

Data Mining Tool & Analytic Project

Overview

This project involves a data mining tool that scrapes vehicle data from a specific auction website. Initially inspired by Peter Kim's R-based project for scraping data from the Steam client, this tool has evolved significantly since its inception in 2019.

After undergoing various iterations and exploring different languages and frameworks, the project has been revitalized using Python, offering enhanced performance and functionality.

Features

- **Data Mining:** The core script (`scrape.py`) is designed to mine data from a selected vehicle auction site. It extracts detailed information about each vehicle.
- **Database Integration:** Scraped data is stored efficiently in an SQLite database.
- **Discord Integration:** The tool integrates with the Discord API to enable automated messaging. A Discord bot sends notifications about new vehicle listings that meet predefined criteria.
- **Dynamic Configuration:** The script supports dynamic year threshold for vehicle selection, allowing it to automatically adjust the criteria based on the current year.

Usage

- Ensure Python 3.11.6 or later is installed.
- Install required dependencies: `pip install -r requirements.txt`.
- Configure environment variables for database and Discord API credentials.

To run the entire project as scheduled:

Run `python scheduler.py`

To run individual scripts, proceed in the following order:

1. Start the data mining process (`scrape.py`):

`python scrape.py` - Mines and saves the vehicle data to the SQLite3 database

2. To check and update the database:

`python check_db.py` - Checks the database with the site and remove any records of vehicles that's no longer available (sold or under offer)

3. To send listings to Discord:

`python to_discord.py` - Sends listings that meet the current criteria; defined in the script

Requirements

- Python 3.11.6+
- Selenium WebDriver for web scraping.
- SQLite for database management.
- Discord API token for bot integration.
- ChromeDriver 119.x.x (compatible with Chrome version).

Notes

- The script requires ChromeDriver to be either in the script's directory or included in the system's PATH. For Windows users, a specific version of ChromeDriver is necessary, while macOS users will need the macOS-compatible version.
- A `.env` file must be placed in the same directory as the script for loading environment variables.
- To update and test functionality with different User Agents, refer to: <https://pypi.org/project/fake-useragent/>

Future Enhancements

- Extending the tool's functionality to support multiple auction sites.
- Improving modularity and script organization
- Enhancing the Discord bot to interactively respond to user queries about vehicle listings.