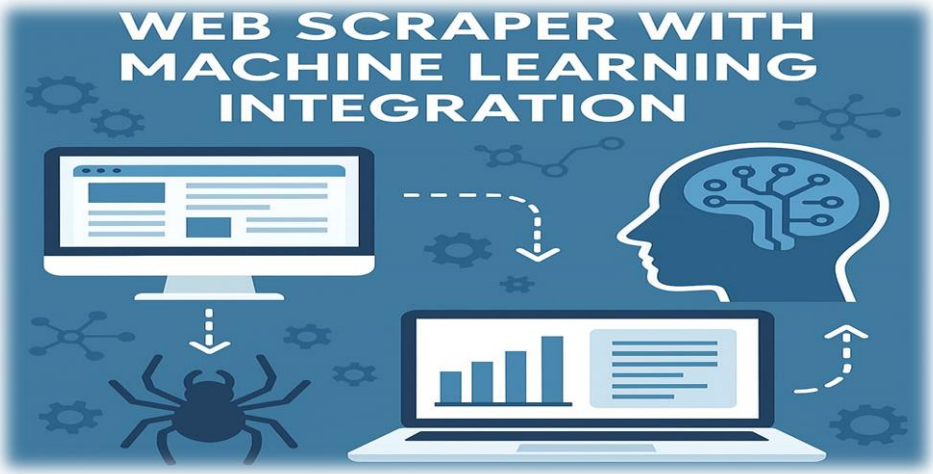
Report: Web Scraper with Machine Learning Integration



Problem Statement

The project aims to build a web scraper to extract book titles and star ratings from 'Books to Scrape'. The data is then analyzed using machine learning to classify reviews into positive or negative sentiment based on ratings and group them into clusters with unsupervised learning. The main challenge is to follow the website's scraping rules while applying mathematical analysis.

How the Web Scraper Works

The scraper sends HTTP GET requests with User-Agent headers to mimic a browser and avoid blocking. It uses BeautifulSoup to parse the HTML and extract book titles (used as review text) and star ratings (converted to numerical values 1–5). A 1-second delay is added between requests to respect server load and follow ethical scraping practices.

Machine Learning Approach and Justification

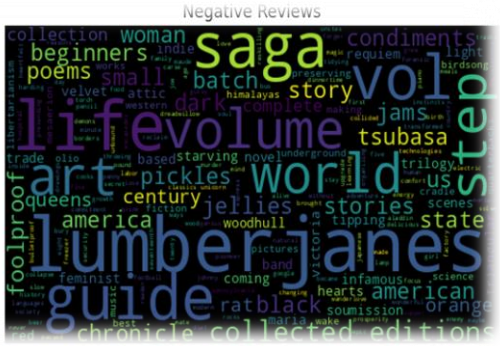
Supervised Learning (Classification):

- Models: Logistic Regression, Random Forest Classifier, XGBoost, SVM.
- Task: Predict if a review is positive (4–5 stars) or negative (1–3 stars).
- Justification:
    - Logistic Regression: Simple and interpretable.
    - XGBoost: Handles complex data and non-linear relationships.

Unsupervised Learning (Clustering):

- Method: KMeans clustering to group reviews without labels.
- Justification: Helps uncover natural groupings and validates sentiment patterns from raw text.

Word Cloud of Positive and Negative reviews



Mathematical Concepts Used

Mathematical Concept	Application
Statistics	Word frequency distribution, TF-IDF feature construction
Linear Algebra	TF-IDF vectors and PCA dimensionality reduction
Probability	Logistic Regression (class probabilities)
Optimization	Gradient Descent for model training (Logistic Regression, SVM)
Distance Metrics	Euclidean distance in KMeans clustering

Key Challenges and Solutions

Challenge	Solution
Handling special characters and stopwords	Applied **regular expressions** and used **NLTK stopword removal** to clean the text.
Avoiding website blocking during scraping	Implemented **polite headers** and **sleep delays** to ensure ethical scraping.
Small dataset due to limited pages	Used **TF-IDF** feature extraction and **regularized ML models** to prevent overfitting and improve model performance.
Visualizing high-dimensional data	Reduced dimensionality using **PCA** to project TF-IDF vectors into 2D for easier visualization of clusters

Conclusion and Future Work:

This project successfully integrates web scraping, text preprocessing, and machine learning. Despite using a small dataset, the models showed strong performance in classification metrics (accuracy, precision, recall).

Future Improvements:

- Increase data volume for better model generalization.
- Tune hyperparameters with GridSearchCV for optimal performance.
- Explore deep learning models (e.g., BERT) for advanced sentiment analysis.
- Create a web app for real-time sentiment predictions based on book titles.