Web Scraping of Used Honda Cars from Cars24

TEAM - D

BATCH 1009 CP31

TEAM MEMBERS

TEAM - D Batch 1009 CP31

Bhavya P C (Team lead)
Deepti Gupta (Team lead)
DIVYA P S (Co-Team lead)
Dhruvi Mistry (Co-Team lead)
Chandrashekar V
Dewanshu Nagi
Chinnapareddy Vishnu Vardhan Reddy
Devireddy Venkateswara Rao
Chandraprakash Vaishnav
Deepanshu Pawar
Bineesh Mathew

INDEX

PROJECT OBJECTIVE

- Goal: To programmatically scrape used Honda car data from the Cars24 website.
- Target major cities like Mumbai, Bengaluru, and Delhi-NCR.
- Collect key data: Car Name, Price, KM Driven, Year, Fuel Type, Transmission and Location.
- Produce a final, clean CSV file for analysis.

OUR TOOLKIT

- **Selenium**: The main automation library that provides the commands for our script to navigate pages, scroll, and interact with the website like a user.
- **WebDriver**: The core engine of Selenium. It translates our script's commands into actions for a specific browser and is what we used to launch and control Chrome in the background (headless mode).
- **BeautifulSoup**: To parse the complex HTML and accurately extract car details.
- **Pandas**: To structure, clean, and export the final data into a CSV file.

OUR PROCESS

- Setup: Launched a headless Chrome browser using Selenium.
- Scrape: Loaded city URLs and scrolled down to trigger all dynamic listings.
- Extract: Used BeautifulSoup to pull raw text for car details from specific HTML classes.
- Clean: Wrote custom logic to split text, remove symbols (₹), and convert strings to numbers.
- Export: Saved the final, clean data into a Pandas DataFrame and exported it to CSV.

HURDLES WE OVERCAME

Problem: Static vs. Dynamic Data

- Initially, BeautifulSoup alone failed because it couldn't see the car listings, which were loaded dynamically with JavaScript.
- Solution: We used Selenium WebDriver to simulate a browser, fully rendering all the dynamic content.

Problem: Data Rendering Speed

- Even with scrolling, scraping too quickly would miss data that hadn't finished loading yet.
- Solution: Implemented strategic delays (time.sleep) to give the page enough time to render completely before extraction.

Problem: Inconsistent Data

- Data like "₹ 5.5 Lakh" and "45,000 KM" was messy and not ready for analysis.
- Solution: Wrote custom logic and safe parsing to clean and standardize the data into usable numbers.

THE FINAL DATASET

- Successfully created a clean CSV file with 80+ Honda car listings.
- Captured key columns including Car Name, Car Model, Price, KM Driven, Year, Fuel Type, Transmission and Location.
- Data Snippet:

Year	Make	Model	Price	KM Driven	Fuel Type	Transmission	Location_code	Location
2016	Amaze	1.2L I-VTEC VX AT	385000.0	56960.0	Petrol	Auto	MH-01	Mulund West, Mumbai
2022	City	1.5L I-VTEC VX	965000.0	23960.0	Petrol	Manual	MH-48	Goregaon, Mumbai
2018	Jazz	1.2L I-VTEC V CVT	454000.0	37850.0	Petrol	Auto	MH-02	Thakur Village Kandivali E Mumbai
2021	City	1.5L I-VTEC VX	933000.0	22620.0	Petrol	Manual	MH-04	Thakur Village Kandivali E Mumbai
2013	Amaze	1.2L I-VTEC S	204000.0	100000.0	CNG	Manual	MH-01	Goregaon, Mumbai

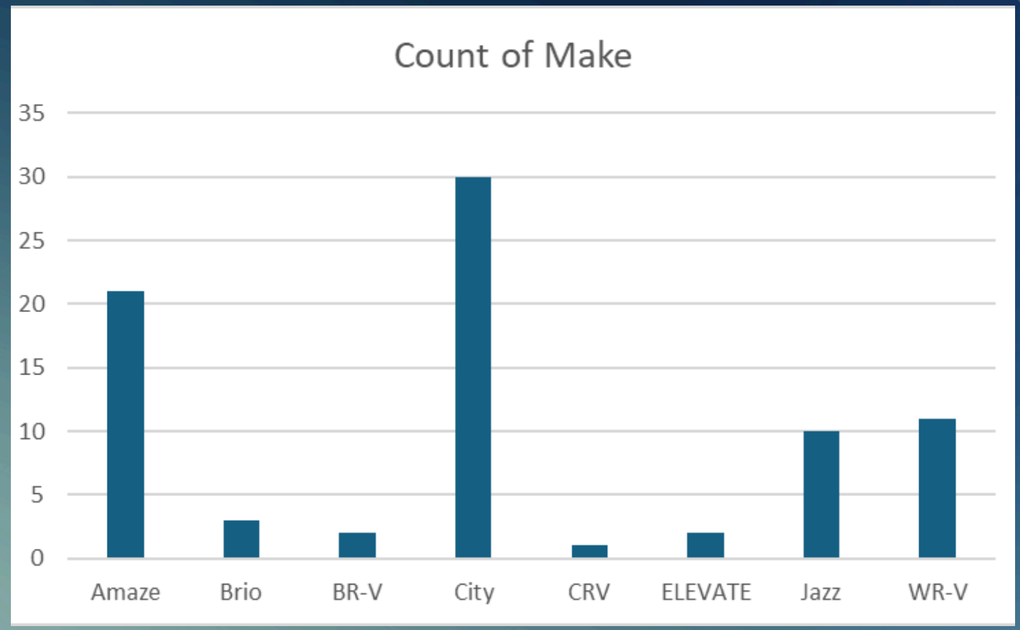
KEY INSIGHTS

- A typical used Honda on the market is a 7.5-year-old Honda City with an average price of around ₹5.6 Lakhs.

- The market overwhelmingly prefers Petrol engines and Manual transmissions, which are the most common vehicle types listed.

- Geographically, the market is led by Delhi-NCR, which has the largest supply of listings, followed by Bengaluru and Mumbai.

- Each city displays unique characteristics: Delhi features newer cars, Bengaluru commands the highest average prices, and Mumbai offers more budget-friendly options.



FUTURE WORK

- Handle infinite scrolling to gather all listings from modern, dynamic websites.
- Extend the scraper to cover more cities and car brands.
- Enhance the scraper to extract owner data from individual car pages.
- Build a machine learning model using the data to predict fair prices for used cars.
- Create an interactive web dashboard to display the results in real-time for users.

CONCLUSION

We successfully built a scraper for a dynamic website and produced a clean, insightful dataset.

THANK YOU