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.

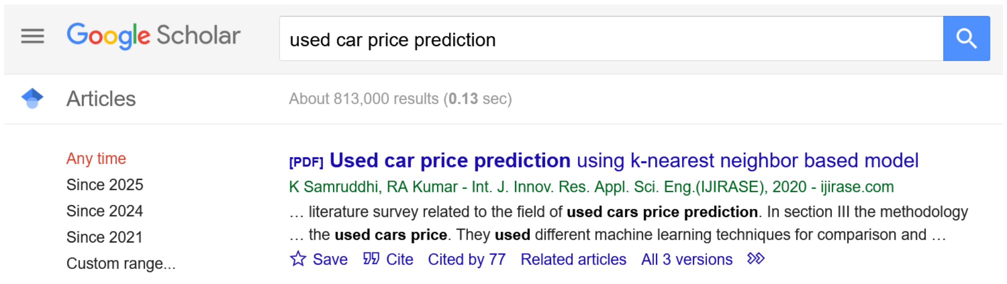
Data: Describe your dataset.

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 seems to be a popular topic. Searching on Google Scholar for "used car price prediction" yields almost a million results.



used car price prediction

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 …
Cited by 77

Figure 1: Screenshot of Google Scholar Search. Source: Google Scholar

However, looking for papers that solely try and explore the data and find interesting patterns is lot more difficult. Mostly, the exploration of all the data is done in the context of a prediction model, which is why we also aim to create a usable prediction model as part of our assignment.

Here, as the literature review is a small portion of the report, I will just mention some of the interesting conclusions we have found from others. Firstly, there is seemingly no real consensus on what the best model is for predicting car prices. The the data is usually scraped by the authors [1][2][3], or taken from a source like Kaggle [4][5].

Most of the papers that compared multiple models, found that more complex models, such as **XGBoost** [5], **Random Forest** [6][3] or **Neural Netwroks**[7] worked the best, but the differences were often not so significant, the best models achieved were usually between *85-95%* accuracy. There were some surprises too, some have found that rather interestingly other methods worked better for them, such as **SVM** [8], **LightGBM** [9] or **Linear Regression** [2] in case of a smaller dataset.

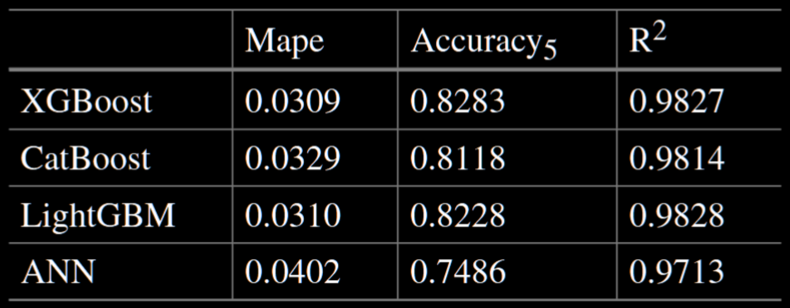
Some other interesting findings include for example Zheng Y. [4] for whom the most challenging part of price prediction in his case of XGBoost, Random Forest and Linear Regression was of predicting the price of premium cars, the models constantly underestimated them, leading to worse accuracy. The author also suggests taking extra special care when cleaning outliers in price. Chen et al. [10] split their dataset into three distinct groups, once using just cars of the same make and build year, once just a car series and once all data together. They found that random forest worked all round better than linear regression, and that the more records and more features they included, the greater this distance became. This and other papers also suggest that we should not bother with linear regression in cases of larger, more complex datasets. Nandan et al. [5] had a similar experience to us with their large amount of missing values and they also had features reminescent of ours in their Kaggle dataset, they solved the missing values issue with the following method, which they found most optimal and we shall also try to implement in our case:

> "To replenish the missing values in the data, the IterativeImputer technique is employed, with a variety of estimators being developed and their respective MSEs being generated ... the ExtraTreesRegressor estimator is preferable for the imputation strategy in the case of missing value"

Huang et al. [11] had similarly many features to us, leading them to use a forward feature selection which we also plan on implementing due to the large number of features we have. One more interesting thing to note is how they came up with their final prediction, which is a weighted average of their different already well functioning models.

> "...weighted average fusion of XGBoost, CatBoost, LightGBM, and ANN. Its R2 can reach 0.9845, and the prediction effect is the best."

As a conclusion of the literature review, we can safely say that there is no real consensus on what the best model may be for predicting car prices, which may be due to many factors, like the different datasets, different preoprocessing steps and different features. However, we can also see that the more complex models perform better in exploring more



	Mape	$Accuracy_5$	R^2
XGBoost	0.0309	0.8283	0.9827
CatBoost	0.0329	0.8118	0.9814
LightGBM	0.0310	0.8228	0.9828
ANN	0.0402	0.7486	0.9713

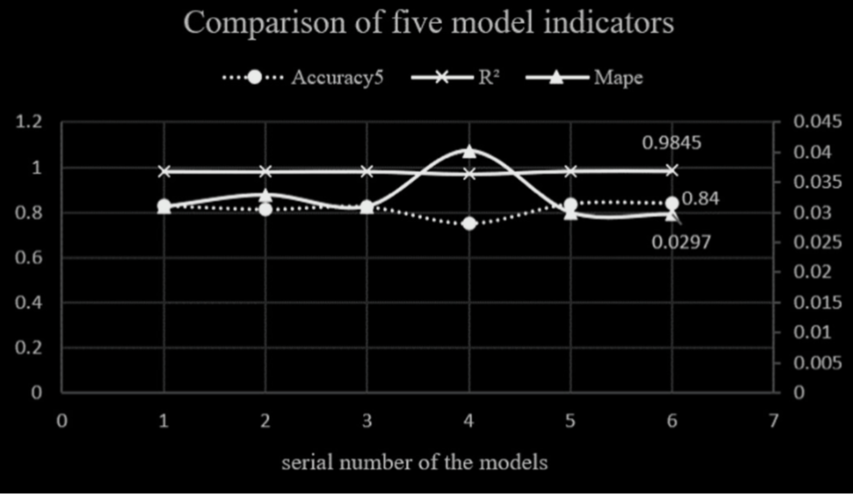


Figure 2: Huang et al. comparative performance of models. Source: [11]

nuanced relationships, leading to better predictions as corroborated by Voß S. [6]. Models such as XGBoost and Random Forest seem to be some of the most popular ones, and they are also the ones that we will be expecting to work the best for our case.

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.

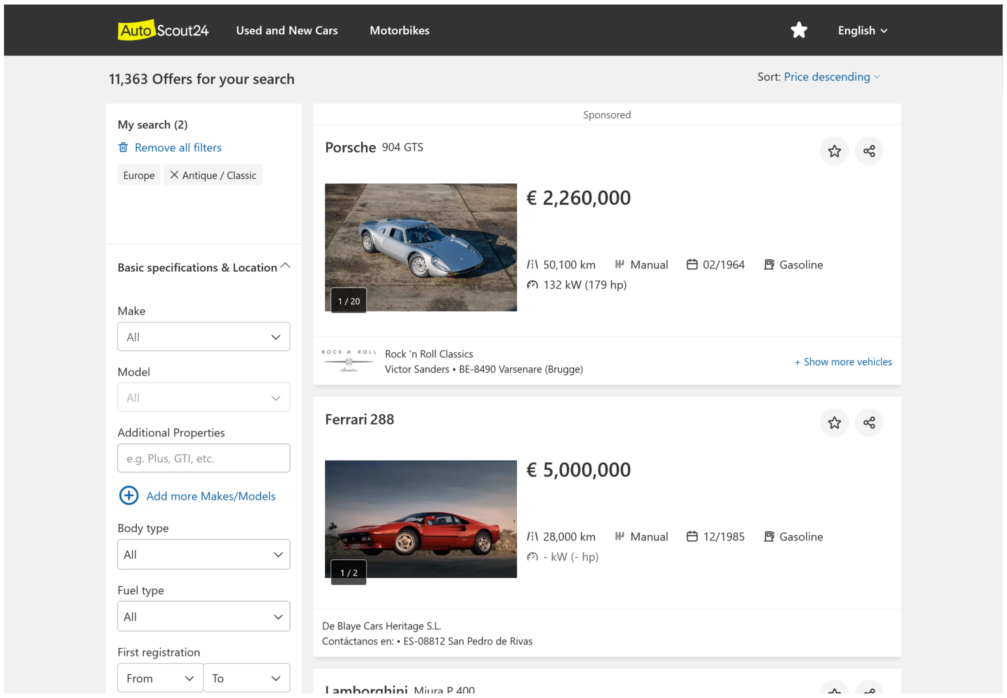
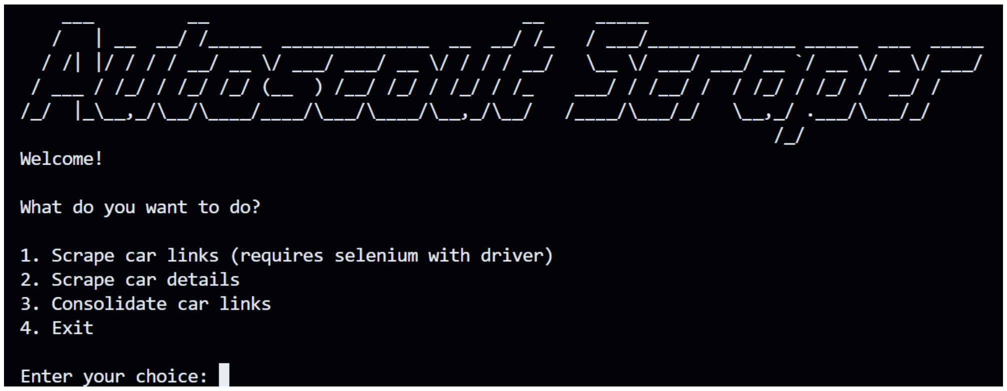


Figure 3: Screenshot of Autoscout24 Homepage. source: Autoscout24

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 challenges we have faced.

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.



```
    ___         __                              __     _____
   /   | __  __/ /_____  ______________  __  __/ /_   / ___/______________ _____  ___  _____
  / /| |/ / / / __/ __ \/ ___/ ___/ __ \/ / / / __/   \__ \/ ___/ ___/ __ `/ __ \/ _ \/ ___/
 / ___ / /_/ / /_/ /_/ (__  ) /__/ /_/ / /_/ / /_    ___/ / /__/ /  / /_/ / /_/ /  __/ /
/_/  |_\__,_/\__/\____/____/\___/\____/\__,_/\__/   /____/\___/_/   \__,_/ .___/\___/_/
                                                                        /_/
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 4: Screenshot of Autoscout Scraper. Source: Autoscout24

- **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

We worked separately in numeral Jupyter notebooks, there have been separate notebooks created per each member exploring different aspects of the data. We first started by cleaning the data at the start of each of our own separate notebooks, but later on we decided to do a comprehensive cleaning of the dataset in a separate joint notebook, which can serve as the basis for all of our future work.

The logic can be found in the appropriately named **data_cleaning.ipynb** notebook and it contains not only the cleaning logic, but our reasoning behind each of the cleaning steps. We will now summarize the the steps taken in a nutshell.

First off, we got rid of useless features, this included completely empty, almost empty columns and and simply features that we would not need for any of our analysis, like **manufacturer_colour** for example.

Then, having seen that there are not too many missing values for our main features, **price**, **mileage** and **power**, we decided to drop the rows with missing values, removing *9879* rows, from our 504k large dataset. This seemed like a reasonable choice.

We also had to transform some columns to make them usable for our purposes:

- **Location:** was transformed from the format of `[city, country code]` to just the country code, as we were not interested in the fine grained location of the car.
- **Fuel consumption:** was similarly transformed from `[float] l/100 km (comb.)` string format to just the float value.
- **First registration:** was transformed to **age_months** by subtracting the first registration date from the current date, as we were only interested in the age of the car, not when it was registered.

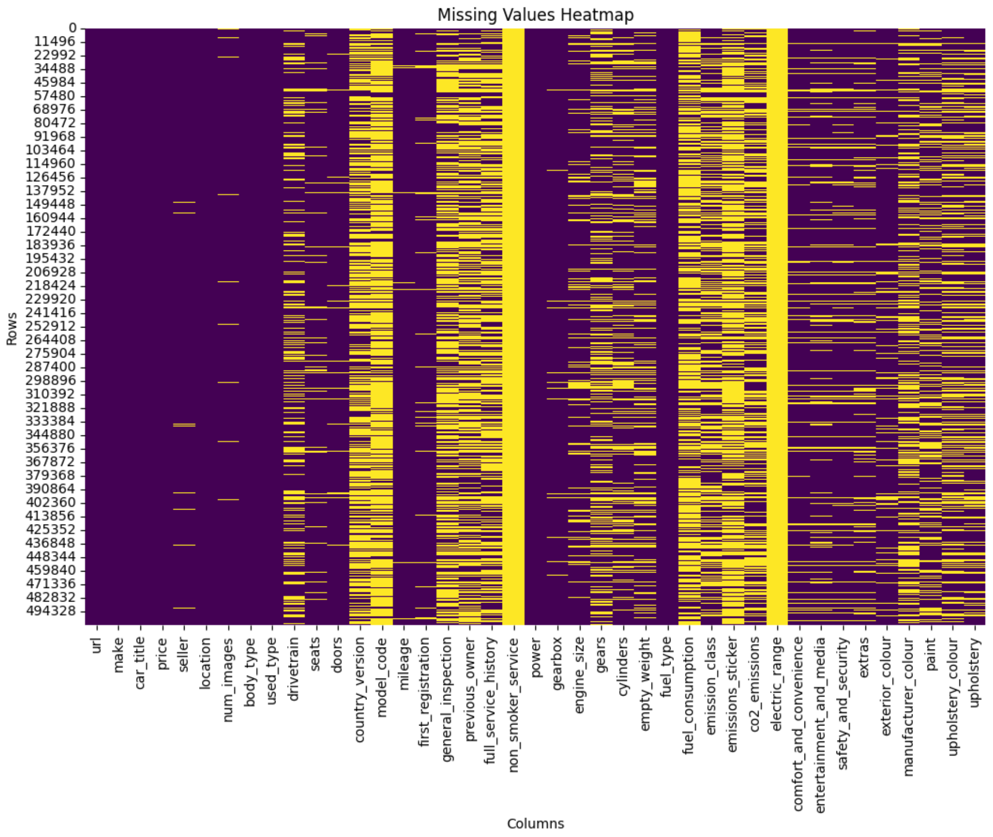


Figure 5: Missing Values Heatmap. Source: Own work

We found a few imputable / inferable values in our dataset as well:

- **Age:** had seeminlgy *21454* missing values, wehich we could all infer from the column **use_type**, whenever this feature was **New** or **Demonstration**, we filled the missing values with *0*, as these cars were not used at all. This left us with no missing values at all in this column.
- **Fuel Type:** given that the fuel type was electric, we could safely infer that the **co2_emission** was **0** and that the **gearbox** was **automatic**.

In the end came the most interesting part, the outlier cleaning. We have inspected 8 of our important numerical features and found that there were some serious outliers in the dataset.

We have used many different strategies to clean the outliers, each based on what felt right for the feature, based on our domain knowledge. Sometimes, we used hard numerical caps, other times used IQR or even 0.05 quantile based capping. As we had so many records to work with, we did not believe winsorization was necessary, so we simply dropped the outliers. One of the more interesting features to clean was the **price**

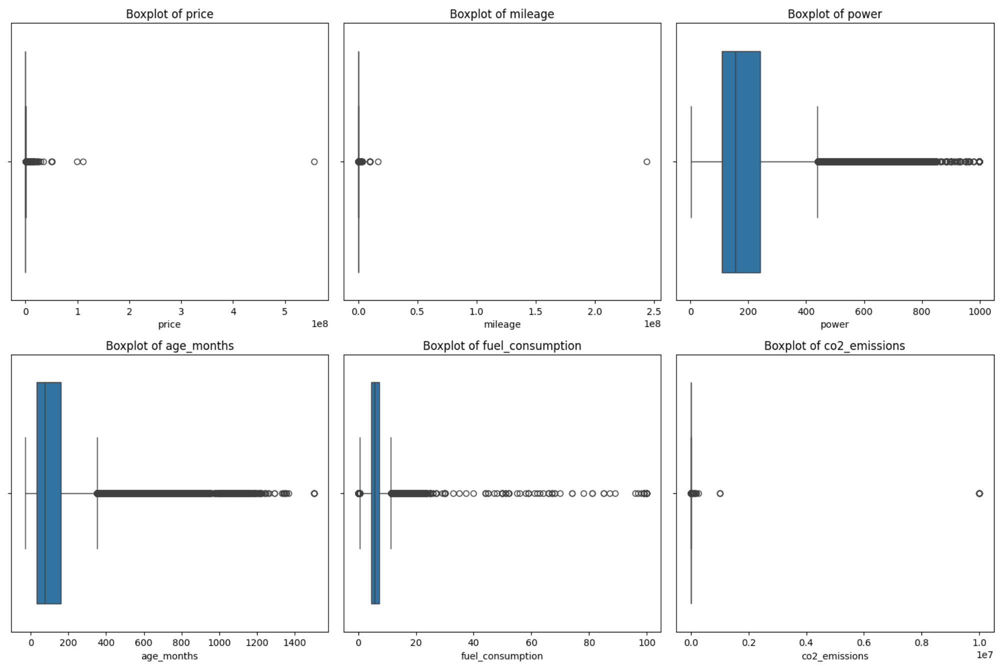


Figure 6: Boxplots of Numerical Features Before Cleaning. Source: Own work

as, this had many outliers on both ends and was arguably our most important feature of all. Here we applied a two step process:

1. We cleaned the outlers with a lenient 2 * IQR based capping, per car make, as this allowed us to keep values that would have been outliers in the general dataset, but were not outliers at all, e.g. Lamborghini.
2. Then we manually inspected both ends of the distribution and set a hard cap of *500* EUR on the lower end and *1000000* EUR on the upper end, cleaning out the most extreme outliers. (This inspection led us to notice interesting things, e.g. towable trailers was listed as a car make, which we removed.)

After removing the outliers for other features, with similar care, but less manual inspection, we were left with a much cleaner dataset, as visible from the following boxplots.

The final dataset was then saved to an alternative parquet file, which we will be using for our further analysis. The cleaning here was on purpose not too harsh, such that each subsequent analysis can decide to further clean the data if needed, but we believe we have already found a good balance between keeping the data and cleaning it. In the end this cleaning process removed *36430* rows from our dataset, which is just *7.2%* of the original dataset. This seems reasonable, and we have addressed the the possible downsides of our process as losing all the data for some ultra premium car brands, like Koenigsegg, Paganni and losing some data in case a regular car brand happens to have premium models as well, e.g. Nissan GTR.

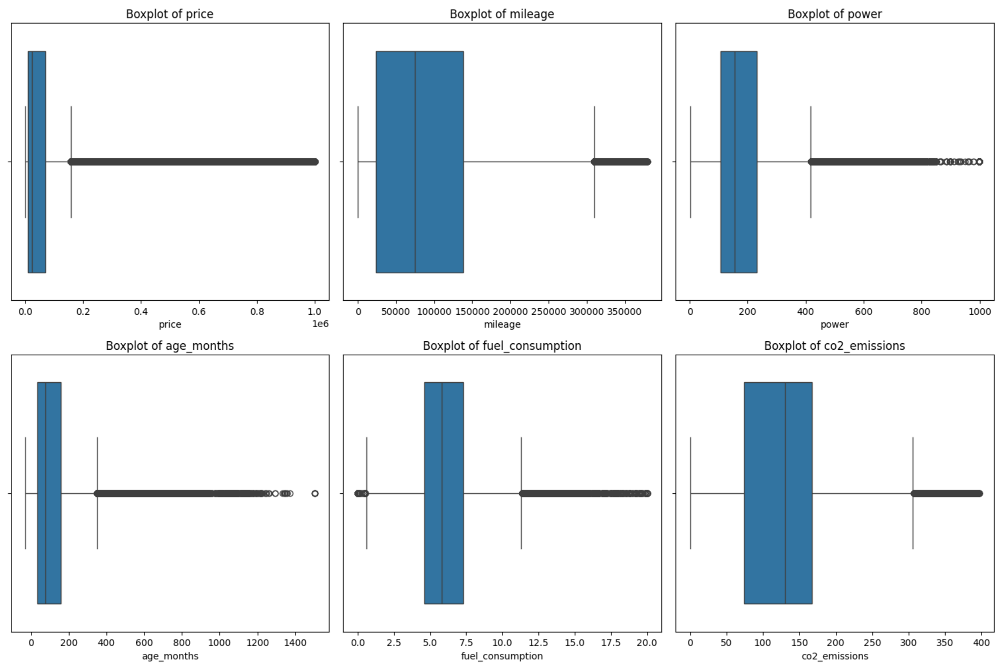


Figure 7: Boxplots of Numerical Features After Cleaning. Source: Own work

3.3 Exploratory Data Analysis

TODO: Here we will present some interesting findings from our exploratory data analysis.

3.4 Tools and Software Used

We used a variety of tools and software for this project, including:

- **Python** primary programming language, used for data scraping, cleaning, and analysis.
- **Selenium** used for web scraping, when user interaction was necessary.
- **BeautifulSoup** for parsing HTML, gathering data from requests.
- **Pandas** for data manipulation and analysis via DataFrames.
- **Matplotlib** and **Seaborn** for data visualization.
- **Jupyter Notebook** for interactive development and documentation.
- **Scikit-learn** for machine learning and model evaluation.
- **Pytorch** for deep learning models, if applicable.

- **LaTeX** for typesetting and writing the report.
- **Zotero** for managing references and citations.
- **Git** for version control and collaboration between team members.

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

[1] E. Gegic and B. Isakovic, “Car Price Prediction using Machine Learning Techniques,” *TEM Journal*, vol. 8, no. 1, pp. 113–118, 2019, ISSN: 2217-8309, 2217-8333. [Online]. Available: `https://www.ceeol.com/search/article-detail?id=746689` (visited on 05/20/2025).

[2] L. Bukvić, J. Pašagić Škrinjar, T. Fratrović, and B. Abramović, “Price Prediction and Classification of Used-Vehicles Using Supervised Machine Learning,” *Sustainability*, vol. 14, no. 24, p. 17 034, 24 Jan. 2022, ISSN: 2071-1050. DOI: `10.3390/su142417034`. [Online]. Available: `https://www.mdpi.com/2071-1050/14/24/17034` (visited on 05/20/2025).

[3] C. Jin, “Price Prediction of Used Cars Using Machine Learning,” in *2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT)*, Nov. 2021, pp. 223–230. DOI: `10.1109/ICESIT53460.2021.9696839`. [Online]. Available: `https://ieeexplore.ieee.org/abstract/document/9696839` (visited on 05/20/2025).

[4] Y. Zheng, "Machine Learning Optimization and Challenges in Used Car Price Prediction," *ITM Web of Conferences*, vol. 70, p. 04 032, 2025, ISSN: 2271-2097. DOI: 10.1051/itmconf/20257004032. [Online]. Available: https://www.itm-conferences.org/articles/itmconf/abs/2025/01/itmconf_dai2024_04032/itmconf_dai2024_04032.html (visited on 05/20/2025).

[5] M. Nandan and D. Ghosh, "Pre-owned car price prediction by employing machine learning techniques," *Journal of Decision Analytics and Intelligent Computing*, vol. 3, no. 1, pp. 167–184, 1 Oct. 8, 2023, ISSN: 2787-2572. DOI: 10.31181/jdaic10008102023n. [Online]. Available: https://www.jdaic-journal.org/index.php/about/article/view/31 (visited on 05/21/2025).

[6] S. Voß and S. Lessmann, "Resale Price Prediction in the Used Car Market,"

[7] E. Liu, J. Li, A. Zheng, H. Liu, and T. Jiang, "Research on the Prediction Model of the Used Car Price in View of the PSO-GRA-BP Neural Network," *Sustainability*, vol. 14, no. 15, p. 8993, 15 Jan. 2022, ISSN: 2071-1050. DOI: 10.3390/su14158993. [Online]. Available: https://www.mdpi.com/2071-1050/14/15/8993 (visited on 05/20/2025).

[8] S. Shaprapawad, P. Borugadda, and N. Koshika, "Car Price Prediction:An Application of Machine Learning," in *2023 International Conference on Inventive Computation Technologies (ICICT)*, Apr. 2023, pp. 242–248. DOI: 10.1109/ICICT57646.2023.10134142. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10134142 (visited on 05/23/2025).

[9] Y. Li, Y. Li, and Y. Liu, "Research on used car price prediction based on random forest and LightGBM," in *2022 IEEE 2nd International Conference on Data Science and Computer Application (ICDSCA)*, Oct. 2022, pp. 539–543. DOI: 10.1109/ICDSCA56264.2022.9988116. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9988116 (visited on 05/20/2025).

[10] C. Chen, L. Hao, and C. Xu, "Comparative analysis of used car price evaluation models," *AIP Conference Proceedings*, vol. 1839, no. 1, p. 020 165, May 8, 2017, ISSN: 0094-243X. DOI: 10.1063/1.4982530. [Online]. Available: https://doi.org/10.1063/1.4982530 (visited on 05/20/2025).

[11] J. Huang, Z. Yu, Z. Ning, and D. Hu, "Used Car Price Prediction Analysis Based on Machine Learning," presented at the 2022 International Conference on Artificial Intelligence, Internet and Digital Economy (ICAID 2022), Atlantis Press, Dec. 2, 2022, pp. 356–364, ISBN: 978-94-6463-010-7. DOI: 10.2991/978-94-6463-010-7_37. [Online]. Available: https://www.atlantis-press.com/proceedings/icaid-22/125977128 (visited on 05/20/2025).

8 Self-Reflection

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