Project Report: Web Scraping of Used Skoda Cars Data from Mumbai City

1. Objective

- **Goal**: Extract structured and meaningful data from a cars24.com website to enable analysis and decision-making.
- **Use Case**: Gather car details such as price, kilometers driven, manufacturing year, fuel type, transmission type, number of owners, price and location to analyze trends in the used car market.

2. Tools & Libraries

- **Python**: The primary programming language used for its simplicity and extensive library support.
- **BeautifulSoup**: A Python library for parsing HTML and XML documents, enabling easy extraction of data from web pages.
- **Requests**: Used to send HTTP requests and retrieve webpage content.
- **Pandas**: Utilized for organizing, cleaning, and analyzing the scraped data.
- **Seaborn:** is utilized for creating insightful data visualizations.
- **Regular Expressions (re):** are employed for efficient pattern matching and text preprocessing.

3. Scraping Workflow

Step 1: Identify Target Website

- Selected a publicly accessible **cars24.com** website with a clear structure and relevant data.
- Ensured compliance with the website's terms of service and ethical scraping practices.

Step 2: Send HTTP Request

- Used the `requests` library to send GET requests to the target website.
- A GET request was sent to the Cars24 URL (https://www.cars24.com/buy-used-skoda-cars-mumbai/?sort=bestmatch&serveWarrantyCount=true&listingSource=Homepage_Filters&storeCityId=2378) to fetch the HTML content.
- Verified the response status code to ensure successful retrieval of the webpage.
- Get the status code <Response [200]> indicates that the request has succeeded.

Step 3: Parse HTML

- Loaded the HTML content into BeautifulSoup for parsing.
- Inspected the webpage structure using browser developer tools to locate specific tags and attributes containing the desired data.

Step 4: Extract Data

- Used `.find()` and `.find_all()` methods to extract data from specific HTML elements such as `<div>`, `<span>`, <p> and `<a>`.
- Implemented loops to iterate through multiple product listings on the page.
- Extracted key details like Kilometers Driven, Year of Manufacture, Fuel Type, Transmission, Number of Owners, Price and Location.

Step 5: Handle Pagination

- Identified pagination links and iteratively scraped data from multiple pages.
- Automated the process to navigate through pages using URL patterns or Selenium for dynamic navigation.

Step 6: Store Data

- Organized the extracted data into a structured format using Pandas.
- Exported the data to a CSV file ‘used_skoda_cars.csv’ for further data cleaning, analysis and visualization.

4. Column Description

- **Kilometers Driven:** Represents the total distance the vehicle has been driven, measured in kilometers. It's a key indicator of a vehicle's usage and potential wear.
- **Year of Manufacture**: Indicates the year in which the vehicle was manufactured. This helps determine the vehicle's age and can influence its value and condition.
- **Fuel Type**: Specifies the type of fuel the vehicle uses, such as petrol, diesel, electric, or hybrid. This affects the vehicle's running cost, environmental impact, and performance.
- **Transmission:** Describes the type of transmission system in the vehicle—either manual or automatic. This feature can impact driving comfort, fuel efficiency, and market demand.
- **Number of Owners**: Indicates how many individuals have previously owned the vehicle. Fewer owners can suggest better maintenance and higher resale value.
- **Price:** Refers to the selling price of the vehicle, expressed in indian INR. It is influenced by various factors including age, condition, brand, and features.
- **Location:** Represents the geographic location where the vehicle is being sold or listed. This can affect price, availability, and buyer interest due to regional demand or regulations.

5. Issues in Raw Data and Solutions

- **Market Trends**:
    - Identified pricing trends, popular product categories, and seasonal variations.
- **Kilometers Driven:**
    - **Issue:** Contains textual suffix (' km') and abbreviated numbers ('k').
    - **Solution**: Needs cleaning, convert '24.90k km' → 24.90 (float) in thousand.
- **Year of Manufacture:**
    - **Issue**: This is a composite field: contains year + model name.
    - **Solution**: Should be split into Year of Manufacture → 2019 (int) and brand → Skoda (string) and model → Rapid (string).
- **Number of Owners:**
    - **Issue**: Categorical with ordinal information embedded as text.
    - **Solution**: Should be transformed to numeric ordinal (e.g., '1st owner' → 1).

- **Price:**
  - **Issue**: Contains currency symbol and unit (₹, lakh).
  - **Solution**: Needs to be cleaned and converted '₹5.95 lakh' → 5.95 (float) in lac.
- **Location:**
  - **Issue**: Likely to have multi-level location data (property name, area, city).
  - **Solution**: Extract area and city into separate fields for geographic modeling.

6. Challenges & Solutions

Challenge 1: Dynamic Content Loading

- **Issue**: Some data was loaded dynamically via JavaScript, making it inaccessible through static HTML scraping.
- **Solution**: Integrated Selenium to simulate browser behavior and retrieve dynamically loaded content.

Challenge 2: Anti-Scraping Measures

- **Issue**: Encountered anti-scraping mechanisms such as CAPTCHA, rate-limiting, and IP blocking.
- **Solution**:
  - Added headers like `User-Agent` to mimic browser requests.
  - Introduced delays between requests to avoid detection.
  - Used proxy servers to distribute requests across multiple IPs.

Challenge 3: Inconsistent HTML Structure

- **Issue**: Variations in HTML tags and attributes across different pages or products.
- **Solution**:
  - Wrote flexible parsing logic with conditional checks.
  - Implemented error handling to skip problematic entries and log issues for review.

Challenge 4: Data Cleaning

- **Issue**: Extracted data often contained Units and special characters in scraped strings (e.g., "km", "₹"), whitespace, or incomplete entries.
- **Solution**:
    - Applied Pandas functions to clean and extract the data.
    - Cleaned using regular expressions (re.sub) to isolate numeric values.
    - Standardized formats for numerical values and text fields.

Challenge 5: Exploratory Data Analysis

- **Issue**: This dataset produces several issues. It reduced statistical power that leads to unreliable results and conclusions, introduces high variability and noise that limits the ability to draw broader conclusion.
- **Solution**:
    - Prioritize the most important variables to explore and avoids the unnecessary complexity.
    - Utilizes simple graphs or plots that summarizes findings without overcomplicating the analysis.

7. Exploratory Data Analysis (EDA)

- **Univariate Analysis**:
    - Examined individual variables using histplot(), boxplot(), and frequency distributions with countplot() functions of seaborn library.
    - Identified central tendencies and variability with statistical summary by using describe() function of pandas in the data.
- **Bivariate and Multivariate Analysis**:
    - Explored relationships between variables using barplot(), boxplot(), violinplot(), stripplot() functions of seaborn library and correlation heatmaps of dataset.
    - Investigated potential causations or associations between data columns.
- **Outlier Impact**:
    - Visualized the influence of outliers on overall data trends using box plots, violin plots and scatter plots.

8. key Insights from Data

1. **Column Distribution**
    a. **Price** columns are not fully continuous in nature, seems slightly left skewed and no extreme values. spread across a moderate range, peaking around ₹7–11 Lakhs. A couple of cheaper models (₹5–6 Lakhs) exist.
    b. **Kilometer driven** column having Normal Distribution, having same mean and median value of 33 thousand.
    c. **Year of manufacture** column is slightly left skewed, all vehicles are of recent years (2019 - 2023).
    d. **Number of Owners column** Most vehicles have had 1 owner, indicating a first-hand user market.
    e. **Transmission** and **Model** columns appears roughly balanced. **Fuel type**, **Brand** and **City** columns doesn't have categories. Area column having unique values.
2. **Pricing Trends**
    a. First owner of cars are costlier than second. Slavia model car has highest cost. Recently manufactured cars are costlier than older cars.
    b. Fist owner of car with auto transmission having higher price rate than second owner car.
    c. Rapido model car with auto transmission have higher price than manual transmission.
    d. KUSHAQ Model cars in Mulund area are costlier than Seawood area. Rapido Model cars in Goregaon area are costlier than Dombivli area.
3. **Kilometers Driven**
    a. The KUSHAQ model is associated with vehicles that have higher running distances, whereas the Rapid model is generally linked to vehicles with lower running distances.

b. Manual and automatic transmission vehicles exhibit comparable running distances, indicating minimal variation based on transmission type.

c. Newer model-year vehicles tend to have higher running distances compared to older vehicles, suggesting increased usage or improved performance over time.

4. **Manufacturing Year**

    a. KUSHAQ model vehicles are predominantly of recent manufacturing years, indicating a newer fleet.

    b. Vehicles with manual and automatic transmissions exhibit similar manufacturing years, reflecting no significant age disparity between the two transmission types.

9. Conclusion

- Ensured compliance with the website's terms of service and avoided scraping sensitive or personal information.
- Successfully scraped and cleaned used Skoda car listings in Mumbai from Cars24.
- Challenges like dynamic class names were overcome with careful selector usage and regular expressions.
- Insights revealed pricing trends based on year, model and mileage, though more data is needed for robust analysis.

Future Improvements

- **API Integration**: Where available, use APIs for cleaner and more reliable data access.
- **Advanced Scraping Techniques**: Implement multithreading or asynchronous requests to improve scraping speed.
- **Expand Data Collection:** Scrape multiple pages or use APIs for larger datasets. Add **image URLs**, **car condition** (if available), and **dealer information**.
- **Data Enrichment**: Combine scraped data with external datasets for deeper insights.
- **Visualization**: Develop dashboards or visual reports to present findings effectively.
- **Advanced Analysis:** Build a predictive model for used car pricing.

10. Repository & Files

- **Notebook:** skoda_cars_webscraping.ipynb (scraping + EDA).
- **Output Files:**
    - used_skoda_cars.csv (raw scraped data).
    - cleaned_skoda_cars.csv (cleaned data).
- **Presentation:** Web-Scraping-Basics-Project-Demonstration.pptx