Honda Car Specification Scraping – Team D

Mini Project – Evoastra Internship
Website: AckoDrive.com
Team Lead: Rakshada Renapurkar
Co-Lead: Subham Maharana
Date: 28 November 2025

Team Members:

- Anurag Ojha
- K Sai Kiran
- Kota Aravind Kumar Reddy
- Rakesh Dabbikar
- Nazim Nazir

Problem Statement

Objective: Extract comprehensive variant-level specifications for Honda cars from AckoDrive.com

Key Challenges:

- Dynamic JavaScript-rendered content
- Multi-page navigation required
- Variant-level detail extraction
- 100% data completeness target

Target: All Honda models with "Express Delivery" availability in Mumbai

Project Overview

Scraped: 27 Honda car variants across 3 models
Extracted: 8 key specification fields per variant
Achieved: 100% data completeness (0 missing values)
Cleaned: Professional data pipeline with type conversion
Documented: Complete notebook, report, and presentation

Models Covered:

- Honda City (9 variants)
- Honda Elevate (12 variants)
- Honda Amaze (6 variants)

Parameters Extracted

This project captures the following key specifications:

- Model Name
- Veriant
- Price
- Fuel
- Engine
- Seating Capacity
- Mileage
- Transmission

Technology Stack

Language: Python

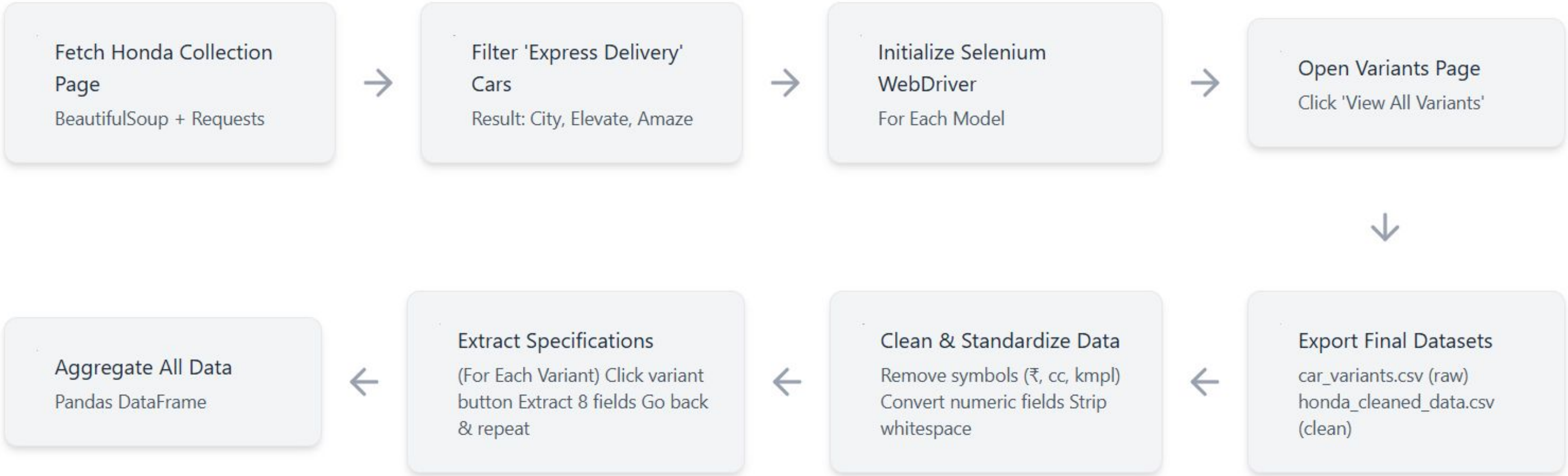
Libraries: BeautifulSoup, Selenium, Requests, re, CSV, Pandas

Tools: Google Colab

Why Selenium? (Not Just BeautifulSoup)

- **Static HTML Parsing**: BeautifulSoup selected for excellent static parsing capability
- **Button Clicks:** Selenium chosen for full button interaction capability
- **JavaScript Execution**: Selenium selected for complete JavaScript support
- **Page Navigation:** Selenium preferred for robust multi-page navigation
- **Dynamic Content**: Selenium chosen for full dynamic content capture

Workflow Diagram



Fetch Honda Collection Page

BeautifulSoup + Requests

Filter 'Express Delivery' Cars

Result: City, Elevate, Amaze

Initialize Selenium WebDriver

For Each Model

Open Variants Page

Click 'View All Variants'

Export Final Datasets

car_variants.csv (raw) honda_cleaned_data.csv (clean)

Clean & Standardize Data

Remove symbols (₹, cc, kmpl) Convert numeric fields Strip whitespace

Extract Specifications

(For Each Variant) Click variant button Extract 8 fields Go back & repeat

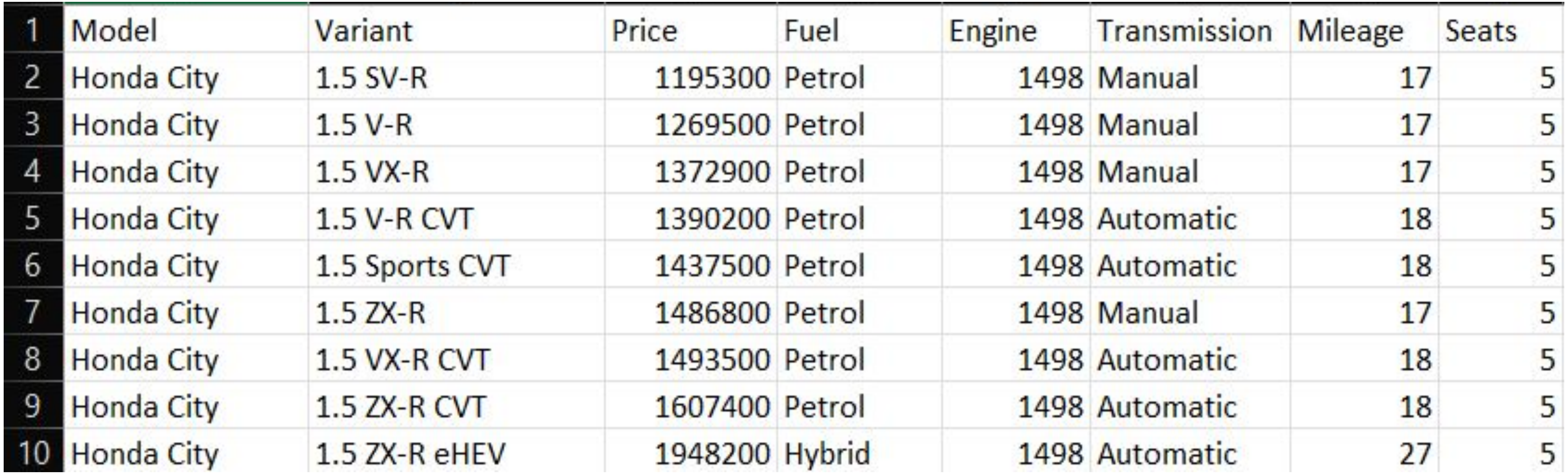
Aggregate All Data

Pandas DataFrame

Step-by-Step Scraping Approach

1. Access Honda collection page
2. Identify & collect all model URLs
3. Fetch each model's specification page
4. Extract engine, mileage, seating, fuel, body type
5. Clean text & normalize outputs
6. Append to master dataset
7. Export final CSV

Dataset Snapshot



1	Model	Variant	Price	Fuel	Engine	Transmission	Mileage	Seats
2	Honda City	1.5 SV-R	1195300	Petrol	1498	Manual	17	5
3	Honda City	1.5 V-R	1269500	Petrol	1498	Manual	17	5
4	Honda City	1.5 VX-R	1372900	Petrol	1498	Manual	17	5
5	Honda City	1.5 V-R CVT	1390200	Petrol	1498	Automatic	18	5
6	Honda City	1.5 Sports CVT	1437500	Petrol	1498	Automatic	18	5
7	Honda City	1.5 ZX-R	1486800	Petrol	1498	Manual	17	5
8	Honda City	1.5 VX-R CVT	1493500	Petrol	1498	Automatic	18	5
9	Honda City	1.5 ZX-R CVT	1607400	Petrol	1498	Automatic	18	5
10	Honda City	1.5 ZX-R eHEV	1948200	Hybrid	1498	Automatic	27	5

Data Cleaning

- **Price Cleaning:** Removed ₹ symbol and commas, then converted to numeric float format
- **Engine Capacity:** Eliminated "cc" suffix, stripped spaces, and converted to integer values
- **Mileage Data:** Removed "kmpl" units and converted to floating-point numbers
- **Seating Capacity:** Stripped "seater" text and converted to integer format
- **Text Fields:** Applied whitespace stripping to all string columns for consistency

Result: Transformed raw scraped data into analysis-ready numeric formats while maintaining data integrity.

Key Insights

Mileage Leaders:

- Overall: Honda City eHEV (27.0 kmpl)
- Petrol Sedan: Honda City Manual (17.0 kmpl)
- Petrol Automatic: Honda City CVT (18.0 kmpl)
- Compact Sedan: Honda Amaze CVT (19.0 kmpl)
- SUV: Honda Elevate Manual (15.0 kmpl)

Transmission Efficiency:

- Manual Average: 16.5 kmpl
- Automatic Average: 17.5 kmpl
- Insight: CVT automatics deliver better fuel efficiency

Challenges Faced

Technical Challenges:

- Stale element references (High frequency)
- Dynamic content loading delays (Medium)
- Discontinued variant filtering (Medium)
- Variable page load times (Low)

Data Quality Challenges:

- Inconsistent text formatting
- Mixed data types (all strings)
- Currency symbols in numeric fields
- Unit suffixes requiring removal

Solutions Implemented

Technical Solutions:

- Re-fetch elements after each navigation
- Strategic 2-4 second waits after clicks
- Detect .discontinued Badge class to skip cars
- Try-except blocks for each specification

Code Quality Solutions:

- Comprehensive exception management
- Progress tracking and debugging logs
- Set-based unique variant tracking
- WebDriver cleanup in finally blocks

Data Quality Solutions:

- Systematic symbol removal with regex
- Explicit type conversion declarations
- Null and duplicate validation checks

Final Deliverables

Code Artifacts

- `Honda_Car_Web_Scraping.ipynb` - Fully documented notebook
- Well-commented Python code with markdown explanations
- Executable from top to bottom

Data Files

- `car_variants.csv` - Raw scraped data (27 rows × 8 columns)
- `honda_cleaned_data.csv` - Production-ready dataset

Documentation

- **Project Report** - 36-page comprehensive technical document
- **Presentation** - Professional slide deck (PDF)

Team Structure

Team Lead – Rakshada Renapurkar

Planning, coordination, documentation

Co-Lead – Subham Maharana

Monitoring, support, assistance

Web Scraping Team:

- Anurag
- Nazim
- Aravind

Data Cleaning Team:

- Rakesh
- Sai Kiran

Report Team:

- Subham
- Rakshada

Presentation Team:

- Rakshada

Conclusion

Project Success Metrics

- 100% Target Achievement - All 27 variants scraped
- 100% Data Completeness - Zero missing values
- 100% Team Participation - All members contributed
- On-Time Delivery - Completed in 4 days
- Quality Documentation - Professional deliverables

What We Demonstrated

- Modern web scraping with browser automation
- Handling dynamic JavaScript content
- Professional data engineering practices
- Ethical data collection
- Team-based development workflow

Team D delivered production-quality code and analysis-ready data

Thank You

Team D – Honda Specification Scraping Project

Delivering clean data with clean code.