Used Car Characteristics

An Explanatory Data Analysis

Dávid Fodor
Faisal Zayeem Ahmed
Márió Palágyi

Submission date: TBD

Datascience Capstone Project - IMC Krems

Abstract

A brief overview of the project, including problem statement, methodology, key findings, and conclusions.

1 Introduction

Problem Statement: Describe the problem.

Objectives: List the objectives and tasks required to be completed.

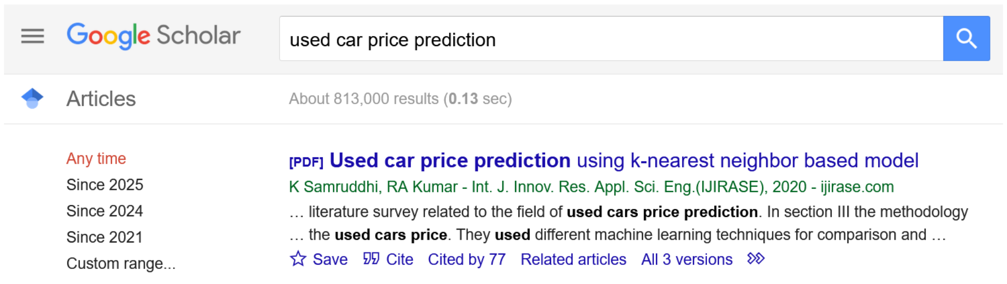
Data: Describe your dataset.

Background Information: Provide context and challenges based on your understanding of the problem.

2 Literature Review

Summary of relevant research and related works in the field. This section does not need to be exhaustive; include background knowledge on how the problem has been addressed before and how you plan to solve it.

Luckily, there is plenty of literature on the topic of used car prices, particularly the prediction car prices seems to be a popular topic. Searching on Google Scholar for "used car price prediction" yields almost a million results.



About 813,000 results (**0.13** sec)

[PDF] **Used car price prediction** using k-nearest neighbor based model
K Samruddhi, RA Kumar - Int. J. Innov. Res. Appl. Sci. Eng.(IJIRASE), 2020 - ijirase.com
… literature survey related to the field of **used cars price prediction**. In section III the methodology … the **used cars price**. They **used** different machine learning techniques for comparison and …

Figure 1: Screenshot of Google Scholar Search

3 Methodology

- **Data Collection, Description and Management:** Describe the steps involved in data collection or preprocessing, including EDA.

- **Analysis Techniques:** Describe your approach to solving the problem. You can include a model diagram to illustrate your method.
- **Tools and Software Used:** List the tools, frameworks, and software used for the project.

3.1 Data Collection

In our case the data collection is a rather interesting part of the project, which took quite some time. Unlike other groups, we had to find our own primary datasource, seeking a used car marketplace which we could scrape. After several trials of scraping different websites, we finally ended up with one that was quite lenient with their robots.txt file, and we could scrape it without any issues. The website we used was `https://www.autoscout24.com/`, which is one of the largest used car marketplaces in Europe boasting more than 2 million listings.

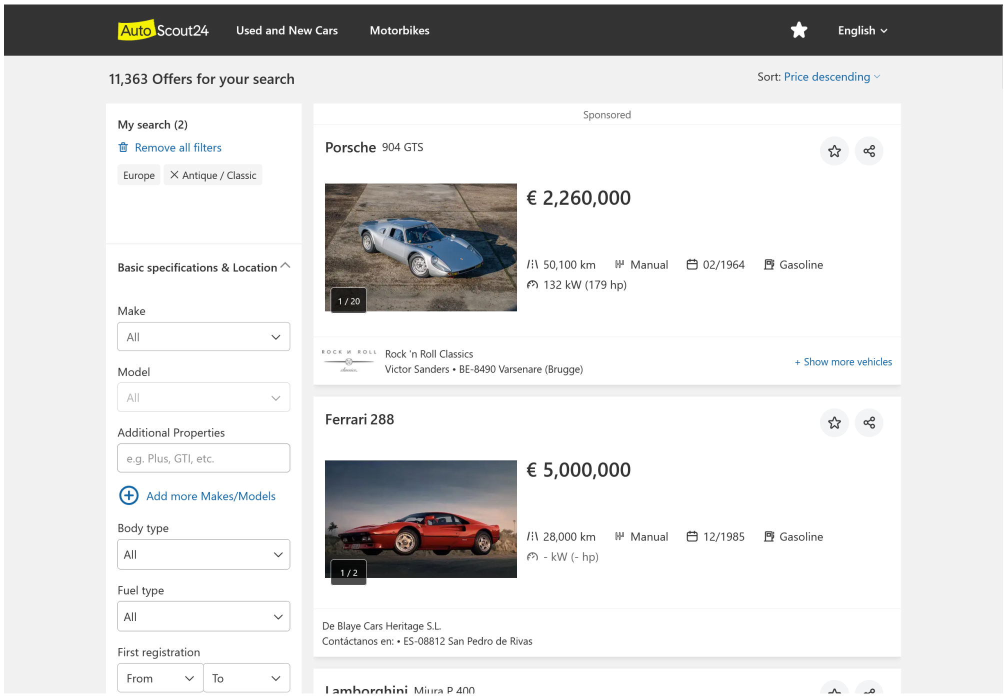
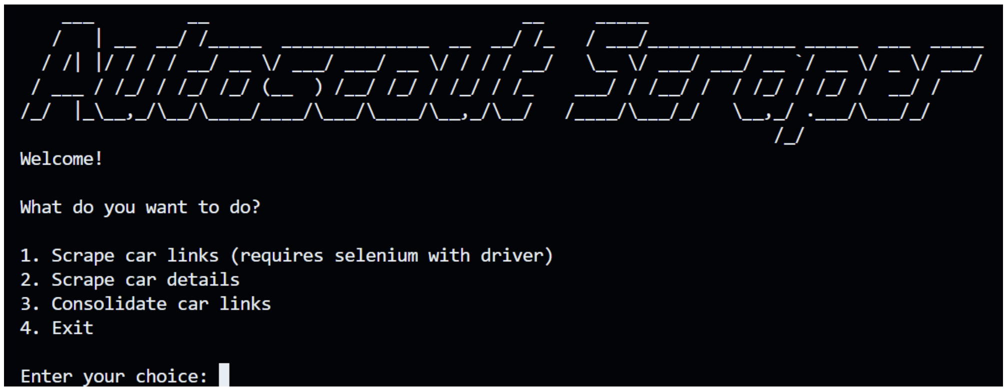


Figure 2: Screenshot of Autoscout24 Homepage

The scraping took place from to , running practically 24/7 on a laptop even though parallelization was used. In the end we have collected **505804** rows of data, which is a lot more than we could have hoped for in, and wrote it to a **parquet** file. The scraping consisted of two main parts, which we will discuss in the following subsections along with the bigger challanges we have faced.



```
    ___         __                              __     _____
   /   | __  __/ /_____  ______________  __  __/ /_   / ___/______________ _____  ___  _____
  / /| |/ / / / __/ __ \/ ___/ ___/ __ \/ / / / __/   \__ \/ ___/ ___/ __ `/ __ \/ _ \/ ___/
 / ___ / /_/ / /_/ /_/ (__  ) /__/ /_/ / /_/ / /_    ___/ / /__/ /  / /_/ / /_/ /  __/ /
/_/  |_\__,_/\__/\____/____/\___/\____/\__,_/\__/   /____/\___/_/   \__,_/ .___/\___/_/
                                                                        /_/
Welcome!

What do you want to do?

1. Scrape car links (requires selenium with driver)
2. Scrape car details
3. Consolidate car links
4. Exit

Enter your choice:
```

Figure 3: Screenshot of Autoscout Scraper

3.1.1 Scraping Car Listing

Due to the nature of the website, we needed user interaction for scraping the car listing, so we used **Selenium** to scrape the URLs pointing to specific car listings. This was achieved through a headless Chromium browser. We faced two hurdles in this change which we had to overcome with simulated interactions.

- **Pagination:** The website uses a classic pagination system on page, meaning we needed to be able to navigate through each page.
- **Max 20 pages per search:** The website limits the number of returned pages to 20, so we had to find a way to methodically filter search parameters to find as many sets of 20 pages as possible.

In theory almost every common feature of cars can be filtered on the page, so with enough patience one may find all 2 million listings. However, we did not have so much time, nor so many cores on our hands, so we tried finding a middle ground, which was the following:

1. Loop through each car make available
2. For each car make, loop through a range of dynamically changing power ranges in kw based on their presumed likelihood e.g. 0-15, 15-20, 20-25, 25-26, 26, 27...
3. For each power range of the car make, loop through the pages available of the pagination
4. On each pagination page, collect the URLs of the car listings
5. Store the URLs in a JSON file, keyed by the make

This constituted most of our time scraping, as the user interaction needed to be waited upon and there were simply a lot of possible permutations of make, power, pagination possibilities to go through. We of course parallelized and worked on multiple cores, also introducing shortcuts, e.g. if we found a make with less than 500 lisitng we did not need to loop through power ranges at all.

3.1.2 Scraping Car Details

Once we have collected all the car URLs, the scraping was quite straightforward. We have tested the scraping on individual listings first, trying to find out what categories of data we can scrape and how we may achieve that. At this stage, we were lucky, as no user interaction was needed at all, not even for **declining cookies**, as the underlying HTML structure already contained the data we needed. This allowed for us to scrape the data in a much more efficient way, as we could simply use the **requests** package and **BeautifulSoup** to parse the HTML and extract the data we needed. This had much lower overhead and allowed us to scrape more listings simultaneously. First fetching into a list in memory, then converting to a df, to then export into a **parquet** file.

Interesting to note that between the two scraping phases, many links seem to have moved, perhaps due to being sold, or delisted so this case also needed to be addressed.

3.2 Preprocessing

Preoprocessing was...

4 Results

Describe experiments and the evaluation protocol. Include tables, graphs, and charts as needed to present your findings and results.

5 Discussion

Interpret and analyze the results, and discuss possible future work and improvements.

6 Conclusion

Provide a summary of the project, key findings, and recommendations. (Approximately 200–350 words.)

7 Bibliography

Include relevant references such as websites, blogs, articles, research papers, etc.

8 Self-Reflection

Add a paragraph or half-page note reflecting on the project.