AutoCommodity Data Collection & Analysis

Analyzing the Relationship Between Used Car & Commodity Prices

Group 206 — Dongyuan Gao, Ramiro Diez-Liebana, Cyriel Van Helleputte

November 2025

Contents

Group 206

Dongyuan Gao · Ramiro Diez-Liebana · Cyriel Van Helleputte

---

1 Project Overview

1.1 The Storyline (Potential Business Problem)

The Swiss used car market is highly competitive. Our **fictional client** AutoHelvetia AG, a leading national **used car dealer**, faces the challenge of optimizing their pricing & purchasing strategy. In recent years, commodity prices are volatile and affects pricing of cars. So AutoHelvetia AG delegated the task to us: to understand the relationship between used car prices and commodity prices.

1.2 Our Solution

This project delivers an **advanced data collection** and **anylsis** framework. Our goal is to collect valuable **market data** and uncover relationships between used car prices and key commodity markets. We develop a tool box using **web scraping of AutoScout24.ch** and integrating with **Yahoo Finance commodity data**, to provide AutoHelvetia AG data-driven insights for:

- **Optimize Pricing Strategies**
- **Gain Competitive Advantage**

1.3 Project Structure

```
project_scraping_CIP_analysis_car_commodity_price/
  Analysis/                # Analysis notebooks and scripts
     RQ1/                 # Research Question 1 script & analysis
     RQ2/                 # Research Question 2 script & analysis
     RQ3/                 # Research Question 3 script & analysis
  Data/                    # Data storage
     API_data_pull/       # API-fetched commodity data & script
     clean_data/          # Processed and cleaned datasets & script
     Scraping/            # Web scraped data and scripts & scraper script
  Documentation.md         # This documentation file
  README.md                # Project overview
  AI_Disclosure.md         # Gen AI usage disclosure and guidelines
  requirements.txt         # Project dependencies
  .gitignore
```

1.4 Initiall Findings

2 Feasibility Research

2.1 Ethical Feasibility of Web Scraping AutoScout24.ch

This web scraping project was evaluated for both technical and legal feasibility. We focused on the academic research context and our analysis of AutoScout24.ch's robots.txt file and terms of service indicates that the project operates within acceptable boundaries for academic research purposes. - **robots.txt Analysis**: - Allowed: General listing pages without filters - Restricted: User account pages (`/de/account/`, `/de/member/`) - Restricted: Filtered search results with specific URL parameters (e.g., `sort=`, `pricefrom=`) - Restricted: Administrative functions

2.1.1 Technical Feasibility

- **Data Extraction**: Ethically extracts vehicle specifications, pricing, and listing details with Scraper and Yahoo Finance API, involving selenium and beautifulsoup.
- **Data Availability**: We found consistent and abundant data, which is appropriate for analysis for both used car listings and Commodity Data

2.1.2 Analytical Feasibility

- **Statistical Methods**: Appropriate statistical methods can be appllied for analysis, including correlation analysis, regression analysis, and time series analysis, etc.
- **Potential Conclusions**: The project can provide potential valuable insights into the relationship between used car prices and commodity prices, helping stakeholders make informed decisions

3 Data Collection

3.1 Web Scraping Implementation

3.1.1 Target Website

- **Primary Source**: AutoScout24.ch (https://www.autoscout24.ch)
- **Target Path**: `/de/autos/alle-marken` (All car listings)
- **Scope**: Used car listings across all makes and models available on the platform

3.1.2 Technical Implementation

3.1.2.1 Core Toolkits

- **Selenium WebDriver**: For browser automation and dynamic content loading
- **BeautifulSoup4**: For HTML parsing and data extraction
- **Custom Parser**: Combines multiple extraction methods(json, html, css, regex) for robustness

3.1.2.2 Scraping Methodology

1. **Pagination Handling**:
   - Iterates through listing pages systematically and click on next page
   - Implements smart navigation with randomized delays (5-15s between pages)
2. **Data Extraction Strategy**:
   - **Primary Method**: Structure-aware parsing using SVG icon titles and sibling elements
   - **Combination of Methods**:
     - JSON structured data extraction
     - CSS class-based element targeting
     - Regular expression fallbacks for critical fields

3.1.3 Data Points Collected

Data Field	Description	Example
`car_model`	Full vehicle make and model	"Volkswagen Golf 2.0 TDI"
`price_chf`	Listing price in CHF	25,900
`mileage`	Vehicle mileage in km	85,200
`engine_power_hp`	Engine power in HP	150
`power_mode`	Fuel/power type	Diesel, Petrol, Electric, Hybrid
`transmission`	Transmission type	Automat, Manuell, Halbautomatik
`production_date`	Production date	2018
`listing_url`	Direct URL to the listing	[Link]

3.2 API Integration

[Document the API integration, including: - APIs used (Yahoo Finance, etc.) - Authentication process - Data retrieval methods - Rate limits and handling]

4 Data Transformation and Cleaning

Across the project we apply a standard data-science cleaning cadence: validate dataframe, coerce types, handle missing values with data quality strategies, and normalize key features before exporting analysis-ready datasets.

4.0.1 1. Autoscout Listing Standardization

Script: `Data/clean_data/Autoscout_Cleaner_Standardizer.py` - **Brand & Model Extraction**: Uses regex-based patterns to parse the `car_model` field into `brand` and base `model` tokens, removing unwanted information (e.g., VW TIGUAN TSI 2.0 S VERSION BERN TOP ZUSTAND vs VW TIGUAN TSI 2.0 S). - **Model Normalization**: Applies a replacement dictionary to outlier variants (e.g., "TESLA Model Y" → "Model Y"). Keeping consistent models during analysis. - **Field Selection & Export**: Outputs a curated schema (`brand`, `model`, `car_model`, pricing, powertrain, and URL fields) to `Autoscout_Cleaned_Standardized.csv`, preserving only analytics-ready columns.

4.0.2 2. Commodity Price Cleaning

Script: `Data/clean_data/load_data_cleaning.py` - **Type Coercion**: Converts `Date` to datetime and commodity columns to numeric by replacing European decimal separators. - **Missing Value Strategy**: - Reports gaps before/after processing for clarity. - Fills missing commodity prices with a 7-day rolling mean, then rounds to two decimals. - **Temporal Date Standardization**: Generates a formatted `Date` string (`%d-%m-%Y`) for later joins and saves the cleaned series as `Data/Final Data/yahoo_spot_cleaned.csv`.

4.0.3 3. Scraper Output Post-processing

Script: `Data/Scraping/Scraper.py` - **Schema Consistency**: The scraper normalizes fuel types, transmission labels, and numeric fields ("N/A" fallbacks) to reduce later data cleaning steps. - **Hybrid Extraction**: Combines JSON-LD fields with HTML parsing, ensuring critical attributes (`price_chf`, `mileage`, `production_date`, `listing_url`, etc.) are captured.

4.0.4 4. Research Dataset Post-Processing

Script: `Analysis/RQ3/RQ3_Analysis.py` - **Autoscout Cleaning & Missing Value Imputation**: - Converts `production_date` strings (including "Neues Fahrzeug") to October 2025 and makes a `Month` period column. - Imputes continuous fields (`price_chf`, `mileage`, `engine_power_hp`) with rounded means. - Customized fill for `consumption_l_per_100km` (EV=0, then model mean, brand mean, global mean). - Categorical fills (`power_mode`, `transmission`) based on model majority vote, defaulting to "Unknown". - **Commodity Cleaning & Missing Value Imputation**: - Builds monthly periods, fills gaps from daily `Date` entries, and aggregates to one row per month. - Keeps `_Monthly_Avg` features and drops duplicates for consistent joins. - **Merge Output**: Produces `Final_Merged_Data_RQ3.csv`, the **master clean dataset**

5 Analysis and Methodology

5.1 Research Questions

6 Results and Findings

[Present key findings, including:] 1. [Research Question 1] 2. [Research Question 2] 3. [Research Question 3] - Summary statistics - Visualizations - Key insights - Limitations

7 Conclusion and Limitations

[Provide conclusions and potential future improvements]