metchnikoff's daring theory of immunity—that voracious cells he called phagocytes formed the first line of defense against invading bacteria—would eventually earn the scientist a Nobel Prize, shared with his archrival, as well as the unofficial m oniker "Father of Natural Immunity." But first he had to win over skeptics, especially those who called his theory "an oriental fairy tale." Using previously inaccessible archival materials, author Luba Vikhanski chronicles Metchnikoff's remarkable life and discoveries in the first moder n biography of this hero of medicine. Metchnikoff was a towering figure in the scientific community of the early twentieth century, a tireless humanitarian who, while working at the Pasteur Institute in Paris, also strived to curb the spread of cholera, syphilis, and other deadly diseases. In his later years, he startled the world with controversial theories on longevity, launching a global craze for yogurt, and pioneered research into gut microbes and aging. Though Metchnikoff was largely forgotten for nearly a hundred years, Vikhanski documents a remarkable revival of interest in his ideas on immunity and on the gut flora in the science of the twenty-first century. ...more

Product Information

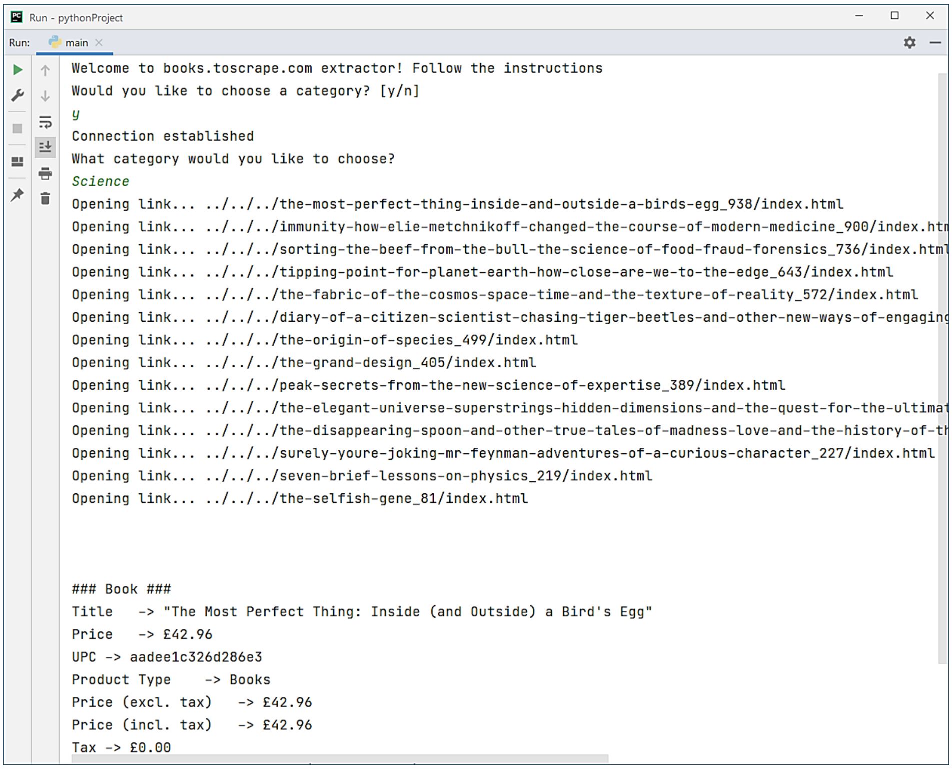
UPC	e4f74c16de34d440
Product Type	Books
Price (excl. tax)	£57.36
Price (incl. tax)	£57.36
Tax	£0.00
Availability	In stock (16 available)
Number of reviews	0

Figure 3.46. Sample product page on the "Books to Scrape" website

Source: Own study.

The following paragraphs present the method of the web scraping tool but adapted to the "Books To Scrape" page. However, it should be borne in mind that the sandbox page has a much less complicated structure than the discussed OTOMOTO; therefore, the possibilities of presenting the tool are limited.

Figure 3.47 shows the output of the web scraping program and the communication with a user. Program messages, including information about retrieving data from specific links, are shown here, as are requests to enter the input. Due to the very simplified filtering tool on the sandbox page, there is one question about the desire to narrow down the search and one request for input. In the case of the OTOMOTO website, communication with the customer is much more complex and comprehensive (in accordance with the assumed operation algorithm visualized in the process diagram in Figure 3.6); however, due to the previously mentioned legal doubts, it is not possible to show an example of operation on the landing page with automotive advertisements.

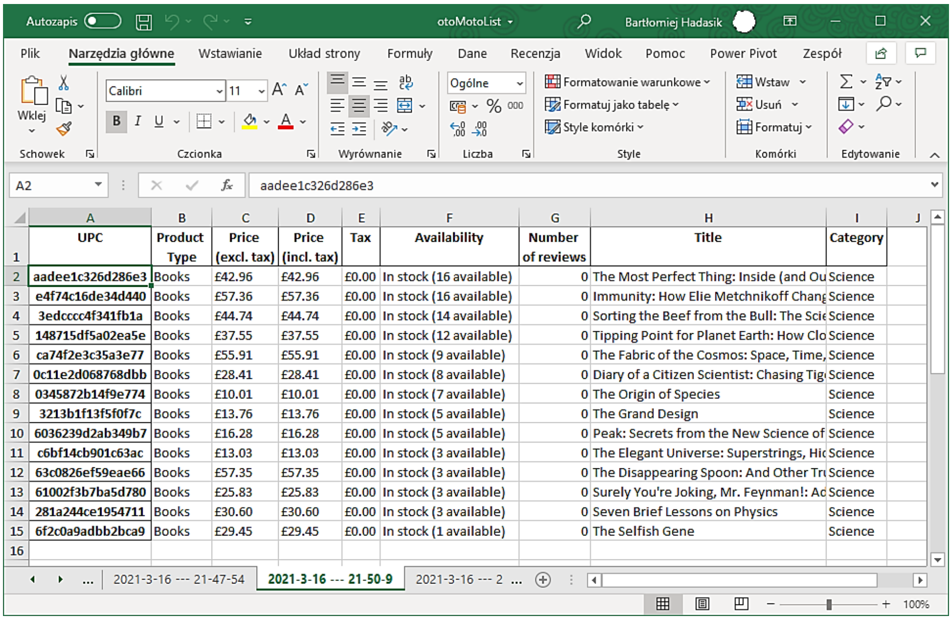


```
Welcome to books.toscrape.com extractor! Follow the instructions
Would you like to choose a category? [y/n]
y
Connection established
What category would you like to choose?
Science
Opening link... ../../../the-most-perfect-thing-inside-and-outside-a-birds-egg_938/index.html
Opening link... ../../../immunity-how-elie-metchnikoff-changed-the-course-of-modern-medicine_900/index.htm
Opening link... ../../../sorting-the-beef-from-the-bull-the-science-of-food-fraud-forensics_736/index.html
Opening link... ../../../tipping-point-for-planet-earth-how-close-are-we-to-the-edge_643/index.html
Opening link... ../../../the-fabric-of-the-cosmos-space-time-and-the-texture-of-reality_572/index.html
Opening link... ../../../diary-of-a-citizen-scientist-chasing-tiger-beetles-and-other-new-ways-of-engaging
Opening link... ../../../the-origin-of-species_499/index.html
Opening link... ../../../the-grand-design_405/index.html
Opening link... ../../../peak-secrets-from-the-new-science-of-expertise_389/index.html
Opening link... ../../../the-elegant-universe-superstrings-hidden-dimensions-and-the-quest-for-the-ultimat
Opening link... ../../../the-disappearing-spoon-and-other-true-tales-of-madness-love-and-the-history-of-th
Opening link... ../../../surely-youre-joking-mr-feynman-adventures-of-a-curious-character_227/index.html
Opening link... ../../../seven-brief-lessons-on-physics_219/index.html
Opening link... ../../../the-selfish-gene_81/index.html



### Book ###
Title   -> "The Most Perfect Thing: Inside (and Outside) a Bird's Egg"
Price   -> £42.96
UPC -> aadee1c326d286e3
Product Type    -> Books
Price (excl. tax)   -> £42.96
Price (incl. tax)   -> £42.96
Tax -> £0.00
```

Figure 3.47. Program operation (in the Python shell) – based on the "Books to Scrape" website

Source: Own study.

All data from the individual product pages of the “Books to Scrape” website is transferred to the spreadsheet in a Microsoft Excel file in an ordered, i.e., tabular form. Data from individual subpages is contained in separate lines of the sheet, identified by UPC, in accordance with the algorithmic assumptions. The exported data (output) in the Microsoft Excel spreadsheet is shown in Figure 3.48.



	UPC	Product Type	Price (excl. tax)	Price (incl. tax)	Tax	Availability	Number of reviews	Title	Category
2	aadee1c326d286e3	Books	£42.96	£42.96	£0.00	In stock (16 available)	0	The Most Perfect Thing: Inside (and Ou	Science
3	e4f74c16de34d440	Books	£57.36	£57.36	£0.00	In stock (16 available)	0	Immunity: How Elie Metchnikoff Chang	Science
4	3edcccc4f341fb1a	Books	£44.74	£44.74	£0.00	In stock (14 available)	0	Sorting the Beef from the Bull: The Scie	Science
5	148715df5a02ea5e	Books	£37.55	£37.55	£0.00	In stock (12 available)	0	Tipping Point for Planet Earth: How Clo	Science
6	ca74f2e3c35a3e77	Books	£55.91	£55.91	£0.00	In stock (9 available)	0	The Fabric of the Cosmos: Space, Time,	Science
7	0c11e2d068768dbb	Books	£28.41	£28.41	£0.00	In stock (8 available)	0	Diary of a Citizen Scientist: Chasing Tig	Science
8	0345872b14f9e774	Books	£10.01	£10.01	£0.00	In stock (7 available)	0	The Origin of Species	Science
9	3213b1f13f5f0f7c	Books	£13.76	£13.76	£0.00	In stock (5 available)	0	The Grand Design	Science
10	6036239d2ab349b7	Books	£16.28	£16.28	£0.00	In stock (5 available)	0	Peak: Secrets from the New Science of	Science
11	c6bf14cb901c63ac	Books	£13.03	£13.03	£0.00	In stock (3 available)	0	The Elegant Universe: Superstrings, Hic	Science
12	63c0826ef59eae66	Books	£57.35	£57.35	£0.00	In stock (3 available)	0	The Disappearing Spoon: And Other Tru	Science
13	61002f3b7ba5d780	Books	£25.83	£25.83	£0.00	In stock (3 available)	0	Surely You're Joking, Mr. Feynman!: Ad	Science
14	281a244ce1954711	Books	£30.60	£30.60	£0.00	In stock (3 available)	0	Seven Brief Lessons on Physics	Science
15	6f2c0a9adbb2bca9	Books	£29.45	£29.45	£0.00	In stock (1 available)	0	The Selfish Gene	Science

Figure 3.48. The exported data (output) in the Microsoft Excel spreadsheet

Source: Own study.

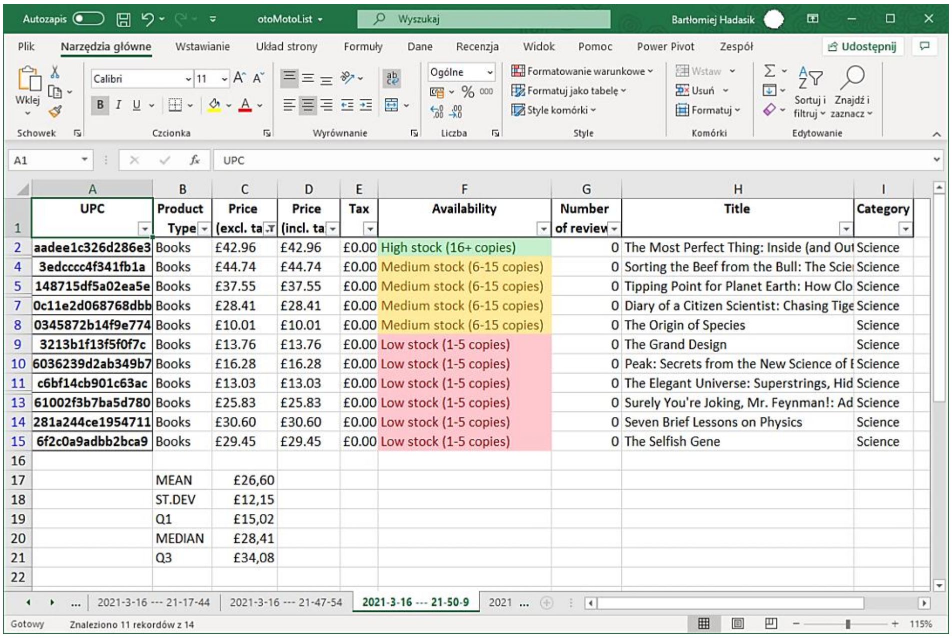
The data presented in this way can be further analyzed directly using the spreadsheet tools. It is worth underlining at this point, however, that the collected data set from the “sandbox” training platform is less extensive than in the case of downloading data from the OTOMOTO websites. As a result, the analyses must be carried out to a limited extent. Examples of data processing from the “Books to Scrape” website may be:

- the average gross price of a book in the “Science Fiction” category,
- median, quartiles or some percentiles of net prices of all books,
- standard deviation of the availability of books from the “Romance” category,
- the number of books with more than one review,
- segmentation of books from a given category in terms of availability into groups: small stock (1-5 copies), medium stock (6-15 copies), high stock (16+ copies).

Data grouped in this way can also be filtered using the spreadsheet tools:

- displaying books with high stock only,
- displaying books only from the "History" category (if the web scraping tool filter has not been applied),
- displaying books priced less than £50.

Basic analysis of the descriptive statistics was performed on the collected data set. An additional price filter was adopted (books cheaper than £50) and categories (books from the "Science" category). Below the data table are the values of the calculated basic descriptive statistics for the "price excluding tax" parameter: mean, standard deviation, median, and first and third quartiles. Conditional formatting was also used for the "accessibility" parameter. This analysis is presented in Figure 3.49.



	A	B	C	D	E	F	G	H	I
1	UPC	Product Type	Price (excl. ta	Price (incl. ta	Tax	Availability	Number of review	Title	Category
2	aadee1c326d286e3	Books	£42.96	£42.96	£0.00	High stock (16+ copies)	0	The Most Perfect Thing: Inside (and Out	Science
4	3edcccc4f341fb1a	Books	£44.74	£44.74	£0.00	Medium stock (6-15 copies)	0	Sorting the Beef from the Bull: The Scie	Science
5	148715df5a02ea5e	Books	£37.55	£37.55	£0.00	Medium stock (6-15 copies)	0	Tipping Point for Planet Earth: How Clo	Science
7	0c11e2d068768dbb	Books	£28.41	£28.41	£0.00	Medium stock (6-15 copies)	0	Diary of a Citizen Scientist: Chasing Tige	Science
8	0345872b14f9e774	Books	£10.01	£10.01	£0.00	Medium stock (6-15 copies)	0	The Origin of Species	Science
9	3213b1f13f5f0f7c	Books	£13.76	£13.76	£0.00	Low stock (1-5 copies)	0	The Grand Design	Science
10	6036239d2ab349b7	Books	£16.28	£16.28	£0.00	Low stock (1-5 copies)	0	Peak: Secrets from the New Science of E	Science
11	c6bf14cb901c63ac	Books	£13.03	£13.03	£0.00	Low stock (1-5 copies)	0	The Elegant Universe: Superstrings, Hid	Science
13	61002f3b7ba5d780	Books	£25.83	£25.83	£0.00	Low stock (1-5 copies)	0	Surely You're Joking, Mr. Feynman!: Ad	Science
14	281a244ce1954711	Books	£30.60	£30.60	£0.00	Low stock (1-5 copies)	0	Seven Brief Lessons on Physics	Science
15	6f2c0a9adbb2bca9	Books	£29.45	£29.45	£0.00	Low stock (1-5 copies)	0	The Selfish Gene	Science
16									
17		MEAN	£26,60						
18		ST.DEV	£12,15						
19		Q1	£15,02						
20		MEDIAN	£28,41						
21		Q3	£34,08						
22									

Figure 3.49. Sample analysis of the exported data in Microsoft Excel

Source: Own study.

In the case of data collected from the OTOMOTO website (which cannot be presented in this work due to ethical and legal doubts), analyses can be made on a larger and more diverse number of categories and in a much wider spectrum, using a larger number of statistical methods. Examples of the types of analysis that could be performed on this type of data set could be:

- the dominant of an engine displacement of a specific car model in a given price group,
- average mileage of vehicles until 2010,
- the number of cars with automatic transmission up to 2005,
- average engine power of Volkswagen cars from 2012,
- the number of Opel Astra cars from 2007-2010 with a diesel engine,
- median price of Toyota Yaris cars with a 1.3 liter gasoline engine,
- standard deviation of the prices of vehicles with a power from 100 HP to 150 HP in the years 2015-2019,
- the dominant color for cars not older than seven years with engine displacement up to 1500 cm^3,
- average prices of 4WD gasoline cars with automatic transmission,
- median of the year of production of cars imported from Germany with 4-wheel drive at a price not exceeding PLN 50,000.

Additionally, such filters for the following categories can be applied as:

- color,
- type of drive,
- type of gearbox,
- type of fuel.

This way, it is possible to create constrictions such as:

- private advertisements of red cars with automatic transmission,
- Honda CR-V cars with a power of at least 150 HP and four-wheel drive,
- Renault Clio, Opel Corsa together with Kia Rio cars from the 2015-2017 production years and price up to PLN 25,000,
- Volvo V60 gasoline cars with a mileage of up to 150,000 km for a price not higher than PLN 70,000.

Such filters can also be used after using defined filters for a given search (i.e., after the use of a web scraping program) and by means of inter-sheet calculations.

Having designed and written a web scraping program after analyzing the OTOMOTO website and taking into account several important contexts from the data mining environment, a few conclusions, which are worth noting, can be drawn. Conclusions along with a discussion and deliberation on the future of data extraction are described in the following final Chapter.

3.4. Discussion

The purpose of Chapter 3 was to construct a data extraction tool (using the web scraping approach), which was designed to alleviate information asymmetry. It was indeed constructed utilizing the Python programming language (as an executive portion), with the preceding use of conceptualization and visualization approaches (such a process diagram) as well as an examination of a specific website. The tool has been successfully evaluated both on the OTOMOTO target website (in a legally permissible private context with non--public results) as well as on a sandbox training site named "Books to Scrape", which is structurally comparable to the automobile classifieds website in a premium. This indicates that online scraping techniques may employed for proper and valuable data extraction on accessible databases or unordered sets different from each other.

Moreover, the research carried out in the fashion provided in Subchapter 3.3.7 demonstrates that the extraction of information is possible to be conducted with the assistance of unstructured data transformation by correctly aggregating it. Organizing data, with its accurate preceding download, is done in a straightforward manner utilizing the web scraping tool. The information gathered with its support, utilized appropriately and in the right environment, also accumulates enormous worth, which translates into improved judgments made by customers. Hence, for consumers (possible customers) having at their fingertips an assortment of data that may be subjected to elaborate computations depending on their preferences aids making a decision regarding a given purchase. For example: a buyer evaluating the typical price (or its median) for a car model in a specified range of the production year may identify whether a specific offer has an over- or understated pricing. Another example is a color analysis of a specific model: a customer who wants to calculate the dominant of a given color of a vehicle in a given price group (or taking other parameters into account) can see whether the particular offer with the car's dream color is unique or if there are many other advertisements similar to it. This may be viewed as the stabilizing of the information asymmetry phenomena, which consequently is the attainment of the work's purpose assumed at the very beginning.

A major adverse component of the online scraping approach was also observed, namely the requirement to alter the code to get data from a certain website. Data extraction, as explained in Chapter 2, may be loosely separated into two categories: web scraping and web crawling. The first procedure gathers data from certain websites depending on the parameters provided by the programmer (user), whereas the second one acquires data based on the rules of moving between pages, likewise imposed by the programmer. In this work, the first strategy was employed, and although the latter looks to be more dynamic, the restrictions imposed are also very inflexible and should be tailored to the original website from where the data collecting begins as well as the full crawling process. The only approaches to data extraction that do not require the intervention of a programmer in the process of tailoring the tool to a particular website are those that use machine learning or artificial intelligence (AI) in the case of activity prediction by the data acquisition program. Nonetheless, the process of transferring the online scraping tool code from this work (suited to the OTOMOTO ads portal) to the "Books to Scrape" website was short and did not involve any programming effort. The only changes that needed to be done were altering the target site, fixing the names and types for HTML selectors, and tweaking the program's interface with the user to enhance program's readability. As a conclusion, then, it may be argued that when we have a skeleton of a data retrieval tool adapted to a certain site, it can further be utilized, with modest modifications, on pages having a similar structure and functioning on a similar concept.

As previously stated, data content mining tools can extract a plethora of valuable information that can later be transformed into valuable knowledge; however, the external setting, which determines the avenues of using automated web harvesting tools, hinders the benefits of data extraction techniques. At this stage, particularly, it is required to analyze the formal and legal factors and the ensuing operational structure. The Act on the protection of databases in force in Poland enables the use of disseminated databases in a limited and private or scientific magnitude, but in the case of using automatic tools for "systematic and repeated" data retrieval, the law protects the database owner, protecting him against abuse of his "legitimate interests". By such language, the Act leads directly to the regulations of using the database, which are created by its administrator. When the database administrator explicitly reserves the right for third parties to use the database, whether through an automated scraping tool or otherwise, a customer who wants to gain a broader understanding of the exposed data (for example, through appropriate aggregation) does not have that option. The same applies in the case of scientific study, even if the scientist appropriately anonymized the collected data and presents it in a form that would not breach the rights of the data owner (which may, for example, include protecting it from hostile use by commercial rivalry). As a consequence, it can be stated that the relevant law is not favorable to the decrease of information asymmetry (in the context of future data processing that does not cause harm to the administrator), because customers have finite possibilities of employing the findings given in this way. Researchers and scientists, who have a duty to keep impartial towards the commercial sphere, also have a reasonably effective limited scope in the scientific area for dispersed data processing. It should be borne in mind that statutes and other legal actions have a higher legal force than corporate regulations, because regulations must be adapted to the applicable law, and not the other way around. On the other hand, however, as in the case of the Polish Act on the protection of databases, referring to lower-level normative acts (such as website regulations) and considering them as the basis of implementing the law in a given scenario is equally relevant. This is because it is in line with the Latin legal sentence *lex specialis derogat legi generali*, which translates as "a special law supersedes the general law". This principle, also used in Polish law, shows how important it is to comply with regulations that provide a framework for operating in a given area.

Conclusions

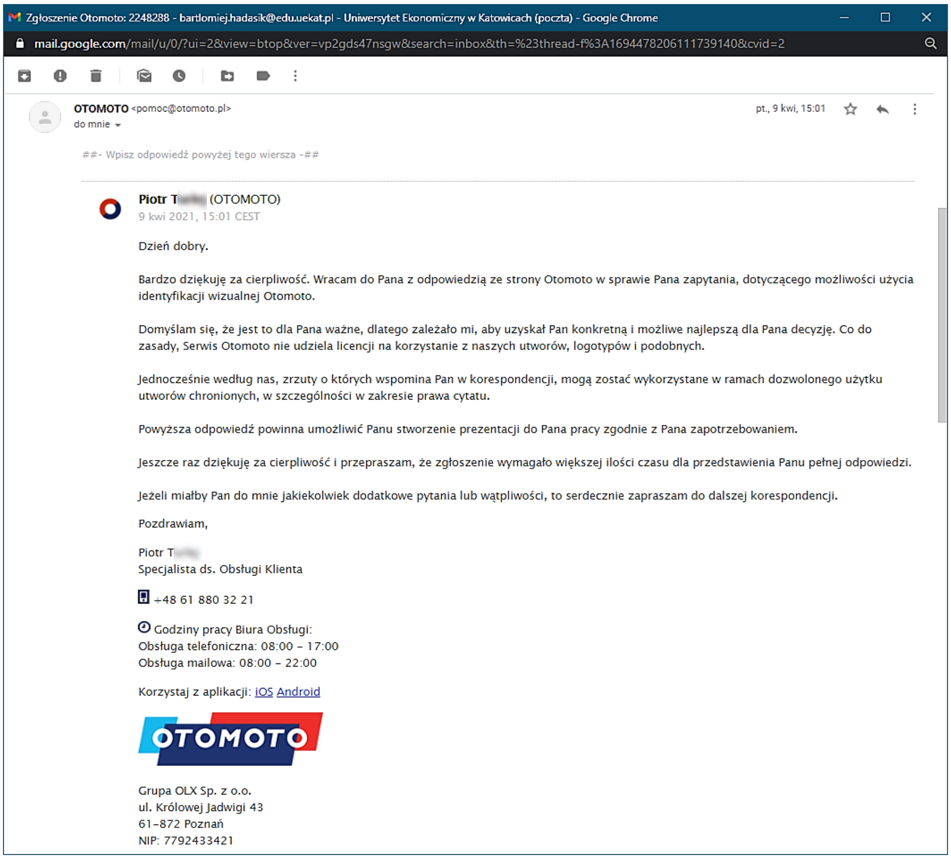
The amount of data in the Internet area is inconceivable, and although an enormous portion of it is classified or not publicly released, the data that can be accessible by everyone, therein using a web browser, is likewise indescribably numerous. The emergence of the information asymmetry problem outlined by Akerlof in 1970 took occurred far earlier before the commencement of the access to the public and freely available Internet network, but it was already noted then that not all market players had equal access to information. Despite its undeniable utility, the Internet has enabled its users to access information more easily; however, it should be pointed out that, in some ways, it has been able, in collaboration with legislators, to exacerbate the trend of increasing inequalities between the donor and the recipient of data. However, various internet tools have been built that obviously increase the convenience of access to data, such as public repositories or advertising portals, which has a favorable influence on the decision-making process. Advertising websites enable visitors learn more about a certain offer by encouraging merchants to share as much info as feasible. Nevertheless, the user of the ads portal has scarce access to additional data processing (via tailored, but private, analysis). Such studies might acquire even more significance not only for the whole gathered data (to a given extent) but also for a particular offer in the overall (macro) environment. Despite the fact that advertising portals by definition give access to data, the owner of such a portal, who maintains the data kept on it, may reserve specified possibilities of future use of it, because of the potentiality afforded by law.

Data extraction techniques, like online scraping, have proved themselves to be helpful in minimizing information asymmetry, giving users with the means to further analyze data and gain information from it. With respect to external factors, including the legislation currently in force to which technologies and their users have to conform, there is a hurdle to collecting value from the gathered information. Despite the fact that data is collected in good faith, reality reduces the legal acquisition and processing of data; however, the created tool is a kind of future expedient that can be fully used if the external environment can operate to the extent that allows its full and unconditional functioning. The perceived legal limits that impact the development of data extraction techniques may form the subject of additional scientific investigation. The creation of legislative proposals or the modification of existing legal provisions may be the subject of subsequent work. These measures would, on the one hand, permit users and scholars to make greater use of the data that has been made available and, on the other hand, would continue to safeguard the database owner from damaging actions on his property (such as, for instance, over--exploitation leading to server overloads).

The utilization of online harvesting techniques that may be employed in getting data from, among others, advertising websites, provides nearly unlimited possibilities. The profusion of data that is available in the public internet domain is not only a topic of interest to individual consumers, but is also becoming a possible study subject for scientists who seek to acquire a major value buried in it. Methods of acquiring data from the Internet, such as web scraping, which are able to extract a considerable quantity of information in a far quicker and simpler method than in the case of manual copying and pasting, come to the rescue. The present legal reality, in spite of that, requires them to be utilized in a way that does not infringe the rights of the data owner and in line with the database legislation. Despite the fact that online scraping techniques offer a vast spectrum of possibilities, it is acceptable in this instance to wait for changes in the environment that will allow them to exploit their potential and therefore make access to information by ordinary people totally equitable and symmetrical.

Appendix

All the screenshots related to the OTOMOTO advertising portal, which this work contains, were used under the right to quote, resulting directly from art. 29 of the Polish Act on Copyright and Related Rights (of 4 February, 1995). This is also confirmed in correspondence with an employee of the Customer Service Department of the OTOMOTO portal (belonging to the OLX group), who indicated that these screenshots can be used in this type of work under the aforementioned right to quote. An e-mail from an OTOMOTO employee of April 9, 2021 with the clarification of this issue is presented in Figure A.1.



Zgłoszenie Otomoto: 2248288 - bartlomiej.hadasik@edu.uekat.pl - Uniwersytet Ekonomiczny w Katowicach (poczta) - Google Chrome

mail.google.com/mail/u/0/?ui=2&view=btop&ver=vp2gds47nsgw&search=inbox&th=%23thread-f%3A1694478206111739140&cvid=2

OTOMOTO <pomoc@otomoto.pl>
do mnie

pt., 9 kwi, 15:01

##- Wpisz odpowiedź powyżej tego wiersza -##

Piotr T[redacted] (OTOMOTO)
9 kwi 2021, 15:01 CEST

Dzień dobry.

Bardzo dziękuję za cierpliwość. Wracam do Pana z odpowiedzią ze strony Otomoto w sprawie Pana zapytania, dotyczącego możliwości użycia identyfikacji wizualnej Otomoto.

Domyślam się, że jest to dla Pana ważne, dlatego zależało mi, aby uzyskał Pan konkretną i możliwe najlepszą dla Pana decyzję. Co do zasady, Serwis Otomoto nie udziela licencji na korzystanie z naszych utworów, logotypów i podobnych.

Jednocześnie według nas, zrzuty o których wspomina Pan w korespondencji, mogą zostać wykorzystane w ramach dozwolonego użytku utworów chronionych, w szczególności w zakresie prawa cytatu.

Powyższa odpowiedź powinna umożliwić Panu stworzenie prezentacji do Pana pracy zgodnie z Pana zapotrzebowaniem.

Jeszcze raz dziękuję za cierpliwość i przepraszam, że zgłoszenie wymagało większej ilości czasu dla przedstawienia Panu pełnej odpowiedzi.

Jeżeli miałby Pan do mnie jakiekolwiek dodatkowe pytania lub wątpliwości, to serdecznie zapraszam do dalszej korespondencji.

Pozdrawiam,

Piotr T[redacted]
Specjalista ds. Obsługi Klienta

+48 61 880 32 21

Godziny pracy Biura Obsługi:
Obsługa telefoniczna: 08:00 – 17:00
Obsługa mailowa: 08:00 – 22:00

Korzystaj z aplikacji: iOS Android

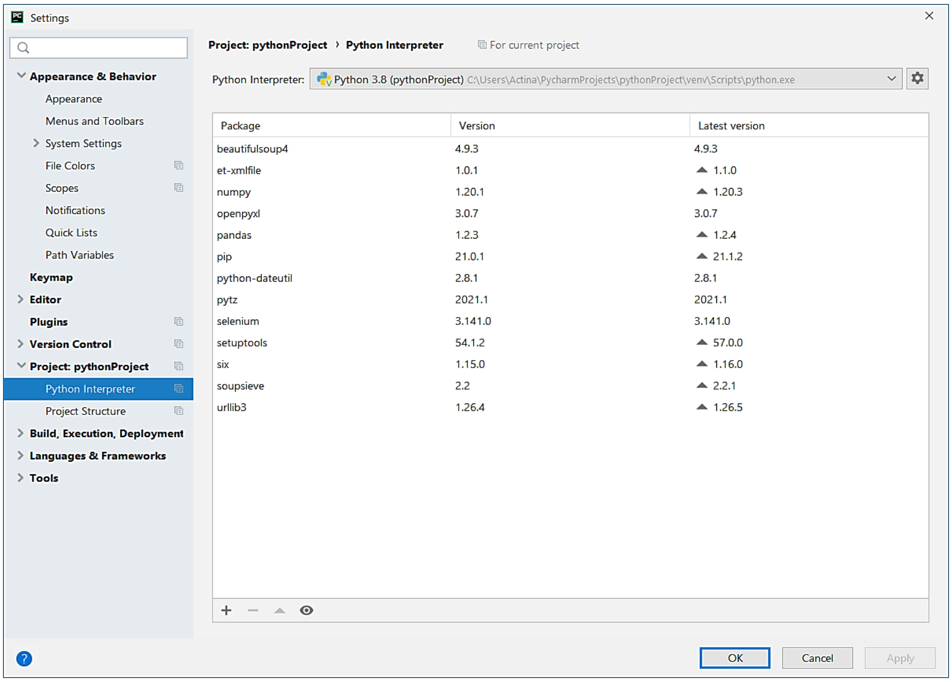
OTOMOTO

Grupa OLX Sp. z o.o.
ul. Królowej Jadwigi 43
61-872 Poznań
NIP: 7792433421

Figure A.1. An e-mail from an employee of the Customer Service Department of the OTOMOTO portal with an explanation of the use of screenshots of this website

Source: Own study (mail.google.com).

The figures below (Figure A.2 and A.3) represent the process of installing external libraries using the built-in PyCharm installer, which executes the pip installer commands in a windowed manner.



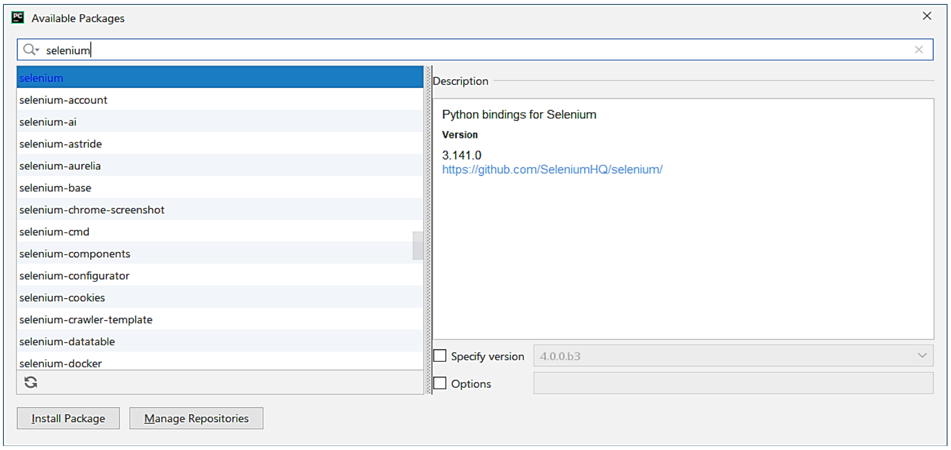
Project: pythonProject › Python Interpreter   For current project

Python Interpreter: Python 3.8 (pythonProject) C:\Users\Actina\PycharmProjects\pythonProject\venv\Scripts\python.exe

Package	Version	Latest version
beautifulsoup4	4.9.3	4.9.3
et-xmlfile	1.0.1	▲ 1.1.0
numpy	1.20.1	▲ 1.20.3
openpyxl	3.0.7	3.0.7
pandas	1.2.3	▲ 1.2.4
pip	21.0.1	▲ 21.1.2
python-dateutil	2.8.1	2.8.1
pytz	2021.1	2021.1
selenium	3.141.0	3.141.0
setuptools	54.1.2	▲ 57.0.0
six	1.15.0	▲ 1.16.0
soupsieve	2.2	▲ 2.2.1
urllib3	1.26.4	▲ 1.26.5

Figure A.2. Installed libraries pane for a given project in the PyCharm environment

Source: Own study.



selenium

selenium
selenium-account
selenium-ai
selenium-astride
selenium-aurelia
selenium-base
selenium-chrome-screenshot
selenium-cmd
selenium-components
selenium-configurator
selenium-cookies
selenium-crawler-template
selenium-datatable
selenium-docker

Description

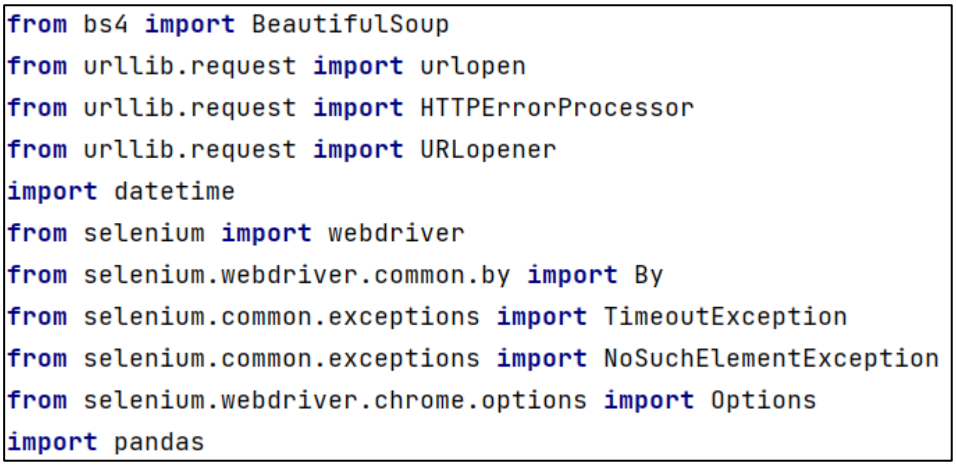
Python bindings for Selenium
Version
3.141.0
https://github.com/SeleniumHQ/selenium/

☐ Specify version 4.0.0.b3
☐ Options

Figure A.3. The window of installing external libraries for the Python language in the PyCharm environment

Source: Own study.

The following code snippet (Figure A.4) is responsible for importing libraries in the otoMotoFunctions.py file.



```python
from bs4 import BeautifulSoup
from urllib.request import urlopen
from urllib.request import HTTPErrorProcessor
from urllib.request import URLopener
import datetime
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.common.exceptions import TimeoutException
from selenium.common.exceptions import NoSuchElementException
from selenium.webdriver.chrome.options import Options
import pandas
```

Figure A.4. Import of libraries in the otoMotoFunctions.py file

Source: Own study.

References

Literature

Abid, A., Rahim, M., & Sheepers H. (2011). Experienced benefits and barriers of e-business technology adoption by SME suppliers. *Communications of the IBIMA*. 2011. http://www.ibimapublishing.com/journals/CIBIMA/2011/791778/791778.pdf

Ackoff, R. L. (1989). From data to wisdom. *Journal of Applied Systems Analysis*, *16*(1989), 3–9.

Ågerfalk, P. J., Fitzgerald, B., & Slaughter, S. A. (2009). Introduction to the special issue – Flexible and distributed information systems development: State of the art and research challenges. *Information Systems Research, 20*(3), 317–328. https://doi.org/10.1287/isre.1090.0244

Aguilar, J. A., Albert, A., Ameli, F., Anghinolfi, M., Anton, G., Anvar, S., ... & Battaglieri, M. (2007). The data acquisition system for the ANTARES neutrino telescope. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 570*(1), 107–116. https://doi.org/10.1016/j.nima.2006.09.098

Ahmad Sabri, I. A., Man, M., Abu Bakar, W. A. W., & Mohd Rose, A. N. (2019). Web data extraction approach for deep web using WEIDJ. *Procedia Computer Science, 163*, 417–426. https://doi.org/10.1016/j.procs.2019.12.124

Akerlof, G. (1970). The market for "lemons": Quality uncertainty and the market mechanism. *The Quarterly Journal of Economics*, *84*(3), 488–500. www.jstor.org/stable/1879431

Almousa, M. (2013). Barriers to e-commerce adoption: Consumers' perspectives from a developing country. *IBusiness*, *05*(02), 65–71. https://doi.org/10.4236/ib.2013.52008

Andress, J., & Winterfeld, S. (2011). Legal system impacts. In *Cyber Warfare* (pp. 207–223). Elsevier. https://doi.org/10.1016/b978-1-59749-637-7.00012-5

Andria, G., Attivissimo, F., Di Nisio, A., Lanzolla, A. M. L., & Pellegrino, A. (2016). Development of an automotive data acquisition platform for analysis of driving behavior. *Measurement, 93*, 278–287. https://doi.org/10.1016/j.measurement.2016.07.035

Arasu, A., & Garcia-Molina, H. (2003, June). Extracting structured data from web pages. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of data* (pp. 337–348).

Arendt, L. (2008). Barriers to ICT adoption in SMEs: How to bridge the digital divide?. *Journal of Systems and Information Technology, 10*(2), 93–108. https://doi.org/10.1108/13287260810897738

Bar-Ilan, J. (2001). Data collection methods on the web for infometric purposes: A review and analysis. *Scientometrics, 50*(1), 7–32. https://doi.org/10.1023/A:1005682102768

Bartuś, K., Batko, K., & Lorek, P. (2017). Diagnoza wykorzystania big data w organizacjach – wybrane wyniki badań, [Diagnose of using big data in organizations – selected results of research]. *Informatyka Ekonomiczna, 45*, 9–20. https://doi.org/10.15611/ie.2017.3.01

Bateson, G. (1972). *Steps to an ecology of mind: Collected essays in anthropology, psychiatry, evolution, and epistemology*. Jason Aronson Inc.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). *Manifesto for agile software development*. https://agilemanifesto.org/

Berners-Lee, T., & Fischetti, M. (1999). *Weaving the Web*. HarperOne.

Bhatt, G. D. (2001). Knowledge Management in organizations: Examining the interaction between technologies, techniques, and people. *Journal of Knowledge Management, 5*(1), 68–75. https://doi.org/10.1108/13673270110384419

Bolesta-Kukułka, K. (2003). *Decyzje menedżerskie*, [Managerial Decisions]. PWE.

Bolin, M., Webber, M., Rha, P., Wilson, T., & Miller, R. C. (2005). Automation and customization of rendered web pages. *Proceedings of the 18th annual ACM symposium on User interface software and technology, UIST '05*, 163–172. ACM.

Boronat, X. A. (2008, September). *A comparison of HTML-aware tools for Web Data extraction* [Master's thesis]. University of Leipzig.

Brown, R. (2018, July 31). The best programming languages for web scraping. In *YourStory*. https://yourstory.com/mystory/530e1cb78f-the-best-programming-l

Brown, S., & Hillegeist, S. A. (2007). How disclosure quality affects the level of information asymmetry. *Review of Accounting Studies, 12*(2–3), 443–477. https://doi.org/10.1007/s11142-007-9032-5

Buchanan, D., Fitzgerald, L., Ketley, D., Gollop, R., Jones, J. L., Lamont, S. S., Neath, A., & Whitby, E. (2005). No going back: A review of the literature on sustaining organizational change. *International Journal of Management Reviews, 7*(3), 189–205. https://doi.org/10.1111/j.1468-2370.2005.00111.x

Buckland, M. K. (1991). Information as thing. *Journal of the American Society for Information Science, 42*(5), 351–360. https://doi.org/10.1002/(sici)1097-4571(199106)42:5<351::aid-asi5>3.0.co;2-3

Carlsson, B. (2004). The digital economy: What is new and what is not?. *Structural Change and Economic Dynamics, 15*(3), 245–264. https://doi.org/10.1016/j.strueco.2004.02.001

Carnap, R. (1937). *Logical Syntax of language* (A. Smeaton, Trans.). Kegan Paul, Trench, Trubner & Co Ltd. (Original work published 1934)

Ceri, S., Bozzon, A., Brambilla, M., Della Valle, E., Fraternali, P., & Quarteroni, S. (2013). *Web information retrieval*. Springer Science & Business Media.

Chang, C. H., Kayed, M., Girgis, M. R., & Shaalan, K. F. (2006). A survey of web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering, 18*(10), 1411–1428. https://doi.org/10.1109/TKDE.2006.152

Chaulagain, R. S., Pandey, S., Basnet, S. R., & Shakya, S. (2017, November). Cloud based web scraping for big data applications. In *2017 IEEE International Conference on Smart Cloud (SmartCloud)* (pp. 138–143). IEEE.

Chen, G. L., Shu, S. B., & Ji, Z. S. (2009). Design and implementation of data acquisition system for power quality monitoring [J]. *Power System Protection and Control*, 3.

Chitura, T., Mupemhi, S., Dube, T., & Bolongkikit, J. (2008). Barriers to electronic commerce adoption in small and medium enterprises: A critical literature review. *Journal of Internet Banking and Commerce*, *13*(2), 1–13.

Chong, A. Y. L., Li, B., Ngai, E. W. T., Ch'ng, E., & Lee, F. (2016). Predicting online product sales via online reviews, sentiments, and promotion strategies. *International Journal of Operations & Production Management, 36*(4), 358–383. https://doi.org/10.1108/ijopm-03-2015-0151

Conrad, E., Misenar, S., & Feldman, J. (2014). Domain 9: Legal, Regulations, Investigations, and Compliance. In *Eleventh Hour CISSP* (pp. 155–170). Elsevier. https://doi.org/10.1016/b978-0-12-417142-8.00009-1

Conrath, D. (1967). Organizational decision-making behavior under varying conditions of uncertainty. *Management Science, 13*, B487–B500.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory*. Wiley.

Davies, B. (2020, August 24). Is web scraping legal in 2020?. *ScrapeDiary*. https://scrapediary.com/is-web-scraping-legal/

de S Sirisuriya, S. (2015, November). A comparative study on web scraping. *Proceedings of 8th International Research Conference*.

Détienne, F. (2002). What is a computer program?. In *Practitioner Series* (pp. 13–20). Springer. https://doi.org/10.1007/978-1-4471-0111-6_2

Dewi, L. C., Meiliana, & Chandra, A. (2019). Social media web scraping using social media developers API and Regex. *Procedia Computer Science, 157*, 444–449. https://doi.org/10.1016/j.procs.2019.08.237

Donaldson, T., Werhane, P. H., & Cording, M. (1983). *Ethical issues in business* (pp. 153–165). New Jersey.

DiLorenzo, T. (2011). A note on the Canard of "Asymmetric information" as a source of market failure. *The Quarterly Journal of Austrian Economics, 14*(2), 249–255.

Dingsøyr, T., Dybå, T., & Abrahamsson, P. (2008). A preliminary roadmap for empirical research on agile software development. *Agile 2008 Conference*. https://doi.org/10.1109/agile.2008.50

Dogucu, M., & Çetinkaya-Rundel, M. (2020). Web scraping in the statistics and data science curriculum: Challenges and opportunities. *Journal of Statistics Education,* 1–11. https://doi.org/10.1080/10691898.2020.1787116

Draper, P., & Paudyal, K. (2008). Information asymmetry and bidders' gains. *Journal of Business Finance & Accounting, 35*(3–4), 376–405. https://doi.org/10.1111/j.1468-5957.2008.02082.x

Dreyer, A. J., Stockton, J. (2013). Internet 'data scraping': A primer for counseling clients. *New York Law Journal*. https://www.law.com/newyorklawjournal/almID/1202610687621

Duan, X., Deng H., Corbitt, B. (2012). Evaluating the critical determinants for adopting e-market in Australian small-and-medium sized enterprises. *Management Research Review, 35*(3-4), 289–308. https://doi.org/10.1108/01409171211210172

Dwivedi, Y. K., Hughes, D. L., Coombs, C., Constantiou, I., Duan, Y., … & Upadhyay, N. (2020). Impact of COVID-19 pandemic on information management research and practice: Transforming education, work and life. *International Journal of Information Management*, 102211. https://doi.org/10.1016/j.ijinfomgt.2020.102211

Eriksson, J. O. (2016). *Evaluation of webscraping tools for creating an embedded webwrapper* [Master's thesis]. KTH Royal Institute of Technology, School of Computer Science and Communication.

Erl, T., Khattak, W., & Buhler, P. (2015). *Big data fundamentals: Concepts, drivers & techniques*. Prentice Hall.

Fedak, V. (2018, February 9). Big data: What is web scraping and how to use it. *Towards Data Science*. https://towardsdatascience.com/big-data-what-is-web-scraping-and-how-to-use-it-74e7e8b58fd6

Fernández-Villamor, J. I., Blasco-García, J., Iglesias, C. A., & Garijo, M. (2011, January). A semantic scraping model for web resources – Applying linked data to web page screen scraping. In *International Conference on Agents and Artificial Intelligence* (Vol. 2, pp. 451-456). SCITEPRESS.

Fernández-Villamor, J. I., Iglesias, C. A., & Garijo, M. (2014). A framework for goal--oriented discovery of resources in the RESTful architecture. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 44*(6), 796–803. https://doi.org/10.1109/tsmcc.2013.2259231

Ferrara, E., De Meo, P., Fiumara, G., & Baumgartner, R. (2014). Web data extraction, applications and techniques: A survey. *Knowledge-Based Systems, 70*, 301–323. https://doi.org/10.1016/j.knosys.2014.07.007

Freedman, S., & Jin, G. Z. (2011). Learning by doing with asymmetric information: Evidence from Prosper.com. *National Bureau of Economic Research*, Working Paper no. 16855. https://doi.org/10.3386/w16855

Gajewski, J.-F., & Li, L. (2015). Can Internet-based disclosure reduce information asymmetry? *Advances in Accounting, 31*(1), 115–124. https://doi.org/10.1016/j.adiac.2015.03.013

Gibbs, J., Kraemer, K. L., & Dedrick, J. (2003). Environment and policy factors shaping global e-commerce diffusion: A cross-country comparison. *The Information Society, 19*(1), 5–18. https://doi.org/10.1080/01972240309472

Glez-Peña, D., Lourenço, A., López-Fernández, H., Reboiro-Jato, M., & Fdez-Riverola, F. (2013). Web scraping technologies in an API world. *Briefings in Bioinformatics, 15*(5), 788–797. https://doi.org/10.1093/bib/bbt026

Goldfein, S., & Keyte, J. (2017). Big data, web 'scraping' and competition law: The debate continues. *New York Law Journal, 258*(49), 1.

Goes, P. (2014). Editor's comments: Big data and IS research. *MIS Quarterly, 38*(3), iii–viii.

Griffin, R. W. (2006). *Podstawy zarządzania organizacjami* [Management]. PWN.

Guertin, J. D. (1996). What is sustainability?. *Tunnelling and Underground Space Technology, 11*(4), 373–375.

Gumah, M. E., & Jamaluddin, Z. (2006). What is the digital economy, and how to measure it. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=bb1a7286179585171a4f25d51d1f379dc69a31ad

Hadasik, B. (2019). *RODO w e-commerce na podstawie sklepu internetowego Under Muscle*, [*GDPR in e-commerce according to Under Muscle online store*] [Bachelor's thesis]. University of Economics in Katowice. https://doi.org/10.5281/zenodo.4486068

Hadasik, B. (2020). Analiza makroekonomiczna sektora e-commerce w obliczu pandemii COVID-19 z użyciem metody PEST/PESTEL. In M. Bogusz, M.Wojcieszak, & P. Rachwał (Eds.), *Poszerzamy Horyzonty.* (20th ed., pp. 64–77). Mateusz Weiland Network Solutions. https://doi.org/10.5281/zenodo.4395725

Haddaway, N. (2015). The use of web-scraping software in searching for grey literature. *Grey Journal, 11*, 186–190.

Harford, T. (2005). *Undercover Economist*. Oxford University Press.

Heisig, P., Ogaza, M. A., & Hamraz, B. (2020). Information and knowledge assessment – Results from a multinational automotive company. *International Journal of Information Management, 54*, 102137. https://doi.org/10.1016/j.ijinfomgt.2020.102137

Hey, J. (2004). The data, information, knowledge, wisdom chain: The metaphorical link. *Intergovernmental Oceanographic Commission, 26*, 1–18.

Hicks, S., & Irizarry, R. (2018). A guide to teaching data science. *The American Statistician, 72*, 382–391. https://doi.org/10.1080/00031305.2017.1356747

Hilbert, M. (2016). Formal definitions of information and knowledge and their role in growth through structural change. *Structural Change and Economic Dynamics, 38*, 69–82. https://doi.org/10.1016/j.strueco.2016.03.004

Hillen, J. (2019). Web scraping for food price research. *British Food Journal, 121*(12), 3350–3361. https://doi.org/10.1108/bfj-02-2019-0081

Hilty, B. (2010, June 13-17). Transforming data into knowledge: Defining the six steps of information management [Conference paper presentation]. *2010 National Farm Management Conference*.

Himmi, K., Arcondara, J., Guan, P., & Zhou, W. (2017). Value oriented Big data strategy: Analysis & case study. *Proceedings of 50th Hawaii International Conference on System Sciences*.

Horton, N. J., Baumer, B. S., & Wickham, H. (2015). Taking a chance in the classroom: Setting the stage for data science: Integration of data management skills in introductory and second courses in statistics. *Chance, 28*(2), 40–50. https://doi.org/10.1080/09332480.2015.1042739

Hurwicz, L. (1951). The generalized bayes minimax principle: A criterion for decision making under uncertainty. *Discussion Paper Statistics, 335*, Cowles Commission.

Hurwicz, L. (1952). A criterion for decision making under uncertainty. *Technical Report, 355*, Cowles Commission.

Jaremen, D., & Nawrocka, E. (2015). Asymetria informacji na rynku usług hotelarskich. *Prace Naukowe Uniwersytetu Ekonomicznego We Wrocławiu,* 379. https://doi.org/10.15611/pn.2015.379.39

Jaszewski, M. (2018). *Generyczny system do pobierania danych z portali internetowych* [Master's thesis]. Polsko-Japońska Akademia Technik Komputerowych.

Kajtazi, M. (2010). Information asymmetry in the digital economy. In C. A. Shoniregun, (Ed.), *Proceedings of the IEEE International Conference on Information Society (i-Society 2010)* (pp. 148–155). http://urn.kb.se/resolve?urn=urn:nbn:se:lnu:diva-7323

Kalinić, Z. (2014). Barriers to higher and faster adoption of e-commerce [Paper presentation]. *3rd International Scientific Conference Contemporary Issues in Economics, Business and Management – EBM 2014*. Faculty of Economics, University of Kragujevac.

Kamiński, F. (2003). Powszechna usługa telekomunikacyjna w Unii Europejskiej i Polsce. *Telekomunikacja i Techniki Informacyjne, 1-2/2003*.

Kapurubandara, M., & Lawson, R. (2006). Barriers to adopting ICT and e-commerce with SMEs in developing countries: An exploratory study in Sri Lanka. *Proceedings of the 2006 Collector Conference on Electronic Commerce (CollECTeR '06)*.

Kaspa, L. P., Akella, V. N. S. S., Chen, Z., & Shi, Y. (2018). Towards extended data mining: An examination of technical aspects. *Procedia Computer Science, 139*, 49–55. https://doi.org/10.1016/j.procs.2018.10.216

Kasper, W., & Streit, M. E. (1999). *Institutional economics: Social order and public policy*. E. Elgar Publishing.

Khalil, S., & Fakir, M. (2017). RCrawler: An R package for parallel web crawling and scraping. *SoftwareX, 6*, 98–106. https://doi.org/10.1016/j.softx.2017.04.004

Kinne, J., & Axenbeck, J. (2018). Web mining of firm websites: A framework for web scraping and a pilot study for Germany. *ZEW-Centre for European Economic Research Discussion Paper*, (18–033).

Kling, R., & Lamb, R. (1999). IT and organizational change in digital economies: A socio-technical approach. *Computers and Society, 29*(3), 17–25. https://doi.org/10.1145/572183.572189

Knight, F. H. (1921). *Risk, uncertainty, and profit*. Houghton Mifflin.

Kool, L., van Veenstra, A. F., Rumpf, G., & Chernovich, E. (2011). Interim Report 1: Barriers to eCommerce and Trustmarks Inventory. *EU online Trustmarks – Building Digital Confidence in Europe*. SMART 2011/0022.

Kosala, R., & Blockeel, H. (2000). Web mining research: A survey. *ACM SIGKDD Explorations Newsletter, 2*(1), 1–15. https://doi.org/10.1145/360402.360406

Kozak, J., Kania, K., & Juszczuk, P. (2020). Permutation entropy as a measure of information gain/loss in the different symbolic descriptions of financial data. *Entropy, 22*(3), 330. https://doi.org/10.3390/e22030330

Krijnen, D., Bot, R., & Lampropoulos, G. (2014). *Automated Web Scraping APIs*. http://mediatechnology.leiden.edu/images/uploads/docs/wt2014_web_scraping.pdf

Krotov, V., & Silva, L. (2018). Legality and ethics of web scraping. *Twenty-fourth Americas Conference on Information Systems*. New Orleans.

Krotov, V., & Tennyson, M. (2018). Research note: Scraping financial data from the web using the R Language. *Journal of Emerging Technologies in Accounting, 15*(1), 169–181. https://doi.org/10.2308/jeta-52063

Krotov, V., Johnson, L. R., & Silva, L. (2020). Tutorial: Legality and ethics of web scraping. *Communications of the Association for Information Systems, 47*(30), 555–581.

Kulkarni, S. P. (2000). The influence of information technology on information asymmetry in product markets. *The Journal of Business and Economic Studies, 6*(1), 55–71. Retrieved November 21, 2021, from https://search.proquest.com/docview/235804115

Kullback, S., & Leibler, R. (1951). On information and sufficiency. *The Annals of Mathematical Statistics, 22*(1), 79–86. http://www.jstor.org/stable/2236703

Kubiczek, J. (2018). Identyfikacja potrzeb wdrażania innowacji w sektorze polskiego górnictwa. In E. Staniewska (Red.), *Potencjał innowacyjny w inżynierii produkcji i technologii materiałów* (pp. 77–80). Wydawnictwo Wydziału Inżynierii Produkcji i Technologii Materiałów Politechniki Częstochowskiej.

Kubiczek, J. (2019). Statystyczna analiza decyzji jako metoda wspomagania podejmowania decyzji w warunkach niepewności. *Nauka, Badania i Doniesienia Naukowe 2019, Nauki techniczne i ścisłe część II* (pp. 86–96).Idea Knowledge Future.

Kubiczek, J., Derej, W., & Hadasik, B. (2021). Virtualization of Poles' buying behavior during the COVID-19 pandemic. *Academic Review of Business and Economics, 1*(1), 31–43. https://doi.org/10.22367/arbe.2021.01.03

Kushmerick, N. (1997). *Wrapper induction for information extraction* [PhD thesis]. University of Washington.

Labadie, C., Eurich, M., & Legner, C. (2020, March). Empowering data consumers to work with data: Data documentation for the enterprise context. *5th International Conference on Wirtschaftsinformatik*.

LaValle, S., Lesser, E., Shockley, R., Hopkins, M. S., & Kruschwitz, N. (2011). Big data, analytics and the path from insights to value. *MIT Sloan Management Review, 52*(2), 21.

Lawson, R. (2015). *Web scraping with Python*. Packt Publishing Ltd.

Lipshitz, R., & Strauss, O. (1997). Coping with uncertainty: A naturalistic decision--making analysis. *Organizational Behavior and Human Decision Processes, 69*(2), 149–163. https://doi.org/10.1006/obhd.1997.2679

Liu, C., Belkin, N. J., & Cole, M. J. (2012). Personalization of search results using interaction behaviors in search sessions. *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval – SIGIR '12*. https://doi.org/10.1145/2348283.2348314

Lodge, M., & Boin, A. (2020). COVID-19 as the ultimate leadership challenge: Making critical decisions without enough data. British and Irish Politics and Policy.

Mach, M. A., & Owoc, M. L. (2001, June). Validation as the integral part of a knowledge management process. In *Proceeding of Informing Science Conference* (pp. 346–351).

Mach, M. A., & Owoc, M. L. (2010). Knowledge granularity and representation of knowledge: Towards knowledge grid. In *Intelligent Information Processing V* (pp. 251–258). Springer. https://doi.org/10.1007/978-3-642-16327-2_31

Mach-Król, M. (2017). Big data analytics in Polish companies—selected research results. *ICT Management for Global Competitiveness and Economic Growth in Emerging Economies (ICTM)*, 85.

Maheedharan, V. (2016, November 11). A detailed overview of web crawlers. In *Cabot Solutions*. https://www.cabotsolutions.com/2016/11/a-detailed-overview-of-web-crawlers/

Makowiec, M. (2008). Przeobrażenia w funkcjonowaniu przedsiębiorstw uwarunkowane technologią teleinformatyczną. *Zeszyty Naukowe Uniwersytetu Ekonomicznego w Krakowie, 765*, 141–155.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. H. (2011). *Big data: The next frontier for innovation, competition, and productivity*. McKinsey Global Institute.

Mao, J., Liu, Y., Luan, H., Zhang, M., Ma, S., Luo, H., & Zhang, Y. (2017, August 7). Understanding and predicting usefulness judgment in web search. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '17: The 40th International ACM SIGIR conference on research and development in Information Retrieval*. https://doi.org/10.1145/3077136.3080750

Martyniak, M. (2015). Strona internetowa jako narzędzie sprzedaży i promocji oferty na rynku nieruchomości. In M. Woźniak, Ł. B. Pilarz, M. Drewniak (Eds.), *Polscy doktorzy i doktoranci w rozwoju światowej myśli naukowej* (pp. 21–30). Mateusz Weiland Network Solutions,

McFarland, R. K., Guertin, J. D., & Pelizza, S. (1996). North American Tunneling '96: What is sustainability? *Tunnelling and Underground Space Technology*, *11*(4), 373–375. https://doi.org/10.1016/s0886-7798(96)00037-5

Merrell, R. C. (2002). Trespass to chattels in the age of the internet. *Washington University Law Review, 80*(2), 675–703.

Mesenbourg, T. L. (2001). Measuring of the digital economy. *The Netcentric Economy Symposium*. University of Maryland.

Miles, R., & Hamilton, K. (2006). *Learning UML 2.0.* O'Reilly Media, Inc.

Mingers, J., & Walsham, G. (2010). Toward ethical information systems: The contribution of discourse ethics. *MIS Quarterly, 34*(4), 833–854. https://doi.org/10.2307/25750707

Mitchell, R. (2013). *Instant web scraping with Java*. Packt Publishing Ltd.

Mitchell, R. (2015). *Ekstrakcja danych z językiem Python: pozyskiwanie danych z Internetu (Wyd. II)* [Web scraping with Python: collecting data from the modern web (2nd ed.)]. CA: O'Reilly Media.

Mooney, S. J., Westreich, D. J., & El-Sayed, A. M. (2015). Epidemiology in the era of big data. *Epidemiology, 26*(3), 390. https://doi.org/10.1097/EDE.0000000000000274

Mufid, M. R., Basofi, A., Mawaddah, S., Khotimah, K., & Fuad, N. (2020, September). Risk diagnosis and mitigation system of COVID-19 using expert system and web scraping. *2020 International Electronics Symposium (IES)*. https://doi.org/10.1109/ies50839.2020.9231619

Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., & Yu, B. (2019). Interpretable machine learning: Definitions, methods, and applications. *arXiv preprint arXiv: 1901.04592*.

Newell, S., & Marabelli, M. (2015). Strategic opportunities (and challenges) of algorithmic decision-making: A call for action on the long-term societal effects of 'datification'. *The Journal of Strategic Information Systems, 24*(1), 3–14. https://doi.org/10.1016/j.jsis.2015.02.001

Olender-Skorek, M., Czarnecki R., & Bartoszewska B. (2011). Czynniki hamujące rozwój e-usług w Polsce. In H. Babis, R. Czaplewski (Eds.), *Drogi dochodzenia do społeczeństwa informacyjnego: Stan obecny, perspektywy rozwojowe i ograniczenia* (2nd ed., pp. 79–82), Scientific Publishing House of the University of Szczecin.

Olszak, C., & Mach-Król, M. (2018). A conceptual framework for assessing an organization's readiness to adopt big data. *Sustainability, 10*(10), 3734. https://doi.org/10.3390/su10103734

Olszak, C. M., & Zurada, J. (2020). Big data in capturing business value. *Information Systems Management, 37*(3), 240–254. https://doi.org/10.1080/10580530.2020.1696551

Oxford English Dictionary (1989). 2nd ed. Clarendon Press.

Pair, C. (1990). Programming, programming languages and programming methods. In J.-M. Hoc, T. R. G. Green, R. Samuray, & D. Gilmore (Eds.), *Psychology of programming* (pp. 9–19). Academic Press.

Pereira, R. C., & Vanitha, T. (2015). Web scraping of social networks. *International Journal of Innovative Research in Computer and Communication Engineering, 3*(7), 237–240.

Petersen, C., & Plenborg, T. (2006). Voluntary disclosure and information asymmetry in Denmark. *Journal of International Accounting, Auditing and Taxation, 15*(2), 127–149. https://doi.org/10.1016/j.intaccaudtax.2006.08.004

Pinto, J. K., & Prescott, J. E. (1988). Variations in critical success factors over the stages in the project life cycle. *Journal of Management, 14*(1), 5–18. https://doi.org/10.1177/014920638801400102

Poggi, N., Berral, J. L., Moreno, T., Gavalda, R., & Torres, J. (2007, December). Automatic detection and banning of content stealing bots for e-commerce. In *NIPS 2007 workshop on machine learning in adversarial environments for computer security* (Vol. 2).

Rangaswamy & Sampath Kumar, B T & Manjunatha, G. (2017). Internet as a source of information: Usage among the faculty members and students. *Library Waves, 3*(1), 36–42. https://doi.org/10.6084/m9.figshare.11574126

Reddy, N. A., & Divekar, Brig. R. (2014). A study of challenges faced by e-commerce companies in India and methods employed to overcome them. *Procedia Economics and Finance, 11*, 553–560. https://doi.org/10.1016/s2212-5671(14)00220-2

Redziak, Z. (2013). Uncertainty in decision-making. *Zeszyty Naukowe AON, 2*(91), 116–130.

Render, B., Stair, R. M., Hanna, M. E. (2006). *Quantitative analysis for management*. Pearson Prentice Hall.

Rodzeń, D. (2011). Promocja oferty handlowej pośredników w obrocie nieruchomościami z wykorzystaniem portali ogłoszeniowych. *Nierówności Społeczne a Wzrost Gospodarczy, 23*(2011), 209–221.

Santarek, K. (2017). *Rola asymetrii informacji w zarządzaniu*. www.ptzp.org.pl/files/konferencje/kzz/artyk_pdf_2017/T1/t1_259.pdf

Savage, L. J. (1961). The foundations of statistics reconsidered. In J. Neyman (ed.), *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability* (4/1, pp. 575–586). University of California Press.

Schmidt, N-H., Erek, K., Kolbe, M. L., & Zarnekow, R. (2009). Towards a procedural model for sustainable information systems management. *Proceedings of the 42nd Hawaii International Conference on Systems Science* (pp. 1–10).

Schrenk, M. (2007). *Webbots, spiders, and screen scrapers: A guide to developing Internet agents with PHP/CURL*. No Starch Press.

Senge, P. M., Carsted, G., & Porter, P. L. (2001). Innovating our ways to the next industrial decline. *MIT Sloan Management Review, 42*(2), 24–38.

Shannon, C. E. (1948, July, October). A mathematical theory of communication. *The Bell System Technical Journal, 27*, 379–423, 623–656.

Sharma, N. (2004). *The Origin of Data Information Knowledge Wisdom (DIKW) Hierarchy*. Updated: February 4, 2008.

Slamet, C., Andrian, R., Maylawati, D. S., Suhendar, Darmalaksana, W., & Ramdhani, M. A. (2018). Web scraping and naïve Bayes classification for job search engine. *IOP Conference Series: Materials Science and Engineering, 288*, 012038. https://doi.org/10.1088/1757-899x/288/1/012038

Slonneger, K. (1995). *Formal Syntax and semantics of programming language*. Addison-Wesley Publishing Company.

Smithson, M. (1989). *Ignorance and uncertainty: Emerging paradigms*. Springer Verlag.

Snell, K., Care, D. (2013, December). Use of online data in the big data era: Legal issues raised by the use of web crawling and scraping tools for analytics purposes. Bloomberg Law. https://news.bloomberglaw.com/us-law-week/use-of-online-data-in-the-big-data-era-legal-issues-raised-by-the-use-of-web-crawling-and-scraping-tools-for-analytics-purposes

Snell, J., Menaldo, N. (2016). Web scraping in an era of big data 2.0. *Electronic Commerce & Law Report*, 21 ECLR 920, 6/8/16. Bloomberg BNA. ISSN 1098-5190. https://www.perkinscoie.com/images/content/1/5/v2/156775/Snell-web-scraping-BNAI.pdf

Someh, I., Davern, M., Breidbach, C. F., & Shanks, G. (2019). Ethical issues in big data analytics: A stakeholder perspective. *Communications of the Association for Information Systems, 44*, 718–747. https://doi.org/10.17705/1cais.04434

Stal, J., & Paliwoda-Pękosz, G. (2017, October 23-24). Towards integration of mobile technology and knowledge management in organizations: A preliminary study. In J. Kowal et al. (Eds.), *Innovations for human development in transition economies. Proceedings of the International Conference on ICT Management for Global Competitiveness and Economic Growth in Emerging Economies* (pp. 204–214). Available at SSRN: https://ssrn.com/abstract=3110231

Stigler, G. J. (1961). The economics of information. *Journal of Political Economy, 69*, 213–285.

Stiglitz, J. E. (2002). Information and the Change in the Paradigm in Economics. *American Economic Review*, *92*(3), 460–501. https://doi.org/10.1257/00028280260136363

Stiving, M. (2017). B2B pricing systems: Proving ROI. In A. Hinterhuber, & S. M. Liouzu (Eds.), *Innovation in pricing* (pp. 137–144). Routledge.

Szpringer, W. (2005). Prowadzenie działalności gospodarczej w Internecie: Od e-commerce do e-businessu. Difin.

Tapscott, D. (1995). The digital economy: Promise and peril in the age of networked intelligence. McGraw-Hill.

Thomas, D. M., & Mathur, S. (2019, June). Data analysis by web scraping using Python. *2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA). https://doi.org/10.1109/iceca.2019.8822022

Thomsen, J. G., Ernst, E., Brabrand, C., & Schwartzbach, M. (2012, July). WebSelF: A web scraping framework. In *International Conference on Web Engineering* (pp. 347–361). Springer.

Tillström, J. (2012). Gamification in automotive marketing: A conceptual framework for implementation [Bachelor's thesis]. Helsinki Metropolia University of Applied Sciences Bachelor of Business Administration.

Ulieru, M., Verdon, J. (2009). Organizational transformation in the digital economy. *Proceedings of the 7th IEEE International Conference on Industrial Informatics*, Cardiff, UK, IEEE, 17–24.

Vakkari, P., Völske, M., Potthast, M., Hagen, M., & Stein, B. (2019). Modeling the usefulness of search results as measured by information use. *Information Processing & Management, 56*(3), 879–894. https://doi.org/10.1016/j.ipm.2019.02.001

van den Berg, J. (2007). Conceptualising and analysing internet threats using a 4-Dimensional Hypercube. *Proceedings of the International Conference on Information Society (I-Society 2007)*.

van Rijnsoever, F., Farla, F., & Dijst, M. R. (2009). Consumer car preferences and information search channels. *Transportation Research, D14*, 334–342.

vanden Broucke, S., & Baesens, B. (2018). Introduction. In *Practical web scraping for data science* (pp. 3–23). Apress. https://doi.org/10.1007/978-1-4842-3582-9_1

Vargiu, E., & Urru, M. (2013). Exploiting web scraping in a collaborative filtering-based approach to web advertising. *Artificial Intelligence Research, 2*(1). https://doi.org/10.5430/air.v2n1p44

Watson, H. J. (2014). Tutorial: Big Data Analytics: Concepts, Technologies, and Applications. *Communications of the Association for Information Systems*, 34(1), 1247–1268.

Weick, K. E. (1979). *The social psychology of organizing*. Addison Wesley.

Weick, K. E. (1995). *Sensemaking in organizations*. Sage.

Wieder, B., & Ossimitz, M.-L. (2015). The impact of business intelligence on the quality of decision making – A mediation model. *Procedia Computer Science, 64*, 1163–1171. https://doi.org/10.1016/j.procs.2015.08.599

Wigan, M. R., & Clarke, R. (June, 2013). Big data's big unintended consequences. *Computer, 46*(6), IEEE, 46–53. https://doi.org/10.1109/MC.2013.195

Wirth, N. (1976). *Algorithms + data structures = programs*. Prentice Hall.

Xu, W., Liu, L., & Shang, W. (2017). Leveraging cross-media analytics to detect events and mine opinions for emergency management. *Online Information Review, 41*(4), 487–506. https://doi.org/10.1108/oir-08-2015-0286

Yang, W., & Meyer, K.E. (2015). Competitive dynamics in an emerging economy: Competitive pressures, resources, and the speed of action. *Journal of Business Research, 68*(6), 1176–1185.

Yi, L., Liu, B., & Li, X. (2003). Eliminating noisy information in Web pages for data mining. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining – KDD '03. the ninth ACM SIGKDD international conference*. https://doi.org/10.1145/956750.956785

Yi, L., & Liu, B. (August, 2003). Web page cleaning for web mining through feature weighting. *IJCAI*, 43–48.

Zaied, A. N. H. (2012). Barriers to E-commerce adoption in Egyptian SMEs. *International Journal of Information Engineering and Electronic Business*, *4*(3), 9–18. https://doi.org/10.5815/ijieeb.2012.03.02

Zamora, A. (2019). Making room for big data: Web scraping and an affirmative right to access publicly available information online. *The Journal of Business Entrepreneurship & the Law, 12*(1), 202–227.

Zeleny, M. (1987). Management support systems: Towards integrated knowledge management. *Human Systems Management, 7*(1987)1, 59–70.

Zhai, Y., & Liu, B. (2006). Structured data extraction from the web based on partial tree alignment. *IEEE Transactions on Knowledge and Data Engineering, 18*(12), 1614–1628. https://doi.org/10.1109/tkde.2006.197

Zhao, B. (2017). Web scraping. In *Encyclopedia of big data* (pp. 1–3). Springer. https://doi.org/10.1007/978-3-319-32001-4_483-1

Zhou, Z., & Mashuq, M. (2014). *Web content extraction through machine learning*. Stanford University, 1–5. http://cs229.stanford.edu/proj2013/ZhouMashuq-WebContentExtractionThroughMachineLearning.pdf

Internet sources

Cambridge International Examinations (2015). *Cambridge International AS & A Level (Information Technology, 9626): Topic 1.1 Data, information and knowledge*. https://www.cambridgeinternational.org/Images/285017-data-information-and-knowledge.pdf

Centrum Pomocy OTOMOTO (2021). *Regulamin dla klientów biznesowych*. https://pomoc.otomoto.pl/hc/pl/articles/360003999633

CloudFlare (n.d.). *What is DDoS attack?*. https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/

Crummy.com (n.d.). *Beautiful Soup Documentation*. https://www.crummy.com/software/BeautifulSoup/bs4/doc/

European Union (n.d.). *Data protection under GDPR*. https://europa.eu/youreurope/business/dealing-with-customers/data-protection/data-protection-gdpr/index_en.htm

Eurostat (September, 2020). *Digital economy and society statistics – households and individuals*. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Digital_economy_and_society_statistics_-_households_and_individuals

FindLaw (December, 2018). *Trespass to Chattels*. https://www.findlaw.com/injury/torts-and-personal-injuries/trespass-to-chattels.html

Google Developers (November, 2020). Googlebot. In *Google Search Central*. https://developers.google.com/search/docs/advanced/crawling/googlebot

IAB Polska (2013). *Bariery i potencjał rozwojowy w handlu elektronicznym dobrami fizycznymi i treściami cyfrowymi*. http://iab.org.pl/legislacja/stanowiska/ekspertyza-zwiazku-pracodawcow-branzy-internetowej-iab-polska-bariery-i-potencjal-rozwojowy-w-handlu-elektronicznym-dobrami-fizycznymi-i-tresciami-cyfrowymi

IBM (2018). *The Four V's of Big Data*. http://www.ibmbigdatahub.com/infographic/four-vs-big-data

Legal Information Institute. (n.d.). Cease and Desist Letter. In *Wex*. Cornell Law School. https://www.law.cornell.edu/wex/cease_and_desist_letter

Merriam-Webster. (n.d.). Kitchen-sink. In *Merriam-Webster.com dictionary*. https://www.merriam-webster.com/dictionary/kitchen-sink

Miniwatts Marketing Group (2020). Internet World Stats: Usage and Population Statistics. http://www.internetworldstats.com/

Polish General Inspectorate of Personal Data (currently: Polish Office for Personal Data Protection) (n.d.). *Co to jest zbiór danych osobowych*, (What is the personal data set). https://www.giodo.gov.pl/530/id_art/2657

Pydata.org. (n.d.). *About pandas*. https://pandas.pydata.org/about/

Python Software Foundation. (n.d.). *Python Software Foundation*. https://www.python.org/psf/

Selenium.dev. (n.d.). *About Selenium*. https://www.selenium.dev/about/

Stanford University Libraries. (n.d.). What is Fair Use?. In *Copyright & Fair Use*. https://fairuse.stanford.edu/overview/fair-use/what-is-fair-use/

Twitter (May, 2012). *Terms of Service* (Version 6). https://twitter.com/it/tos/previous/version_6

W3C (2019a). *About W3C*. https://www.w3.org/Consortium/

W3C (2019b). *W3C Mission.* https://www.w3.org/Consortium/mission

W3Schools (2020). *JavaScript HTML DOM*. In *W3Schools.com*. https://www.w3schools.com/js/js_htmldom.asp

WEF (2015). *Expanding Participation and Boosting Growth: The Infrastructure Needs of the Digital Economy*. World Economic Forum, Geneva. http://www3.weforum.org/docs/WEFUSA_DigitalInfrastructure_Report2015.pdf

WirtualneMedia.pl (August, 2020). *Ranking serwisów o motoryzacji: otoMoto liderem (TOP10)*. https://www.wirtualnemedia.pl/artykul/ranking-serwisow-o-motoryzacji-oto-moto

Legal acts

Ustawa z dnia 27 lipca 2001 r. o ochronie baz danych (Dz. U. 2001 nr 128 poz. 1402). [Polish Act of July 27, 2001 on the protection of databases].

Ustawa z dnia 5 lutego 1995 r. o prawie autorskim i prawach pokrewnych (Dz. U. 1994 nr 24 poz. 83). [Polish Act of February 5, 1995 on Copyright and Related Rights].

Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation; GDPR).

List of Figures

List of Tables

Explore the transformative role of web scraping in mitigating information asymmetry within e-commerce through this scholarly volume. With a focus on the automotive sector, particularly the prominent OTOMOTO portal in Poland, this book provides a nuanced examination of automated tools designed to empower market participants. Tailored for researchers, academics, and data scientists, as well as legal professionals and policymakers who may find valuable perspectives on the regulatory landscape surrounding web scraping and its implications for information transparency. Emphasizing the imperative of informed decision-making, this work serves as a comprehensive resource for understanding and harnessing the potential of web scraping in the digital marketplace.

Bartłomiej Hadasik (born in 1996) – Research and Teaching Assistant in the Department of Business Informatics at the University of Economics in Katowice. Currently pursuing a PhD on trust-oriented research in consumer-organizational relationships, with research interests extending to managerial aspects of sustainable development and IT-driven issues in business. Recognized for outstanding academic achievements, including distinctions in the Scientific Society of Economic Informatics' Diploma Thesis Competition, as well as Rector's awards for young researchers. Author of over 20 scientific publications, indexed in international databases, such as Scopus or Web of Science.

e-mail: bartlomiej.hadasik@ue.katowice.pl

ORCID: https://orcid.org/0000-0001-6604-1970

ISBN 978-83-7875-889-1
e-ISBN 978-83-7875-890-7

University of Economics in Katowice