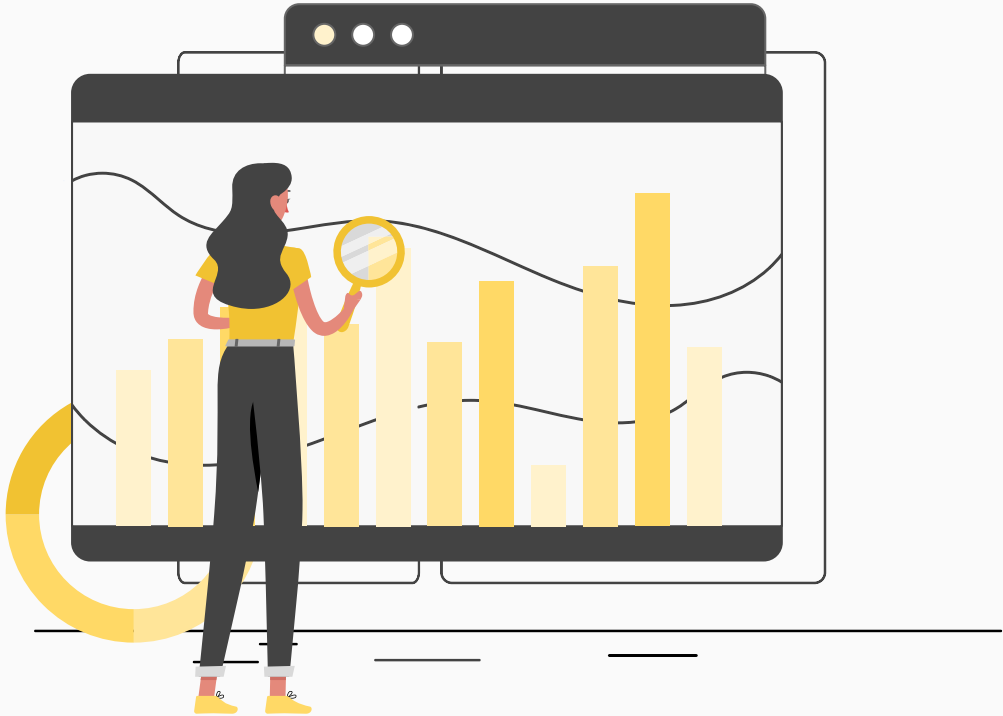
BY:
TEAM L

WEB SCRAPING MINI-PROJECT I

CONTENT

PROBLEM STATEMENT

The goal of this mini-project is to enhance web scraping skills by extracting and analyzing car details from Cars24.com. The focus will be on gathering data specific to the Delhi location. Key details to be collected include kilometers driven, year of manufacture, fuel type, transmission, and price.



METHODOLOGY

In this project, we aimed to scrape and analyze car details from the Cars24 website. The process involved using both Selenium and BeautifulSoup to efficiently extract the required data. Here's a step-by-step explanation of how we achieved this:

- **Environment Setup**:
  - We installed the necessary Python libraries: Selenium for browser automation and BeautifulSoup for parsing HTML content.
- **Initialization**:
  - We set up the Selenium WebDriver (e.g., ChromeDriver) to automate the browser and navigate to the Cars24 Delhi page.
  - The WebDriver was initialized to open the desired URL.
- **Loading the Web Page**:
  - After navigating to the Cars24 Delhi page, we ensured the page was fully loaded. This step was crucial for dynamic content, as it might require additional time to appear.
- **Handling Dynamic Content**:
  - To load all car listings, we scrolled down the page programmatically using Selenium. This ensured that lazy-loaded content was fully visible and accessible for scraping.
- **Extracting Page Source**:
  - Once the content was fully loaded, we extracted the page's HTML source using Selenium. This HTML source was then passed to BeautifulSoup for parsing.

- **Parsing HTML with BeautifulSoup**:
  - BeautifulSoup was used to parse the HTML content. We identified the HTML structure of the car listings and extracted relevant details such as car name, kilometers driven, year of manufacture, fuel type, transmission, and price.
  - We located specific tags and classes that contained the data we needed. For example, car names were found within specific heading tags, while prices and other details were located within list items or div elements.

- **Data Extraction**:

  - By iterating through the parsed HTML elements, we extracted the required details for each car. This included:
    1. **Name**: The car's name extracted from the appropriate heading tag.
    2. **Kilometers Driven**: The distance the car had been driven, found within a list item.
    3. **Year of Manufacture**: The year the car was manufactured, located similarly within a list item.
    4. **Fuel Type**: The type of fuel the car uses.
    5. **Transmission**: The type of transmission (automatic or manual).
    6. **Price**: The price of the car, extracted from a designated price tag.

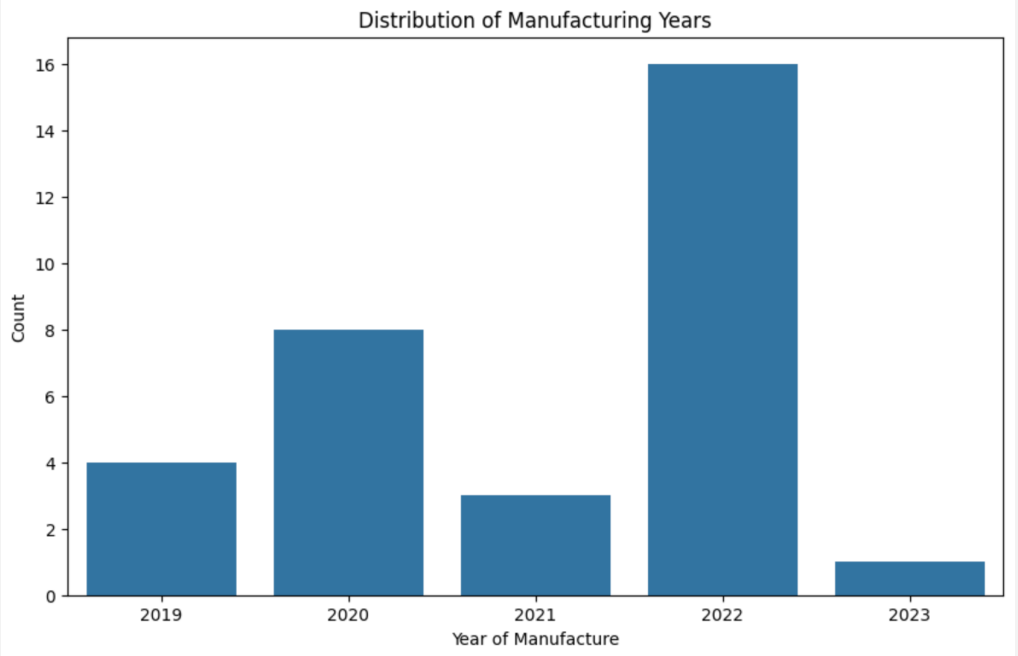
- **Storing the Data**:
  - The extracted data was structured and stored in a suitable format, such as a CSV file or a database, to facilitate further analysis.

- **Analysis**:
  - With the data collected, we performed various analyses to derive insights. This could include statistical analysis, price comparison, or identifying trends based on the kilometers driven and the year of manufacture.

DATA VISULIZATION

Distribution of Car Manufacture in years

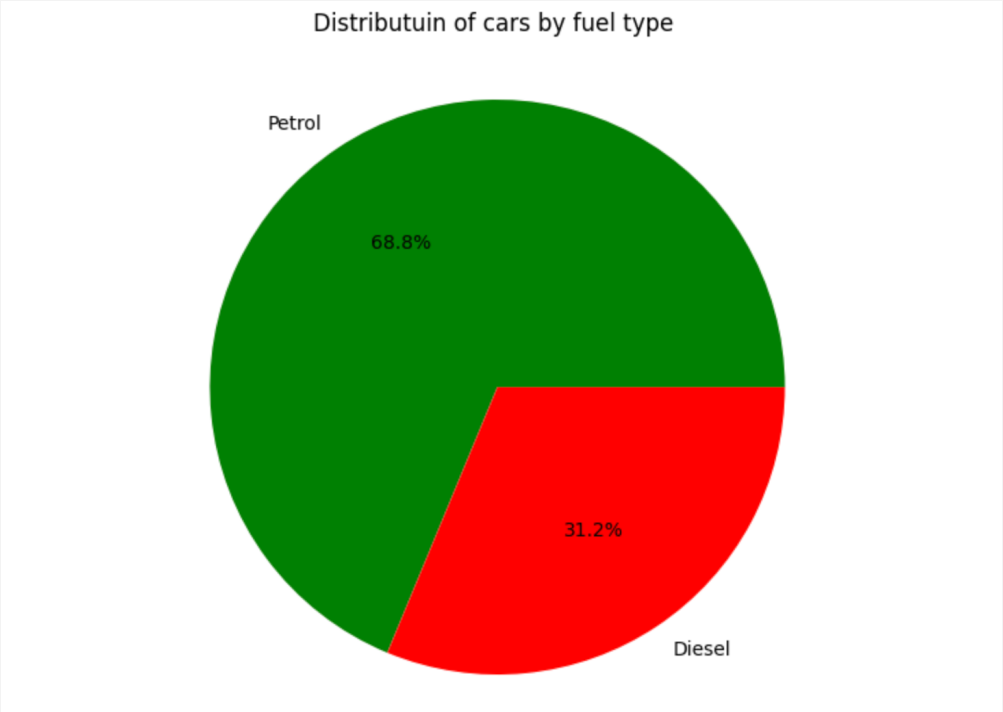
The bar chart represents the distribution of car manufacturing over different years. The x-axis displays the years of manufacture, while the y-axis indicates the number of cars produced in each respective year. Each bar corresponds to a specific year, with its height reflecting the number of cars manufactured during that year.



DATA VISULIZATION

Distribution of Car by fuel type

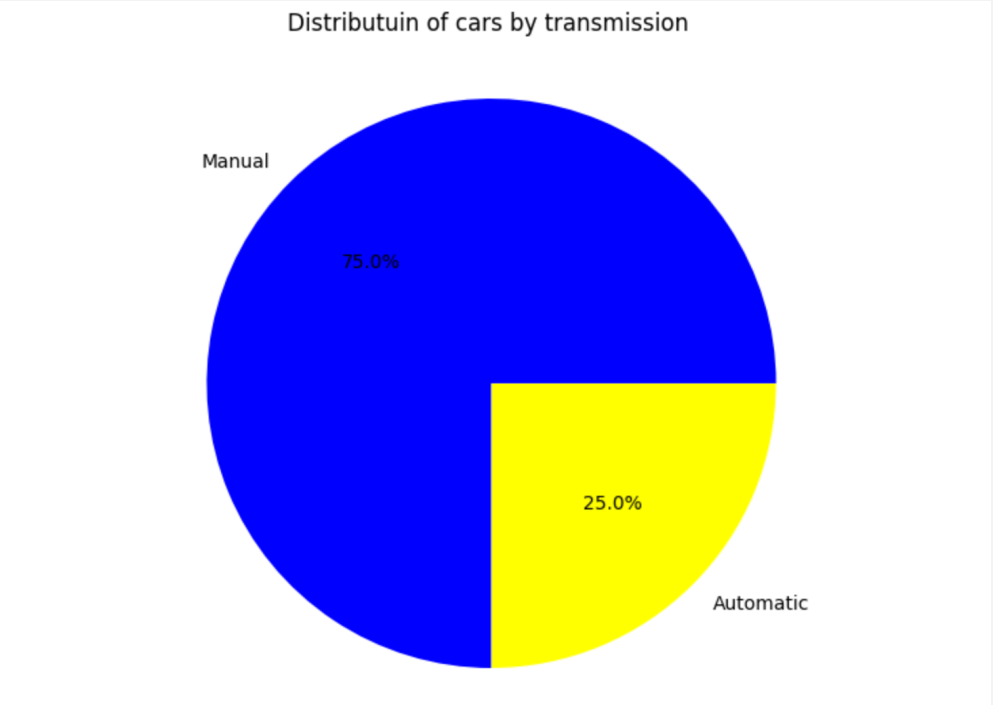
The pie chart represents the distribution of cars by fuel type. It shows that 68.8% of the cars use petrol as their fuel, while 31.2% use diesel. The larger segment of the pie chart is labeled as petrol, occupying more than two-thirds of the chart, while the smaller segment represents diesel, taking up a little less than one-third of the chart.



DATA VISULIZATION

Distribution of Car by transmission

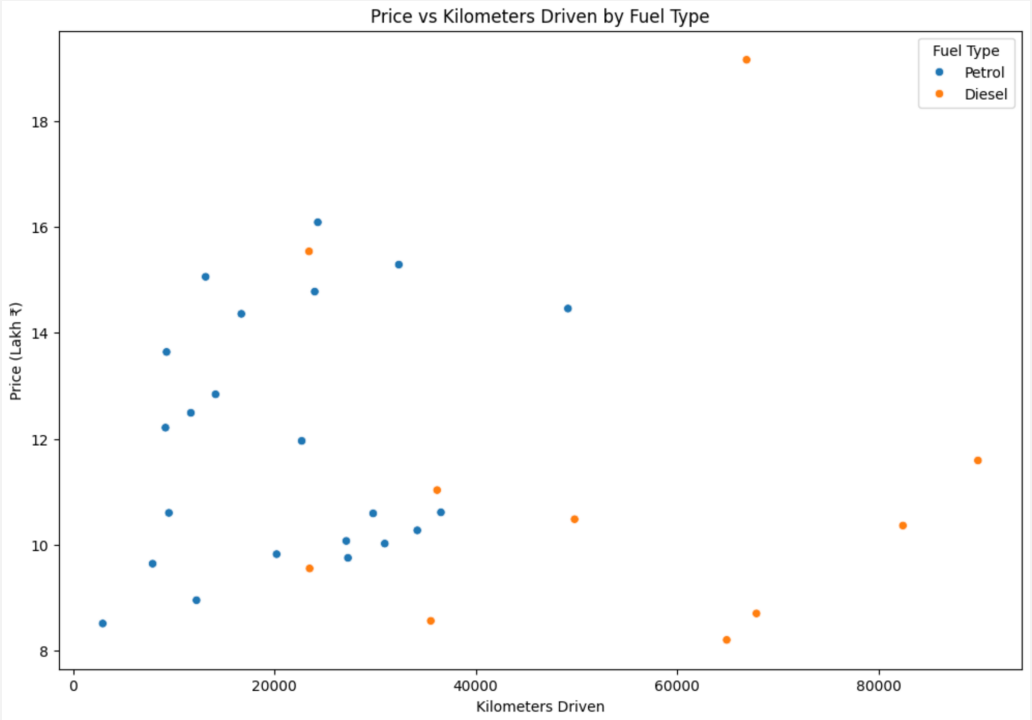
The pie chart illustrates the distribution of cars by transmission type. It shows that 75% of the cars have manual transmission, represented by a larger segment of the pie chart. The remaining 25% of the cars have automatic transmission, represented by a smaller segment. This visual representation makes it easy to see that manual transmission is more common than automatic transmission among the cars in the dataset.



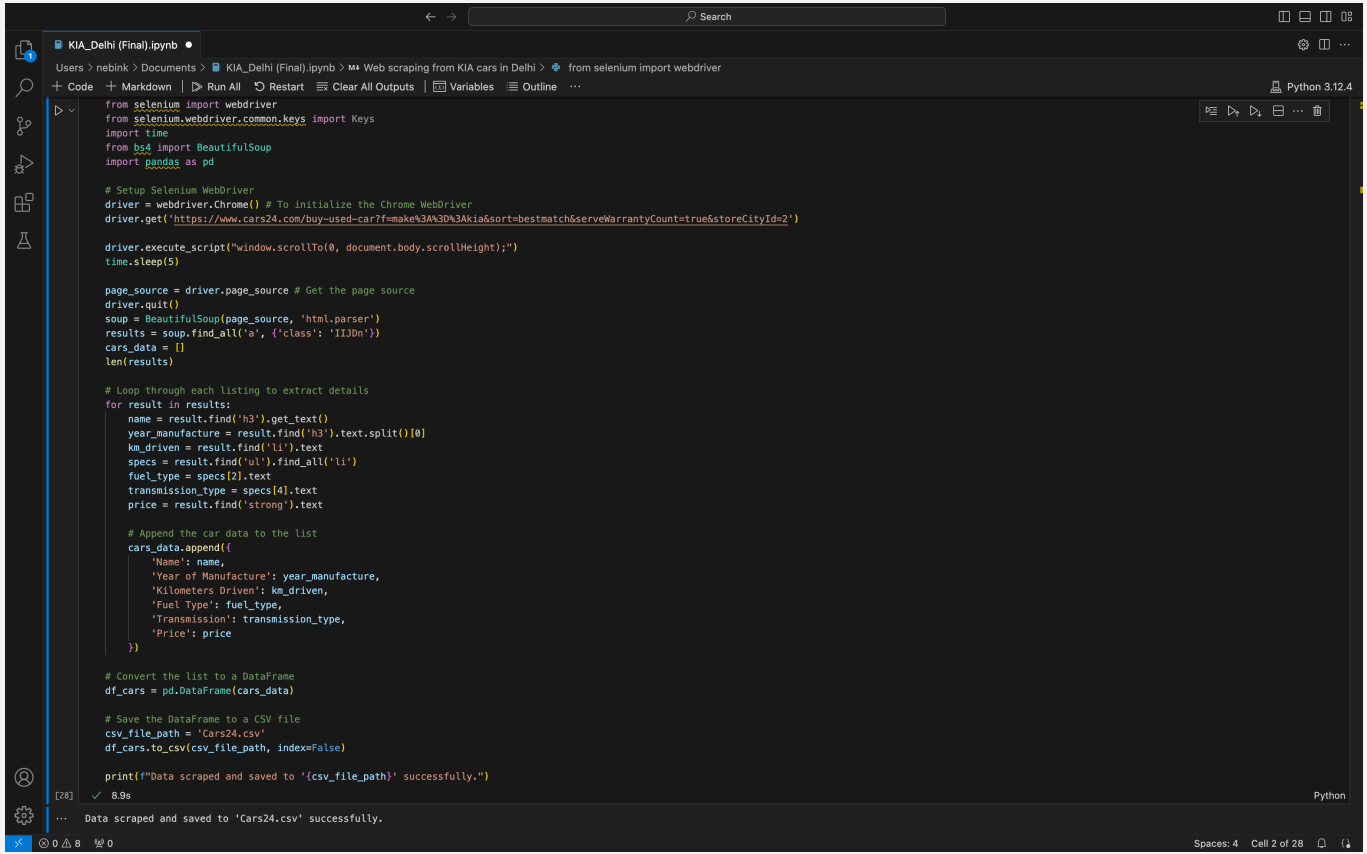
DATA VISULIZATION

Scatter Plot of Car Prices vs. Kilometres Driven by Fuel Type

The scatter plot represents the relationship between the price of cars and the kilometers they have been driven, differentiated by fuel type. The y-axis indicates the price of the cars, while the x-axis shows the kilometers driven. Each point on the plot represents a car, with blue points indicating petrol cars and orange points indicating diesel cars. This visual distinction allows for an easy comparison of how car prices vary with kilometers driven for different fuel types.

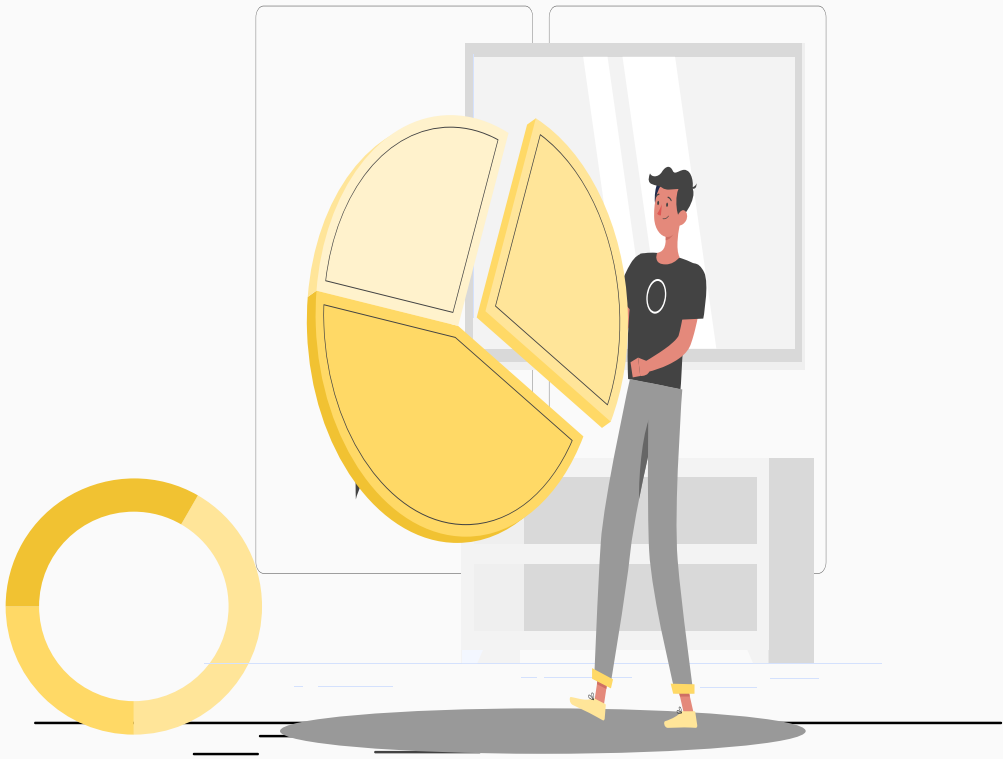


CODE FOR WEB SCRAPING

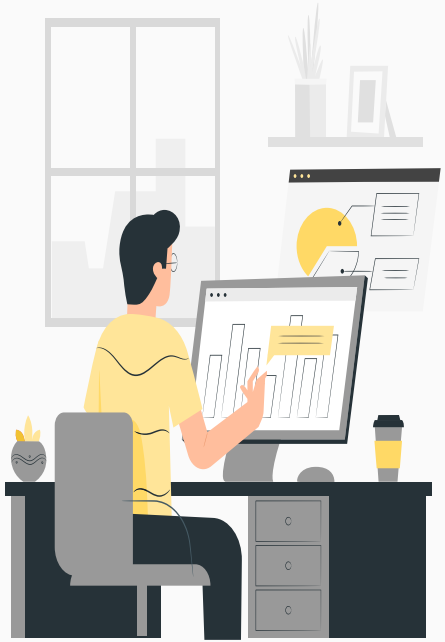


OPINION ON WEB-SCRAPING

As a team of interns, our first experience with web scraping was both challenging and rewarding. Initially, we were overwhelmed by the complexity of automating browser interactions and parsing HTML content. However, as we delved deeper into using tools like Selenium and BeautifulSoup, we began to appreciate the power of web scraping for data extraction and analysis. The process of turning raw web data into structured information was incredibly satisfying, and we gained valuable insights into the intricacies of web technologies. This hands-on experience significantly enhanced our technical skills and boosted our confidence in tackling real-world data problems as a cohesive team.



SKILLS ACQUIRED



WEB SCRAPING

Learned how to use modules like BeautifulSoup and Selenium to Scrap data from the web

DATA HANDLING

Learned how to effectively store and handle data in a csv file

TEAM COORDINATION

Learned how to work together as a team to complete this project

DATA VISUALIZATION

Learned how to use the scrap data and use it to represent the data in a graphical way

CHALLENGES FACED

LAZY LOADED CONTENT

Were not able to extract all the data stored in the page due to some data being stored as lazy loaded content. We overcame that with the use of Selenium

DATA INCONSISTENCY

Ensuring consistency and accuracy across scraped data sources, particularly dealing with variations in formatting and missing data fields from web scraping.

DATA CLEANING COMPLEXITY

Addressing complexities in data cleaning and preparation due to non-standardized formats, such as handling textual anomalies (e.g., removing symbols, converting units) and ensuring data uniformity for meaningful analysis.

CONCLUSION

This web scraping project effectively demonstrates the ability to extract valuable data from dynamic web pages using Python and BeautifulSoup. By targeting a specific website, in this case, ackodrive.com, we successfully gathered detailed information about Kia cars, including car names, prices, kilometers driven, year of manufacture, fuel type, and transmission.

Overall, this project highlights the practical applications of web scraping in real-world scenarios, providing a foundational skill set for automating data collection and enhancing data-driven decision-making processes.