**Final Report: Movie Review Analysis Project**

**Introduction**

The Movie Review Analysis Project is a comprehensive data management and analysis initiative aimed at extracting meaningful insights from movie reviews. This project involved web scraping, sentiment analysis, and database management to facilitate a deeper understanding of audience opinions about movies. Using Python and several key libraries, this project successfully demonstrated how data can be collected, processed, and stored to provide actionable insights. The following report provides an overview of the project's objectives, methodology, challenges, results, and future improvements.

**Project Objectives**

The primary goal of this project was to scrape movie reviews from Rotten Tomatoes and perform sentiment analysis on the collected data. Specifically, the objectives included:

1. Extracting movie reviews and associated metadata, such as reviewer names, sources, and review dates.
2. Preprocessing the collected textual data for analysis.
3. Performing sentiment analysis to classify reviews as positive or negative.
4. Storing the processed data in a structured SQLite database for further querying and analysis.
5. Leveraging the data to train and evaluate a sentiment analysis model to refine predictions.

**Methodology**

The project was organized into three main phases: web scraping, data processing, and data storage. Each phase required careful planning and execution to ensure the reliability and accuracy of results.

1. **Web Scraping** The first phase involved extracting movie reviews from Rotten Tomatoes using web scraping tools like requests, pyquery, and BeautifulSoup. The scraper was designed to capture essential information, including the reviewer’s name, source, review text, and date of publication. Special attention was given to handling both positive and negative reviews to maintain a balanced dataset.
2. **Data Processing** Once the reviews were collected, the data was subjected to a series of preprocessing steps. This included cleaning the text, removing stop words, and standardizing the format. Sentiment analysis was then performed using scikit-learn, which involved feature extraction techniques like TF-IDF to convert textual data into numerical representations. The processed data was classified into positive and negative reviews based on predefined criteria.
3. **Data Storage** The processed data was organized and stored in a SQLite database. Three main tables were created:
   * positive\_reviews: Containing reviews classified as positive.
   * negative\_reviews: Containing reviews classified as negative.
   * final\_reviews: A consolidated table summarizing all reviews with their sentiment labels and additional metadata.
4. This database structure ensured efficient querying and easy access to the data for further analysis.

**Tools and Technologies**

The following tools and technologies were critical to the success of this project:

* **Python Libraries**:
  + pandas and numpy for data manipulation.
  + requests, pyquery, and BeautifulSoup for web scraping.
  + scikit-learn for sentiment analysis and machine learning.
  + sqlite3 for database management.
* **Jupyter Notebook**: Used to document the code and ensure reproducibility.
* **SQLite Database**: Provided a lightweight and reliable means to store and query data.

**Challenges and Solutions**

During the development of the project, several challenges arose:

1. **Handling Web Scraping Limitations**: Websites often change their structures or impose restrictions on scraping. To address this, the scraper was designed to adapt to changes in HTML structure, and delays were introduced between requests to avoid being flagged.
2. **Data Quality Issues**: Incomplete or inconsistent review data required robust preprocessing techniques to clean and normalize the text before analysis.
3. **Balancing Sentiment Data**: Ensuring an equal representation of positive and negative reviews was crucial for training an unbiased sentiment analysis model. This was achieved by carefully curating the dataset.

**Results**

The project successfully achieved its objectives. Key results include:

1. **Data Collection**: Hundreds of movie reviews were successfully scraped, encompassing diverse opinions and sentiments.
2. **Sentiment Analysis**: The sentiment analysis model demonstrated strong performance in distinguishing between positive and negative reviews, thanks to robust preprocessing and feature extraction.
3. **Database Storage**: The SQLite database provided an efficient way to store and query the data, facilitating future analysis and exploration.

**Conclusion and Future Improvements**

The Movie Review Analysis Project demonstrated the potential of leveraging web scraping, sentiment analysis, and database management to extract insights from textual data. The integration of various Python libraries and tools ensured a seamless workflow, while the use of SQLite allowed for efficient data storage and retrieval.

However, there are areas for improvement and expansion:

1. **Scalability**: The current scraper could be extended to cover additional review platforms or larger datasets.
2. **Model Enhancement**: Incorporating deep learning models, such as those based on transformers, could improve the accuracy of sentiment analysis.
3. **Visualization**: Adding data visualization capabilities to present findings in an interactive and user-friendly format would enhance the project's usability.
4. **Automation**: Automating the pipeline for periodic data collection and analysis could provide more dynamic insights over time.

In summary, this project not only fulfilled its objectives but also provided a solid foundation for future work in data analysis and machine learning. The tools and methodologies employed are adaptable and can be extended to other domains where textual data analysis is required.

For the conclusions of all the charts, please see the uploaded folder.