Web Scraping Used Car Data

A project on extracting, cleaning, and analyzing data for used Skoda cars in Mumbai.

Team Members:

Team Leader : Swastika Mondal

Co-Leader : Mudit Mathur

Members:

Ajay Nishad

Surendra Singh

Abdul Rab

Saikiran Satu

Mogal Siddikha Begum

Ajit Kumar

Puskar Chandra

INTRODUCTION

- We, the interns of EVOASTRA VENTURE INC, have undertaken a mini project titled **“Web Scraping Based Used Car Data Extraction from Cars24”** under the guidance of our respected faculty. This project focuses on extracting structured data from the Cars24 website using Python and BeautifulSoup, specifically targeting used Skoda car listings in Mumbai.

- Our team consists of 9 dedicated members who have collaboratively worked on different aspects of the project such as understanding the web structure, writing the scraping logic, cleaning and storing the data, and finally presenting the insights through organized output.

- The project not only helped us understand the real-time applications of web scraping but also enhanced our knowledge in HTML parsing, handling dynamic content, and ethical scraping practices. This hands-on experience has significantly improved our understanding of data extraction techniques and their importance in modern data-driven applications.

Project Overview

Data Collection

Scraping used Skoda car listings from a popular website.

Data Cleaning

Preprocessing raw data for consistency and accuracy.

Data Analysis

Exploring key insights and trends from the cleaned dataset.

Data Visalization

Enhancrd the interpretability of our scraped data, turning raw figures into actionable insights

Step 1: Setup and Request

Install Packages

Begin by installing necessary libraries like 'requests' for HTTP requests.

```
pip install requests
```

Import Libraries

Import 'pandas', 'requests', 'BeautifulSoup', 'numpy', 'matplotlib.pyplot', and 'seaborn'.

```
import pandas as pdimport requestsfrom bs4 import BeautifulSoup
```

Set Target URL & Headers

Define the website URL and user-agent headers for the request.

```
website = "https://www.cars24.com/..."headers = {"User-Agent"
: "Mozilla/5.0"}
```

Send HTTP Request

Execute the GET request to the target website.

```
response = requests.get(website)response.status_code
```

A status code of 200 indicates a successful connection.

Step 2: Extracting Car Details

Parse HTML Content

Use BeautifulSoup to parse the HTML content from the response. This converts the raw HTML into a traversable object.

```
soup = BeautifulSoup(response.content, 'html.parser')
```

Find Car Listings

Locate all car listing elements using their specific HTML tags and classes.

```
results = soup.find_all('a', {'styles_carCardWrapper__sXLIp'})
```

The `len(results)` command helps verify the number of listings found.

Step 3: Data Extraction Loop

Iterate through each car listing to extract specific details such as year, kilometers driven, fuel type, transmission, price, location, and owner information.

```
for car in results:    year = car.find('span', class_='sc-czgmHJ glJeWQ').text.strip().split()[0]    kms = car.find_all('p', class_='sc-czgmHJ ctvwTc')[0].text.strip()    price = car.find('p', class_='sc-czgmHJ frDqEi').text.strip()    # ... append to respective lists
```

Empty lists are initialized to store each extracted data point, ensuring structured collection.

Step 4: Data Cleaning & Preprocessing

Raw data requires cleaning to ensure consistency and usability for analysis. This involves converting text-based values into numerical formats and handling missing data.

1

Parse KM Driven

Convert 'km' strings (e.g., '10k km') to integers (e.g., 10000).

2

Parse Price

Convert price strings (e.g., '₹5 lakh') to integers (e.g., 500000).

3

Capitalize & Handle N/A

Standardize fuel, transmission, and owner fields, replacing "N/A" with None.

Step 5: Final DataFrame & Export

After cleaning, the data is compiled into a Pandas DataFrame, making it ready for analysis. The cleaned data is then saved to a CSV file.

```
clean_df = pd.DataFrame({ 'Year': clean_yrs, 'KM Driven': clean_kms_list, 'Fuel': clean_fuel, 'Transmission': clean_trans, 'Owner': clean_owner, 'Price (INR)': clean_price, 'Location': clean_loc})clean_df.to_csv("cars24_skoda_cleaned.csv", index=False)print("Cleaned data saved to cars24_skoda_cleaned.csv")
```

Step 6: Exploratory Data Analysis (EDA)

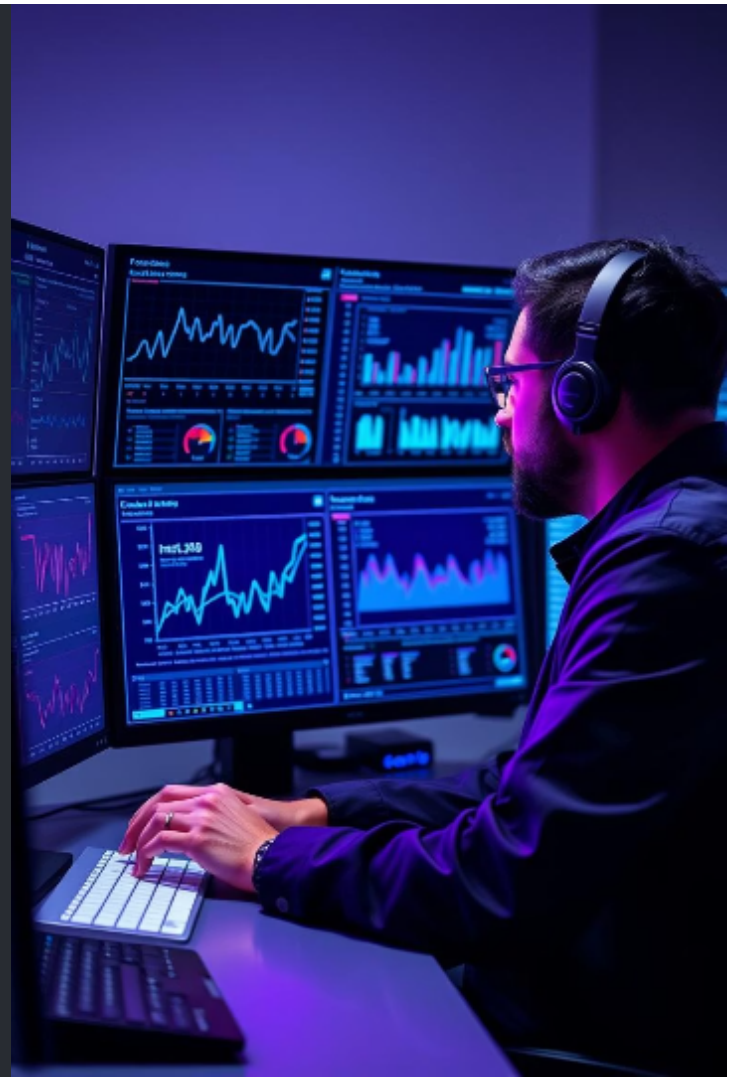
EDA helps understand the dataset's characteristics, identify patterns, and detect anomalies.

Dataset Info

- Shape (rows, columns)
- Column names and types
- Missing values per column
- Number of duplicate rows

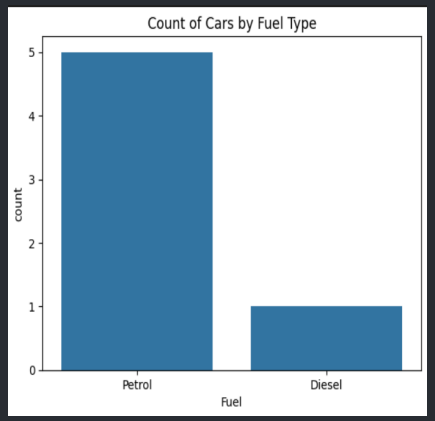
Summary Statistics

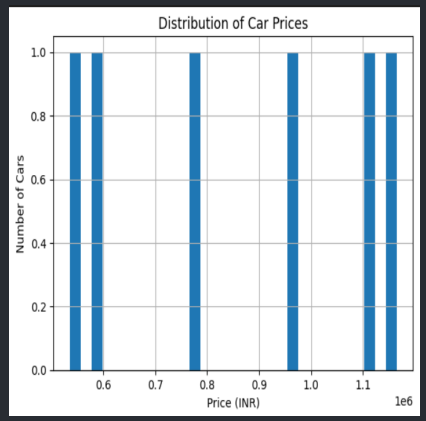
- Statistical summary of numerical columns
- Unique value count for each column

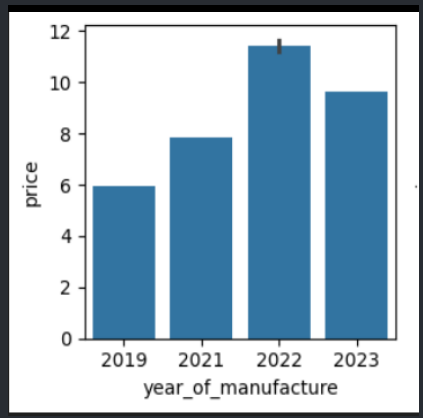


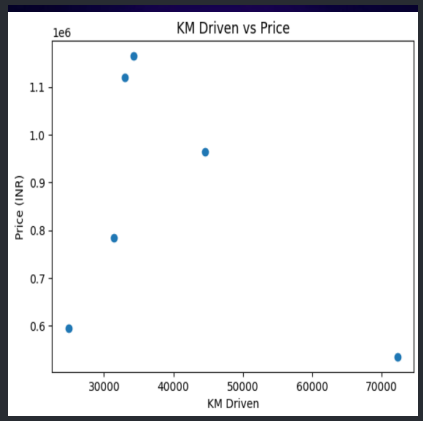
Key Visualizations

Visualizing data helps uncover insights and trends.









- **Fuel Type Distribution:** Petrol cars are more frequently listed than diesel, showing fuel preference trends.
- **Price Distribution:** Most used Skoda cars fall within a specific price range, with noticeable peaks.
- **Year of Manufacture vs Price:** Newer models (2022–2023) generally have higher prices compared to older ones.
- **KM Driven vs Price:** There's an inverse relation — higher mileage cars tend to be cheaper.

Impact, Contribution, Challenges & Collaboration

This project had a signif cant impact on my understanding of real-world web scraping and data analysis by bridging theoretical knowledge with practical implementation. By extracting real-time data from *Cars24.*

we contributed to a realistic dataset that could help analyze car market trends, especially focusing on Skoda vehicles. The insights derived, such as price dependency on fuel type, model year, and mileage, have real-world relevance for both buyers and sellers in the used car market

However, the journey wasn't without challenges — frequent website structure changes, dynamic content loading, and request blocking demanded quick problem-solving and adaptation using tools like BeautifulSoup, requests, and time delays with IP rotation

Despite individual contributions, smooth coordination, timely communication, and knowledge sharing with my teammates helped us successfully complete the project. This collaboration simulated a real software development environment, enhancing our soft skills along with technical growth.

THANK YOU