Name – Shruti Asolkar
Date – 9-Dec-2024

Final Project Report - Individual

"Insight-Scan" – A Sentiment Analysis Tool for Google App Store Reviews

The Link to the Project's Public GitHub Repository - https://github.com/shrutia003/insight-scan

Project Abstract:

This project introduces the implementation of a tool that automates the collection and sentiment analysis of app reviews from the Google Play Store. The major purpose was to speed up the extraction and classification of user feedback into positive, negative, and neutral labels, allowing developers to prioritize crucial comments and improve user satisfaction. The report describes the problem in depth, proposes a solution, presents experimental results, and makes recommendations for further work.

Introduction:

With millions of applications accessible on the Google Play Store, developers rely on user feedback to determine customer happiness and identify areas for improvement. Manually assessing these reviews is time-consuming and inefficient, particularly for popular applications with thousands of ratings. The goal of this project is to automate the procedure through web scraping and machine learning. By collecting app reviews, classifying their sentiments, and saving the results, the project provides actionable insights to developers.

Problem Statement:

App developers encounter difficulties in handling and evaluating the large volume of user feedback. The key issues are:

- **Volume of Data**: Manually sorting through thousands of reviews is labor-intensive.
- **Prioritization**: Sentiment analysis is necessary for prioritizing critical feedback, such as bug reports or feature requests.
- **Timeliness**: Delays in discovering and addressing issues can harm user retention and app ratings.

This project addresses these difficulties by automating review gathering and sentiment analysis, thereby minimizing manual work and offering timely insights.

Proposed Solution:

The solution has two major components:

1. **Web Scraping**: Selenium is used to collect reviews from the Google Play Store based on dynamic content. The user enters the app's name, then the scraper navigates to the review area to gather input.
2. **Sentiment Analysis**: Preprocessed reviews are classified as positive, negative, or neutral using machine learning models.

Name – Shruti Asolkar
Date – 9-Dec-2024

Pipeline Overview:

- Input: App name.
- Output: Two CSV files containing raw reviews and sentiment classifications, and a text file summarizing evaluation metrics.

Implementation:

Tools and Technologies

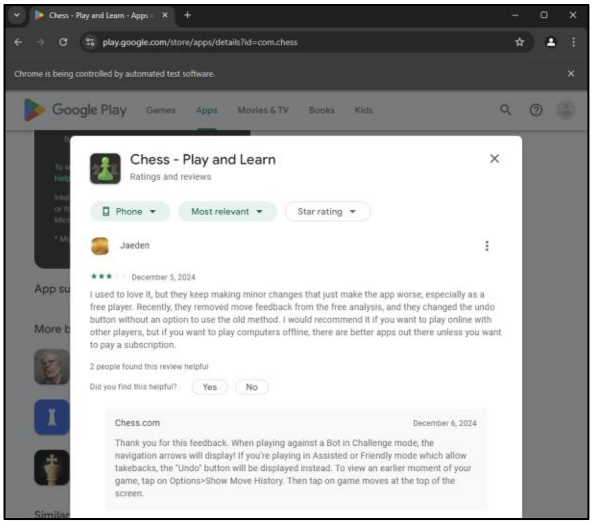
- Python: Primary programming language.
- Selenium: Automates browser interactions for web scraping.
- scikit-learn: Provides the machine learning library for sentiment analysis.
- Natural Language Toolkit (nltk): Prepares text data for machine learning.
- Pandas: Handles data manipulation and storage.

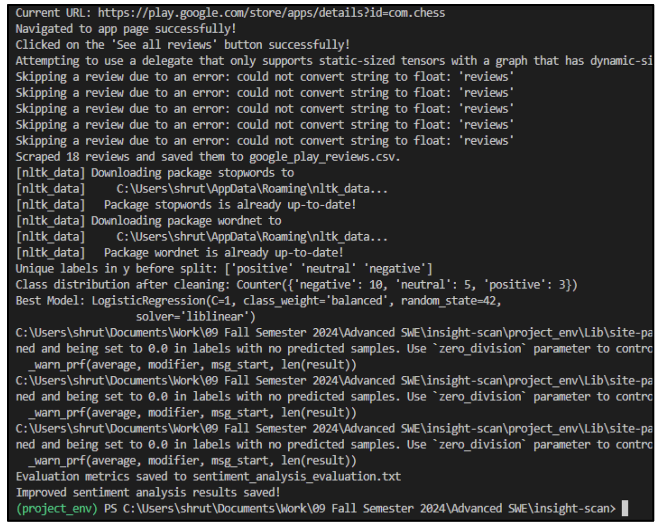
Steps

1. **Web Scraping**: Using Selenium, the scraper collects reviews from the app's Google Play page.
2. **Text Preprocessing**: The reviews undergo tokenization, removal of stop words, and vectorization.
3. **Model Training**: A machine learning model is trained using pre-labeled data to classify reviews into sentiments.
4. **Evaluation**: The model's performance is assessed using metrics such as accuracy, precision, recall, and F1-score.
5. **Output**: Results are stored in structured CSV files and summarized in a text file.

Screenshots of web-scraper in action:

Name – Shruti Asolkar
Date – 9-Dec-2024





Experimental Results:

The evaluation of the sentiment analysis model yielded remarkable results, highlighting both strengths and places for further development. The dataset utilized is comprised of user reviews extracted from the Google Play Store for various apps such as Chess and Duolingo. The reviews were classified as positive, neutral, or negative using the machine learning process.

Results Summary:

- **Accuracy:** The model had an overall accuracy of 75%, correctly identifying three out of four reviews in the dataset. This is a screenshot of some of the sentiment analysis script's sample results.
    - The first column displays clear, tokenized reviews with no stop words. It also displays the sentiment in the second column and the predicted sentiment in the last column.

price point far high discouraging new player would like learn playing update literally dozen documented loss due running time still time clock attempted get support issue literally 15 minute fighting bot able get prompt human assist 3 day app support flawed clock pricepoint high absolutely allowed app store	negative	negative
who idea remove back forth button playing bot single feature change ruined whole experience im actually struggling go back app enjoy game like see thing evolve make hard fun ton online reviewer also hate feature passionately idiotic	negative	negative
used app platinum subscriber 23 year time price doubled there even higher subscription tier necessary able review game way could update like chess great app gotten greedy casuals either stay free accept limited puzzle manual review expect fleeced	neutral	negative
great app playing chess subscription expensive need drop upper tier lower especially keep mind lichess offering similar thing without subscription really bummed see deal black friday still hoping maybe come fruition	positive	positive

- **Classification Report:**
    - The **negative** class demonstrated a precision of 67%, with a flawless recall of 100%, which resulted in an F1-score of 80%.

Name – Shruti Asolkar
Date – 9-Dec-2024

- A recall, precision, and F1-score of 0% resulted from the unsuccessful prediction of the **neutral** class. This implies that the model has trouble with the neutral category, which is a typical problem in simpler sentiment analysis models because it is ambiguous.
- The **positive** class demonstrated the model's strong ability to accurately detect positive emotion with flawless precision, recall, and F1-score of 100%.

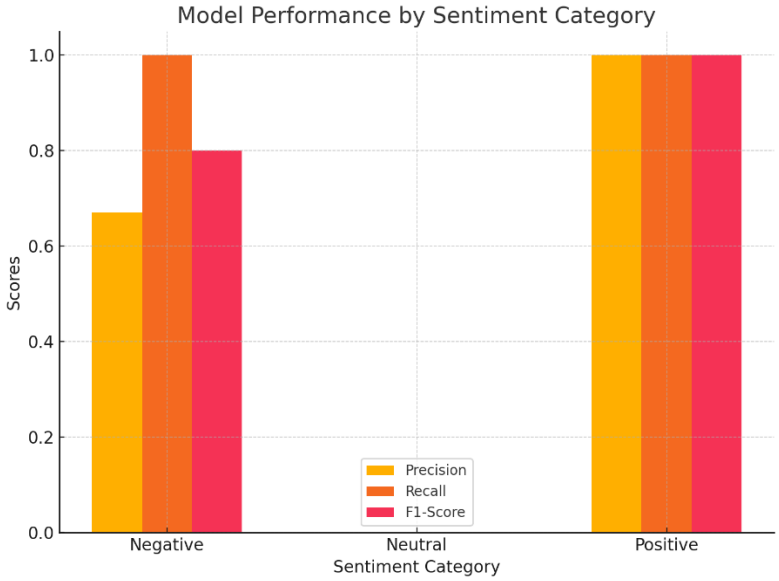
- **Confusion Matrix:**

```
Confusion Matrix:
[[2 0 0]
 [1 0 0]
 [0 0 1]]
```

- The confusion matrix shows that one neutral review was mistakenly categorized as negative, two negative reviews were accurately classified, and one positive review was correctly identified.

Graph to visualize the evaluation results for the model:

The sentiment analysis model's performance in three categories—negative, neutral, and positive—is depicted in the graph. The model performs well in the category of positive sentiment, correctly identifying all positive sentiments without missing or producing false positives. Comparably, the Negative sentiment category does very well, indicating that although the model has a high recall (capturing all true negative occurrences), it occasionally incorrectly labels other sentiments as negative (lower accuracy). With all measures at 0.00 for the Neutral category, the model, however, suffers greatly with this category and is unable to detect neutral thoughts at all. This discrepancy reveals a serious flaw in the model's generalization across sentiment categories, indicating that more training or dataset rebalancing may be necessary to enhance its performance on difficult or underrepresented categories like Neutral.



Name – Shruti Asolkar
Date – 9-Dec-2024

Insights:

- The findings show that polarized sentiments, whether positive or negative, may be strongly detected, but they also show that neutral sentiments are harder to identify.
- The model performs in line with typical trends seen in sentiment analysis systems, which is not surprising considering that neutral sentiments are more difficult to categorize because of their subtlety.

Suggestions for Improvement:

- **Data Augmentation:** Increase the number of reviews scraped in order to increase the quantity of neutral sentiment examples in the dataset to enhance the model's learning of this class.
- **Refinement of Features:** Use combinations of two or three words together (bigrams or trigrams) instead of single words to better understand the context of the text.
- **Hyperparameter Tuning:** Adjust the settings of the machine learning model to improve its performance and handle all sentiment categories more effectively.

Conclusion:

This project effectively illustrates how sentiment analysis of app evaluations can be automated using web scraping and machine learning. The outcomes provide developers with a dependable method to prioritize input and demonstrate notable time savings. But there is still opportunity for development, especially in the areas of managing unbalanced datasets and enhancing neutral review classification accuracy.

Future Work:

1. **Dataset Expansion:** Incorporate reviews from other app stores such as Apple's App Store for more comprehensive analysis. Also, figure out a system to scrape as many reviews as possible from a single page.
2. **Deep Learning Models:** Experiment with neural networks like LSTMs for better sentiment classification.
3. **Feedback Categorization:** Extend the model to identify specific feedback categories, such as bug reports or feature requests.