Web Scraping, Sentiment Analysis, and Visualization for Product Reviews

---

1. Overview

This project involves scraping product reviews from Flipkart, preprocessing the data, performing sentiment analysis, and visualizing the sentiment distribution for multiple products. The goal is to identify key products with the most positive or negative feedback.

---

2. Methodology

Phase 1: Data Collection & Web Scraping

- **Objective**: Scrape reviews for multiple products from Flipkart.
- **Process**:
    1. A list of product URLs and names is provided.
    2. For each product:
        - Reviews are scraped using urllib.request and BeautifulSoup.
        - Extracted fields include:
            - **Product Name**: The name of the product being reviewed.
            - **Review Text**: The content of the review.
            - **Rating**: The rating given by the reviewer (if available).
            - **Reviewer Name**: The name of the reviewer (optional).
        - Hyperlinks from the review pages are also collected.
    3. The scraped reviews are stored in a MongoDB collection named raw_reviews.
- **Assumptions**:
    - The structure of the Flipkart review pages remains consistent.
    - Reviews are paginated, and the script iterates through the specified number of pages.

Phase 2: Data Preprocessing

- **Objective**: Prepare the scraped data for sentiment analysis.
- **Process**:
    1. Load the raw_reviews collection from MongoDB.
    2. Use spaCy for:
        - **Sentence Segmentation**: Splitting the review text into sentences.

        - **Word Tokenization**: Breaking sentences into individual words.
    3. Store the preprocessed data in a new MongoDB collection named preprocessed_reviews.
- **Assumptions**:
    - All reviews have valid text fields for preprocessing.

Phase 3: Sentiment Analysis

- **Objective**: Classify reviews as Positive, Negative, or Neutral.
- **Process**:
    1. Load the preprocessed_reviews collection from MongoDB.
    2. Use TextBlob to:
        - Calculate the **polarity** of each review (range: -1 to 1).
        - Classify the sentiment as:
            - **Positive**: Polarity > 0
            - **Negative**: Polarity < 0
            - **Neutral**: Polarity = 0
    3. Store the sentiment results in a new MongoDB collection named sentiment_analyzed_reviews.
- **Assumptions**:
    - TextBlob provides accurate sentiment classification for the review text.

Phase 4: Visualization

- **Objective**: Visualize sentiment distribution for each product.
- **Process**:
    1. Load the sentiment_analyzed_reviews collection from MongoDB.
    2. Group reviews by product and calculate:
        - Total reviews.
        - Count of Positive, Negative, and Neutral reviews.
        - Average sentiment polarity.
    3. Generate visualizations for each product:
        - **Pie Chart**: Shows the percentage distribution of Positive, Neutral, and Negative reviews.
        - **Bar Plot**: Displays the count of each sentiment category.

4. Save the visualizations as PNG files.

---

3. Tools and Libraries

- **Python Libraries**:
    - urllib.request and BeautifulSoup: For web scraping.
    - pymongo: For interacting with MongoDB.
    - spaCy: For NLP preprocessing (sentence segmentation and word tokenization).
    - TextBlob: For sentiment analysis.
    - matplotlib and seaborn: For data visualization.
    - pandas: For data manipulation and analysis.
- **Database**:
    - MongoDB: Used to store raw, preprocessed, and sentiment-analyzed data.

---

4. Assumptions

- The Flipkart review pages have a consistent HTML structure.
- The product URLs provided are valid and accessible.
- MongoDB is installed and running locally on localhost:27017.
- The required Python libraries are installed.

---

5. How to Run the Project

1. **Setup**:
    - Install the required Python libraries:
    - pip install pymongo spacy textblob matplotlib seaborn pandas
    - python -m spacy download en_core_web_sm
    - Ensure MongoDB is running locally.
2. **Execution**:
    - Add the product URLs and names to the products list in the script.
    - Run the script:
    - python sentiment_analysis.py
3. **Output**:
    - The script will:

- Scrape reviews and store them in MongoDB (raw_reviews collection).
- Preprocess the reviews and store them in MongoDB (preprocessed_reviews collection).
- Perform sentiment analysis and store the results in MongoDB (sentiment_analyzed_reviews collection).
- Generate visualizations for each product's sentiment distribution and save them as PNG files.

---

6. Limitations

- The sentiment analysis relies on TextBlob, which may not handle complex or sarcastic reviews accurately.
- The script assumes that the Flipkart review pages are accessible and have a consistent structure.
- The scraping process may be slow due to delays added to avoid being blocked by the website.

---

7. Future Enhancements

- Use a more advanced sentiment analysis model (e.g., VADER or a pre-trained transformer model like BERT).
- Add functionality to handle dynamic websites using Selenium.
- Include additional visualizations, such as word clouds for Positive and Negative reviews.
- Implement error handling for network issues or changes in the website structure.
