DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE & ISO 9001:2008 Certified)

Accredited by National Assessment & Accreditation Council (NAAC) with 'A' grade

Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-111

A Project Report On

Comparative Analysis of Amazon and Flipkart Reviews

Submitted in partial fulfillment for the award of the degree of

Bachelor of Engineering

in

Computer Science and Engineering

Submitted By

JOEL A	**1DS21CS095**
KARTHIK K H	**1DS21CS096**
KATTA NAVYA	**1DS21CS097**
M KAVACIN	**1DS21CS111**

Under the guidance of

Prof. Akshatha G
Assistant Professor,
CSE, DSCE,
Bengaluru

Vaibhav Varma
Co-Guide,
Mercedes Benz,
Bengaluru

2024 - 2025

Department of Computer Science and Engineering

DAYANANDA SAGAR COLLEGE OF ENGINEERING

Bangalore - 111

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Dayananda Sagar College of Engineering

(An Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE & ISO 9001:2008 Certified)

Accredited by National Assessment & Accreditation Council (NAAC) with ‘A’ grade

Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-111

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project entitled **Comparative Analysis of Amazon and Flipkart Reviews** is a bonafide work carried out by **JOEL A** [**1DS21CS095**], **KARTHIK K H** [**1DS21CS096**], **KATTA NAVYA** [**1DS21CS097**] and **M KAVACIN** [**1DS21CS111**] in partial fulfillment of 7th semester, Bachelor of Engineering in Computer Science and Engineering under Visvesvaraya Technological University, Belgaum during the year 2024-2025

Prof. Akshatha G	**Dr. Ramesh Babu D R**	**Dr. B G Prasad**
Assistant Professor	Vice Principal & HOD	Principal
CSE, DSCE	CSE, DSCE	DSCE
Signature:...........	Signature:...........	Signature:...........

Name of the Examiners: Signature with date:

1........................... ....

2............................. ....

Acknowledgement

We are pleased to have successfully completed the project **Comparative Analysis of Amazon and Flipkart Reviews**. We thoroughly enjoyed the process of working on this project and gained a lot of knowledge doing so.

We would like to take this opportunity to express our gratitude to **Dr B G Prasad**, Principal of DSCE, for permitting us to utilize all the necessary facilities of the institution.

We also thank our respected Vice Principal, HOD of Computer Science & Engineering, DSCE, Bangalore, **Dr. Ramesh Babu D R**, for his support and encouragement throughout the process.

We are immensely grateful to our respected and learned guide, **Prof. Akshatha G**, Assistant Professor, CSE, DSCE and our co-guide **Vaibhav Varma**, for their valuable help and guidance. We are indebted to them for their invaluable guidance throughout the process and their useful inputs at all stages of the process.

We also thank all the faculty and support staff of Department of Computer Science, DSCE. Without their support over the years, this work would not have been possible.

We express our sincere thanks to Project Coordinator **Dr. Ramya R S, Assoc. Prof**, **Dr. Annapoorna B R Asst. Prof.**, **Prof. Aparna M Asst. Prof** and **Prof. Kanchana M Dixit Asst. Prof** of the Department of Computer Science and Engineering for their continues support and guidance. We thank all teaching and non-teaching staff of the Department of Computer Science and Engineering for their kind and constant support throughout the academic Journey.

JOEL A 1DS21CS095
KARTHIK K H 1DS21CS096
KATTA NAVYA 1DS21CS097
M KAVACIN 1DS21CS111

Abstract

The proposed work focuses on analysis of product reviews in Amazon and Flipkart under Electronic, Books, and Food categories. The aim of this study is to ensure that the user is equipped with the necessary tools in regard to machine learning to analyze customer reviews and make sense of the strengths and weaknesses of what each platform has to offer. The starting point of every project is usually data collection and preprocessing, in this case, reviews for certain products are obtained by web scraping Amazon and Flipkart. The raw data is then cleaned and organized in CSV files for the training and testing of some machine learning models for the best prediction results. This process also includes sentiment analysis which uses the Bag of Words model as well as VADER (Valence Aware Dictionary for Sentiment Reasoning) that helps in categorizing reviews as positive, negative or neutral. On the server side, the computation of trained models is done in order to generate comparison metrics. The interface that has been developed with the help of Streamlit, a library, is interactive. The users can choose one of the categories or particular products, so as to get such comparison charts like accuracy score, confusion matrix and bar charts along with accuracy scores of each. As graphical visualization distinguishes their platforms, it provides the mention of data collection for appropriateness. The Organization offers evidence of the shifts in the moods of the current customers as new reviews manipulate the customers. It has practical implications on how it helps the ecommerce systems. For some good reasons, the analysis could be further developed to consider more product categories, and the use of better methods for analyzing sentiments that will enhance usability and insights on the e-commerce platforms.

Contents

1 Introduction **1**

1.1 Overview . . . 1

1.2 Problem statement . . . 2

1.3 Objectives . . . 3

1.4 Motivation . . . 4

2 Literature Survery **6**

3 Problem Analysis and Design **14**

3.1 Analysis . . . 14

3.1.1 Issues With Sentiment Analysis . . . 14

3.1.2 Data Collection Strategy . . . 15

3.1.3 Data processing and assessment . . . 15

3.2 Hardware Requirements . . . 16

3.2.1 Development Environment . . . 16

3.2.2 Data collection and processing Hardware . . . 16

3.2.3 Backend Server . . . 17

3.3 Software Requrirements . . . 17

3.3.1 Operating Systems . . . 17

3.3.2 Programming Languages and Libraries . . . 17

3.3.3 Storage and Database Management . . . 17

3.3.4 Frontend Development Tools . . . 18

3.4 Images . . . 18

4 Implementation **21**

4.1 Overview of System Implementation . . . 21

4.1.1 System Architecture . . . 21

4.1.2 Technology Stack . . . 21

4.1.3 Data Flow and Integration . . . 21

4.2 Module Description . . . 22

4.3 Algorithms . . . 24

4.4 Code Snippets . . . 25

5 Testing **27**

5.1 Unit Test Cases 27

5.1.1 Sample Unit Test Cases 27

5.2 Integration Test Cases 30

5.2.1 Sample Integration Test Cases 30

6 Results **33**

6.1 Results and Analysis 33

6.2 Images 36

6.3 Conclusion of Results 38

6.4 Future Improvements 38

7 Conclusion and Future Scope **40**

7.1 Conclusion 40

7.2 Future Scope 40

List of Figures

3.4.1 System Architecture Diagram 18
3.4.2 Data Flow Diagram 19
3.4.3 Use Case Diagram 19
3.4.4 Sequence Diagram 20
4.4.1 Sending request to backend 25
4.4.2 Asking to scrape data and prepare stats to send to frontend 25
4.4.3 Compound Score converted to positive,negative or neutral 26
4.4.4 Unpacking the backend response in frontend to show stats in the UI 26
6.2.1 Web Scraper for CAAFR 36
6.2.2 Classification Reports for CAAFR 36
6.2.3 Bar Charts for CAAFR 37
6.2.4 CSV Files after scraping 37

1 Introduction

1.1 Overview

E-commerce industry has grown dramatically in the last few years. It has developed online marketplaces such as Amazon and Flipkart that are leaders in this industry. Its impact on consumers altered the way people shopped for goods-those goods were available almost everywhere from electronics, books to groceries, clothing, and everything in between. Thus, hundreds of millions of customers worldwide voice their experience through reviews-these have now become an integral part of e-shopping. Feedback from customers gives great depth into the quality, ease of use, and performance of a specific product as it helps other consumers looking for answers to some purchase questions. With such huge customer feedback on websites like Amazon and Flipkart, extracting useful information remains a challenge. Any review style in any form may be very lengthy and up to a few words. All reviews are subjective, ranging from profoundly positive to extremely negative. This confuses the consumers who would like to decide based on reviews. It is difficult for the businesses to understand this feedback received in reviews, as large amounts of unstructured text data sometimes become tough to identify specific issues or trends. The issues mentioned above will be dealt with by comparing the two reviews from Amazon and Flipkart. Focus will especially be on three categories: electronics, books, and food. It aims to have deeper understandings of the customer's sentiment with insights valuable both for businesses and consumers. Customer reviews will thus be classified into the three main categories: pros, cons, and neutral feedback. This would enable users to understand which aspects of a particular product are praised, criticized, or just acceptable but not exceptional. Another key objective of the project is the automation of the process of the comparison and analysis of reviews. Most traditional comparison methods of reviews are based on manual reading of the feedbacks and interpretations, which is too time-consuming and prone to human bias. To overcome this, we rely on machine learning techniques: Bag of Words and Vader sentiment analysis to process reviews and obtain meaningful insights from them. These methods will enable us to classify reviews by their sentiment, thus giving an extremely clear view of how customers perceive a particular product on both the platforms. The project employs real time web scraping to crawl on Amazon and Flipkart. This therefore assures the analysis is based on the latest customer reviews. This real-time data collection approach reflects the real status of customer satisfaction rather than the archaic or static data. Once all the data is collected and processed, Streamlit is used to build a frontend, which gives it an easily viewable and interactive manner with which users can compare reviews across both platforms, thus making them better able to know which platform is better for purchasing specific products. This would facilitate both consumers and businesses, with this knowledge as regards the thinking or opinions of the customers. Consumers could take advantage of the platform to make more insightful purchases. Businesses could find valuable feedback regarding their products, services, and customer experience alike. Furthermore, this project constitutes a proof of concept that shows how machine learning and data science may apply in improving decision-making processes in e- commerce.

1.2 Problem statement

Online shopping sites revolutionized how customers used to buy: the freedom and flexibility it provided allowed one to see and purchase anything. But such a huge array of products accompanied by reviews becomes too exhaustive for a customer when sifting through all that is on offer. Reviews provide highly significant information about a product's quality and even performance, but they are presented in a scattered and inconsistent manner. This will make it tough for the consumers to quickly compare and decide which product is the best choice. The problem is even more pronounced when comparing products across different e-commerce sites. Amazon and Flipkart are among the biggest e-commerce companies in India, and each of them boasts millions of product listings and millions of customer reviews. However, these reviews are not standardized and hence difficult to measure the customer sentiment of the different platforms. The reviews may differ on one platform from others or that sort of feedback positive, negative, neutral is not well defined and deviates the consumer's attention. Another major issue is that there are just too many reviews on a very popular item. Most customers today read reviews before buying, and when each product page contains thousands of reviews, it is difficult to find relevant or credible feedback. While there exist users leaving longish comments, others write merely vague or extremely brief reviews, which makes it hard to get a real sense of what works for a product and what does not. Of late, for businesses, customer reviews have become a very rich source of feedback on the products, services, and marketing strategies that companies use. Yet, unstructured data with a very large amount of review data is not particularly easy to extract meaningful information from. Businesses are challenged in determining trends in customer feedback-for example, what they find problematic and where there have been many praises. This project will therefore solve these problems through provision of a structured approach to the analysis and comparison of customer reviews for Amazon and Flipkart. This shall be through use of NLP techniques such as Bag of Words and Vader sentiment analysis in grouping reviews into three primary types: pros, cons, and neutral feedback. This will enable the consumers as well as the businesses in understanding what are the major drivers of customer satisfaction and dissatisfaction. Real-time web scraping would rely on the reviews from both streams so that the data used is updated and shows the current feeling of customers. The data will be processed and analyzed for patterns and trends using machine learning algorithms. The results will be presented through a friendly frontend using Streamlit, where user interaction with the data would be possible, and comparison of reviews from products and platforms, etc. This project is likely to have a positive impact on the overall e-commerce experience for both consumers and companies by solving the challenge of review comparison across different platforms and processing an enormous amount of data. Consumers will appreciate clearer, structured insights into the performance of products to enable more informed purchase decisions. Companies will get valuable feedback on their products and services so that they can eventually improve them for greater customer satisfaction and sales boosts.

1.3 Objectives

The main objective of this project is making a data-driven comparative study of customer reviews between Amazon and Flipkart. Within the scope of the project, the following objectives would be covered. Analyzing customer sentiment: The main objective includes classification of the customer reviews under three opinion categories: positive, negative, and neutral. This will be able to derive the overall customer opinion about the products in both platforms. By the natural language processing methods such as Bag of Words and Vader sentiment analysis, it classifies and categorizes the reviews in to relevant categories.

Provide Actionable Insights: Based on the collection and analysis of the reviews, the project will present its results in a way easy for the consumer and business to interpret.

Concluding Findings : The results will be presented on a frontend that has been built using Streamlit, such that it enables users to view the comparison of reviews between different categories and items. The interface will comprise interactive buttons and dropdowns to filter the data by product category (electronics, books, food) or specific products. Users will also be able to view graphs, heatmaps, and accuracy scores that highlight key insights that come out from the sentiment analysis. First, the key objective is to ensure consumers always make an informative choice in the decisions when acquiring the products through the provision of clear and actionable insights on the customer sentiment. The reviews will be categorized into pros, cons, and neutral feedback so that consumers can quickly identify strengths and weaknesses of a product on each platform. This will make it easier for them to choose between the platform that best fits the purchase of specific products, in accordance with customer opinions. Support Business Strategy Development: Businesses can utilize the insights delivered through this project to get deeper knowledge of what the customers prefer, and thus better their offerings. Customer review analysis will allow businesses to know where exactly they perform well and where they can improve. For instance, if complaints over the same issues are frequently received (e.g., poor delivery time or low product quality), an organization can act in the best interest of the customers. It can also be applied to fine-tune marketing strategies and enhance overall customer satisfaction. There is a focus on three categories of products: electronics, books, and food, but it can be extended into this area even more. The potential future expansion might take into account the integration of other categories of products, such as clothing, home appliances, or furniture. The whole model with sentiment analysis can be further improved by refining the algorithms for accuracy and combining the sources with social media feedback or even expert reviews.

1.4 Motivation

The motivation to develop this project comes from the desire to solve a common problem most consumers and businesses are experiencing in the e-commerce space. As a small percent of existing customers puts so much emphasis on reviews to make a purchase decision, millions of other customers rely on reviews in online shopping. Reviews can give a customer insight into product quality and performance; however, the volume and inconsistency of feedback make it hard for consumers to make informed choices. Consequently, often, consumers end up confused between clashing opinions, not knowing the best platform that offers the best product.

As a consumer myself, this has been a constant challenge in trying to sift through hundreds of reviews to even pass judgment on whether or not a particular product is worth purchase. I have often seen that the review itself was either too vague, too detailed, or so obviously didn't address my specific concerns. Sometimes, one platform has better reviews than another, but it's difficult to compare them directly because they're presented in different formats. This frustration inspired me to develop a solution that would streamline the process of comparing reviews across platforms and make it easier for consumers to evaluate products.

From a business perspective, the motivation for this project is that the effective analysis of customer feedbacks is required. Within the same one day, companies like Amazon and Flipkart generate millions of reviews. However, businesses can't analyze all these at their hands. Hence, this project assists in providing a solution through the automation of the sentiment analysis process and giving actionable insights into improving products, services, and the customer experience for businesses. For businesses operating in competitive markets, this kind of feedback can be invaluable in helping them stay ahead of the competition. The project would also help me build more skills in data science, machine learning, and web development. Through this project, I would gain knowledge about techniques related to sentiment analysis, web scraping, and frontend development, which would be extremely helpful going forward with a career in technology and data science.

2 Literature Survery

1. Deep Learning for Sentiment Analysis of E-commerce Reviews (Liu et al., 2020)

Concept/Proposal/Framework: Liu et al. had explored the possibility of applying deep learning models, including CNNs and LSTMs, for sentiment analysis of e-commerce reviews by benching them with traditional machine learning methods like Naive Bayes and Logistic Regression. These modern neural networks were designed to process large volumes of customer feedback, enabling more accurate sentiment classification. Their approach emphasizes the need for deep learning in capturing the non-linear relationships and intricate patterns within textual data that often elude traditional models.

Key benefits: The experiment showed that deep learning models outperform traditional methods significantly, especially in processing large datasets and identifying complex sentiment nuances. CNNs outperform at extracting local features, while LSTMs capture sequential dependencies, thus better understanding the context in customer reviews.

Disadvantages: These models were very effective but required much computational power and extensive training time. Not only that, but they also tend to have overfitting without sufficient training data, which made them less suitable for small-scale projects.

2. Multilingual Sentiment Analysis Using BERT (Zhao et al., 2021)

Concept/Proposal/Framework: Zhao et al. focused on transformer-based models, particularly BERT, for the task of multilingual sentiment analysis in e-commerce. They addressed the issues associated with linguistic diversity in global platforms by pretraining on diverse language datasets and fine-tuning on domain-specific reviews. This framework enabled e-commerce businesses to analyze customer feedback coming from other parts of the world using a unified single model.

Advantages: The model performed exceptionally well in terms of handling variance in linguistic structures, as high accuracy was obtained regarding sentiment classification for various languages. This feature allowed businesses to engage with a global customer base with consolidated feedback analysis.

Disadvantages: Transformer-based models like BERT are highly resource-intensive for computation requirements for training and inference, which becomes a major challenge for smaller businesses. Fine-tuning multilingual models is also data-intensive and time-consuming.

3. Lexicon-based Sentiment Analysis Framework (Wang et al., 2021)

Concept/Proposal/Framework: Wang et al proposed a lexicon-based sentiment analysis framework that could extract pros and cons of products in reviews. Through the exploitation of sentiment lexicons, predefined lists of words with attached sentiment values, the system analyzes textual data into context-specific opinions about products. This approach was based on rules, alternative to machine learning techniques that require much resources.

Advantages: the approach using lexicon is one that achieved high precision and extractability of opinions in predefined contexts. This made it very lightweight and interpretable to people. It was suitable for mainly relatively smaller datasets and domain-specific.

Disadvantages: the approach is not flexible because it requires pre-defined lexicons which make it less effective when dealing with domain-specific language, slang, or newly emerging expressions.

4. Attention-based Neural Networks for ABSA (Kumar et al., 2022)

Concept/Proposal/Framework: Kumar et al. presented an aspect-based sentiment analysis (ABSA) architecture based on attention-based neural networks, which generates fine-grained feedback about specific product attributes such as quality, price, and usability. It applied the mechanism of attention to relevant parts of the text to focus more of the system's power on key sentiment indicators.

Advantages: The model provided fine-grained sentiment insights, enabling businesses to understand customer opinions about specific product features. This level of detail facilitated more targeted improvements and marketing strategies.

Disadvantages: Attention mechanisms require considerable computational resources and may overfit when trained on limited data, reducing their scalability.

5. Cross-Platform Review Analysis on Amazon and Flipkart (Sharma et al., 2020)

Concept/Proposal/Framework: Sharma et al. conducted a comparative analysis of customer reviews on Amazon and Flipkart, on the product electronics. The authors intended to bring out the differences in customers' preferences across the two websites and their tendencies of sentiment.

Advantages: Based on the analysis done, actionable insights have been provided. First, Amazon users value pricing and after-sales service. It states that Flipkart customers value delivery speed and availability. These could be bases for platform-specific marketing strategies.

Disadvantages: The experiment was limited to electronics, so it was not generalizable to other categories of products or domains.

6. Sentiment Preferences for Furniture Reviews (Rao et al., 2021)

Concept/Proposal/Framework: Rao et al. analyzed home furniture reviews from Amazon and Flipkart using machine learning models such as Random Forest and SVM to understand platform-specific sentiment trends.

Benefits: The results indicated major customers' preferences, including the fact that durability was more important in Flipkart and return policies were more critical in Amazon. Vendors could use these insights to engage in appropriate strategic alignment with customer expectations.

Drawbacks: Feature engineering was quite an exhaustive process, leading to additional complexity and time taken to prepare the data for analysis.

7. CNN-GRU Hybrid for Recommendation Systems (Mehta et al., 2021)

Concept/Proposal/Framework: Mehta et al. combined sentiment analysis with recommender systems by integrating CNN and GRU models in order to increase the personalization of product suggestions. The system analyzed customer purchases and reviews in recommending products using user sentiment.

Benefits: Hybrid model with regard to high frequency purchase categories such as electronics and home appliances improved recommendation relevance.

Disadvantages: The system suffered from a sparsity problem in niche categories, leading to the reduced adaptability of the system in less popular domains.

8. Collaborative Filtering with Sentiment Trends (Lee et al., 2022)

Concept/Proposal/Framework: Lee et al. integrated collaborative filtering with sentiment trends to extend sentiment analysis to niche categories such as books and home decoration, thus enhancing recommendation based on user preferences and sentiment insights.

Advantages: Improved user satisfaction since recommendations match both the user preferences and patterns of sentiment.

Disadvantages: The collaborative filtering methods have the problem of cold start, where it is hard to recommend items for new users or for new products.

9. Aspect-Based Sentiment Analysis (Singh et al., 2020)

Concept/Proposal/Framework: Singh et al. constructed an ABSA framework that emphasized certain product aspects such as price, usability, and quality. The system extracted those features from reviews and then performed an independent sentiment analysis.

Benefits: Valuable product-specific strengths and weaknesses were minutely provided to businesses which could help them in streamlining their offerings.

Drawbacks: Involves lengthy and labeled datasets for effective aspect extraction, thereby increasing the time taken during implementation.

10. Machine Learning Framework for Fake Review Detection (Patel et al., 2023)

Concept/Proposal/Framework: Patel et al. proposed a machine learning framework based on Gradient Boosting to identify patterns such as extreme polarity and overly generic language in detecting fake reviews.

Advantages: Achieved high accuracy in detecting fraudulent reviews, maintaining the credibility of e-commerce platforms.

Disadvantages: Genuine reviews with unusual sentiment patterns were occasionally misclassified, affecting model precision.

11. BERT and Social Network Analysis for Fraud Detection (Yadav et al., 2024)

Concept/Proposal/Framework: Yadav et al. combined BERT embeddings with social network analysis for the identification of coordinated fake review campaigns. Their framework simultaneously examined linguistic patterns alongside reviewer behaviors such as review frequency, reviewer clusters, and interactions to detect collusion in the generation of fraudulent reviews.

Advantages: This hybrid approach is highly accurate in detecting complex fake review networks, allowing platforms to better protect their credibility and user trust. Adding BERT, a more advanced contextual understanding, enables the system to detect a sophisticated fraudulent review pattern that might otherwise be missed.

Disadvantages: The approach relied heavily on availability of rich social network data. In the absence of sufficient metadata for reviewers or their interactions, the effectiveness of the system was reduced. Secondly, computational demands for running BERT as well as network analysis were high.

12. Multilingual ABSA with Transformers (Tan et al., 2021)

Concept/Proposal/Framework: Tan et al. worked on multilingual aspect-based sentiment analysis using RoBERTa. It is a transformer-based model. Their system analyzed reviews across various languages and identified sentiment regarding the product attributes, which could be price, quality, or usability.

Advantages: RoBERTa showed significant improvements in managing the highly divergent linguistic structures that were involved in sentiment classification with high accuracy within global contexts. Businesses functioning in multilingual markets could use this system to understand customer feedback across regions without any mess.

Disadvantages: Training and deploying transformer models is computationally intensive, especially in the case of multilingual datasets. It proved to be a hurdle for small-scale businesses. Further, it depended greatly on the quality and availability of good, high-quality, multilingual training data.

13. Dataset Rebalancing for Skewed Sentiments (Ghosh et al., 2021)

Ghosh et al. discussed the problem of skewed sentiment datasets, where a certain class of sentiment overwhelms others. Techniques like oversampling the minor classes and training using weighted loss functions were suggested.

Advantages: These techniques improved the robustness and fairness of sentiment classification models, making the models able to predict accurately even in underrepresented classes like neutral or negative sentiments.

Disadvantages: Oversampling helped in balancing classes but at the cost of introducing artificial biases from duplicate data points. The weighted loss functions were also sensitive and needed calibration to prevent major under-representation in the dominant class.

14. Hybrid Model for Neutral Review Classification (Das et al., 2021)

Concept/Proposal/Framework: A hybrid model proposed by Das et al. used rules with machine learning to classify neutral reviews correctly. It analyzed linguistic cues as well as contextual dependencies to handle ambiguous feedback that was neither overtly positive nor negative.

Advantages: The hybrid approach provided nuances in understanding neutral reviews, enabling businesses to "zero in on subtle customer concerns or preferences that may otherwise go unnoticed.".

Disadvantages: Integration of rule-based and machine learning techniques resulted in higher system complexity, hence requiring continuous maintenance and tuning to guarantee continued consistency.

15. Hierarchical Sentiment Framework (Rajput et al., 2022)

Concept/Proposal/Framework: Rajput et al. developed a multi-level sentiment analysis framework that intends to capture hierarchical structures in customer feedback. The system developed by them had analyzed sentiment on multiple levels including sentences, phrases, and aspects that captured the opinion of the customers in all details.

Advantages: The hierarchical approach provided rich Customer feedback analysis, so businesses could determine their strengths and weaknesses at any granular level. It was very useful for those reviews whose text would convey more than one sentiment.

Disadvantages: Complexity of the model grew exponentially with the size and complexity of the dataset, which made the model computationally expensive to train and time-consuming.

16. Topic Modeling with Sentiment Analysis (Ahmed et al., 2020)

Concept/Proposal/Framework: Ahmed et al. brought together techniques in topic modeling, such as Latent Dirichlet Allocation (LDA), and sentiment analysis for the summarization of massive volumes of customer reviews. Their system was capable of identifying key topics in reviews and assigning sentiment scores to each.

Benefits: This method condensed complex data into useful insights, thus allowing companies to extract significant topics from customer opinions within a very short time. It best used to discover latent patterns or repeated problems within various product groups.

Drawbacks: Subtle feelings were usually hard to find by topic modeling techniques; especially when topics became entangled or reviews contained contradictory opinions.

17. Interactive Dashboard for Sentiment Trends (Sarkar et al., 2021)

Concept/Proposal/Framework: Sarkar et al. designed an interactive dashboard to present sentiment trend and product comparison results. Their system featured dynamic online updates and user- friendly visualizations, like bar charts, pie charts, and heatmaps.

Advantages: Data interpretation made easier to non-technical people, thus automatically making business reviews possible regarding sentiment trends of goods and their performance. Related issues came forth with the help of real-time updates.

Disadvantages: Inability to deliver in-depth analyses on statistics, it would not be the best for users who expect a detailed quantitative analysis.

18. Sentiment Distributions Across Categories (Pavan et al., 2021)

Idea/Proposal/Framework: Pavan et al. (2021) analyzed a distribution of emotions within the different categories of products, which include electronics, groceries, and other fashion categories. According to their study, differences were identified with regards to category-specific emotion trends.

Advantages: Allowed companies to develop category-specific strategy based on preferences, thereby enhancing consumer satisfaction as well as market segmentation.

Disadvantages: This has lower generalization for niche or emerging product categories with less information.

19. Actionable Insights from Neutral Reviews (Reddy et al., 2022)

Concept/Proposal/Framework: Reddy et al. focused on converting neutral reviews into actionable insights by analyzing patterns and underlying sentiments. Their framework categorized neutral reviews based on subtle linguistic cues, enabling businesses to address overlooked concerns.

Advantages: Provided businesses with opportunities to enhance products and services by focusing on aspects that customers were ambivalent about.

Disadvantages: Neutral reviews often lacked sufficient context, making it challenging to derive actionable insights consistently.

20. Pretrained Models for Sentiment Analysis (Chen et al., 2020)

Concept/Proposal/Framework: Chen et al. experimented with the application of pre-trained language models such as GPT on sentiment analysis tasks, emphasizing the role of contextual embeddings in capturing subtle sentiments.

Advantages: The use of contextual embeddings enabled the classification of subtle customer feedback much more accurately than ever before.

Disadvantages: Training the pre-trained models into those specific domains was very extensive and thus increased the training time and computational costs.

21. Comparison of Sequential Sentiment Models (Verma et al., 2021)

Concept/Proposal/Framework: Verma et al. discussed the comparison of sequential deep learning models, such as LSTM and GRU, for sentiment analysis. Their study highlighted the strengths and limitations of each model when it came to processing review datasets.

Advantages: Sequential models can effectively capture dependencies in textual data, making them perfect for analysis within reviews with context-sensitive sentiments.

Disadvantages: Faced challenges in scalability, as these methods were used for very long review datasets, and often the output performance was determined by the quality of the input data.

22. Domain-Specific BERT Tuning (Mishra et al., 2021)

Idea/Proposal/Framework: Mishra et al. fine-tuned BERT for domain-specific sentiment analysis in e-commerce. Their model was designed to understand industry jargon and customer feedback patterns.

Advantages: It was able to offer high accuracy in capturing domain-specific nuances, making it particularly effective for specialized applications.

Disadvantages: Required significant computational resources for fine-tuning and was sensitive to dataset quality.

23. Noise Reduction for Sentiment Accuracy (Chauhan et al., 2022)

Concept/Proposal/Framework: Chauhan et al. proposed noise reduction techniques to enhance sentiment analysis performance by filtering irrelevant or low-quality data.

Advantages: Improved model reliability and accuracy by removing noise from the training datasets.

Disadvantages: Risk of inadvertently filtering out useful edge-case data, reducing the model's flexibility.

24. Feedback-Driven Product Improvement (Nair et al., 2020)

Concept/Proposal/Framework: Nair et al. applied sentiment analysis in identifying areas for product improvement by analyzing customer pain points.

Benefits: Allowed businesses to innovate and address specific customer needs.

Weaknesses: The approach did not include mechanisms by which qualitative data could be integrated with quantitative sentiment scores.

25. Visualization for Decision-Making (Sarkar et al., 2022)

Concept/Proposal/Framework: Sarkar et al. pushed forward data visualization techniques that were enabled through the integration of interactive elements in improving decision-making.

Advantages: Complex datasets simplified into engaging easily understandable visuals, further helping non-technical stakeholders.

Disadvantages: Will have less depth for those who need more elaborate statistical insights or further analytics.

3 Problem Analysis and Design

3.1 Analysis

E-commerce is an industry that has revolutionarily changed the behavior of consumers and how businesses relate with customers. Online sites have become part and parcel of the lives of millions of consumers, and platforms like Amazon and Flipkart have dominated this world of e-commerce. One of the salient features of such platforms is customer reviews where a prospective buyer can attain knowledge about the quality, usability, and performance of the product. Reviews differ significantly in terms of length, detail, sentiment, and helpfulness. This inconsistency presents a challenge to both consumers and businesses and instead of being battered by conflicting reviews from customers, businesses are battling the avalanche of unstructured information to try to drill through.

This project addresses a significant problem especially regarding reviews comparison across channels (Amazon and Flipkart) considering the presence of diversified expectations from the customers, cultural influences, and even linguistic barriers. For instance, the reviews on one platform are mainly based on the experience with delivery, while reviews on the second platform would be based on product feature and post-sale support. Such inconsistency makes it difficult for the consumers to get an accurate and cohesive view of the performance of the product and can lead to decision fatigue.

Another major challenge is the volume of data. A very popular product can have thousands, even millions, of reviews. It's impossible for customers to read and make sense of each one. Over time, this causes patterns and trends in customer sentiment to go undetected by the businesses operating in the market. These reviews are thus rich in potentially extremely valuable feedback and are usually unstructured in form, requiring immense processing to categorize sentiments about the key improvements areas.

3.1.1 Issues With Sentiment Analysis

Sentiment analysis deals with determining user feedback as positive, negative, or neutral. Despite the substantial breakthroughs in technology, its adoption to real-time, unstructured text data derived from numerous sources like reviews is still very challenging. Sentiment analysis models are generally challenged when facing cases of sarcasm, mixed sentiments-where a review incorporates both positive and negative aspects- or domain-specific language-for example, technical terms applicable to electronics. Therefore, the choice of proper models and preprocessing of data is essential for successfully handling such cases.

The relatively straightforward Bag of Words model does quite well in extracting features from texts and has been used extensively for tasks like text classification, which includes sentiment analysis, although it has its own limitations in that words outside of their context appear to be shunned. For example, "good" regarding product could be a good sentiment word when taken alone, but its meaning becomes different based on the rest of the context or sentence. To overcome these challenges, advanced models for sentiment analysis, such as Vader, are deployed, which are highly suitable for short texts, as in this case, reviews. The reason is that such models track intensity, negations, and context much better than the traditional method, BoW.

3.1.2 Data Collection Strategy

Another challenge in the analysis is real-time data collection. The reviews in Amazon and Flipkart are dynamic, as there are new posts made by users. The data collection strategy involves web scraping, which allows for automated extraction of reviews from product pages. To ensure accuracy and timeliness, the scraping framework is designed to handle session IDs, pagination, and dynamic content loading. Scripts intended for web scraping would require optimization to handle high volumes of data. In doing this, one can avoid getting blocked by the website or obtaining an IP ban in case too many requests were made in a time frame that is too concentrated.

3.1.3 Data processing and assessment

Collected reviews are then submitted for processing. Preprocessing of raw text is cleaning the input by eliminating unwanted information from the real input, such as HTML tags and special characters, tokenizing (i.e., often breaking the text into words), and performing any kind of normalization for the text such as converting all text to lowercase, deleting stop words. The processed data is then classified based on sentiment in a model of sentiment analysis, which recognizes pros, cons, or neutral feedback in every single review.

The evaluation of the model is a critical part of this analysis. On performance, the models for sentiment analysis will be evaluated using common evaluation metrics, such as accuracy, precision, recall, and F1-score. For this, the confusion matrix or heatmap could be generated to visually present how well the model will distinguish between positive, negative, and neutral sentiments. Approach: The best model that can exhibit maximum accuracy and reliability in classifying reviews should be identified for actionable insights.

3.2 Hardware Requirements

This project mainly depends upon the complexity of models used for sentiment analysis, volume of data being processed, and whether the systems require real-time data scraping and analysis. The hardware requirements of this project are pretty smooth, considering the simple tasks involved. However, this task should have appropriate hardware as it can provide the easy flow of data processing and train the model efficiently.

3.2.1 Development Environment

A minimum hardware requirement for this project would be the configuration of a standard computer or laptop as below. Processor: A multi-core processor, such as Intel i5/i7 or AMD Ryzen 5/7, should suffice as a bare minimum for computing power in terms of data processing and executing the machine learning models. RAM: At least 8GB for proper processing because most of the processes would be in RAM, especially when dealing with large datasets. For work or models too large, certainly more than 16GB of RAM will be required to prevent any bottlenecks in performance. Storage: Storage should have a minimum capacity of 500GB. SSD would be preferred for this task because the read/write speed will be much faster than it is for the normal hard disk. Graphics Processing Unit (GPU): While sentiment analysis models are not as tight on GPU usage, specific deep learning tasks will benefit from a dedicated GPU. For more basic models like Vader or Bag of Words, a CPU would be enough; however, for more complex models, a dedicated GPU such as an NVIDIA GTX 1050 or better could help speed up model training, especially when using deep neural networks.

3.2.2 Data collection and processing Hardware

In web scraping and collection of real-time data, there are other hardware considerations too, Network Bandwidth Since such real-time web scraping involves the transfer of huge volumes of data between the involved servers, a good internet connection is necessary for the real time collection of data without experiencing any major delays. This is especially essential while collecting reviews from multiple product pages and platforms in real-time. Cloud Computing (Optional): For large-scale data scraping and processing, for instance, cloud computing services such as AWS, Google Cloud, or Microsoft Azure can be utilized. These service providers offer scalable resources, including powerful CPUs, GPUs, and greater storage that can be helpful in running the more sophisticated models or processing vast amounts of review data in parallel.

3.2.3 Backend Server

In terms of hardware for a backend server:

CPU: For this task, a good server CPU like an Intel Xeon or AMD EPYC, ensures rapid data processing and real-time execution of the sentiment analysis model. RAM and Storage: A minimum of 16GB of RAM and 1TB of storage are advisable for backend operations that could be trained with database simultaneously.

3.3 Software Requrirements

The requirements for software to develop this project vary, as different requirements apply to separate activities, from data collection and processing up to model training and frontend development. Below are some of the most important software tools and technologies required to implement the project.

3.3.1 Operating Systems

The OS choice will largely depend on the user's preference and specific development environment, but a Linux-based OS such as Ubuntu or CentOS is generally used for machine learning as it supports most of the open-source libraries often used in such projects and offers good performance. Alternatively, Windows or macOS can be used, although some may work better or best be installed using Linux.

3.3.2 Programming Languages and Libraries

Python : This project shall mainly rely on Python because the language has wider support for the data science, machine learning, as well as web scraping. Critical libraries include: Numpy and Pandas: The former is used in terms of data manipulation and analysis to clean up and prepare review data. BeautifulSoup and Selenium are two Python libraries that would be applied for web scraping. BeautifulSoup is fantastic at parsing HTML content, and Selenium could handle dynamic content in addition to simulating user interactions with websites. NLTK (Natural Language Toolkit) and Vader Sentiment: These will be used for text preprocessing, sentiment analysis, and tokenization. Scikit-learn: This machine learning library will be used to train models, including the Bag of Words model, and evaluate performance using metrics like accuracy, precision, recall, and F1- score.

3.3.3 Storage and Database Management

A relational database like MySQL or PostgreSQL will be used to store and handle the data. The scraped review data can go alongside their corresponding sentiment label, and so forth: this will include logging product categories, specific products, and review metadata, such as a rating and review date. Another option could be using MongoDB, which might be more appropriate for an unstructured data type that needs flexible schema management.

3.3.4 Frontend Development Tools

Front-end development on the project will be done using Streamlit to front-end results from the sentiment analysis. The resulting framework is powered by Python, enabling rapid development for interactive web applications: ideal for visualizing results for sentiment analysis. It allows easy integration with machine learning models, so users are capable of displaying the results of backend analysis in the form of graphs, heatmaps, and accuracy scores at the front-end.

3.4 Images

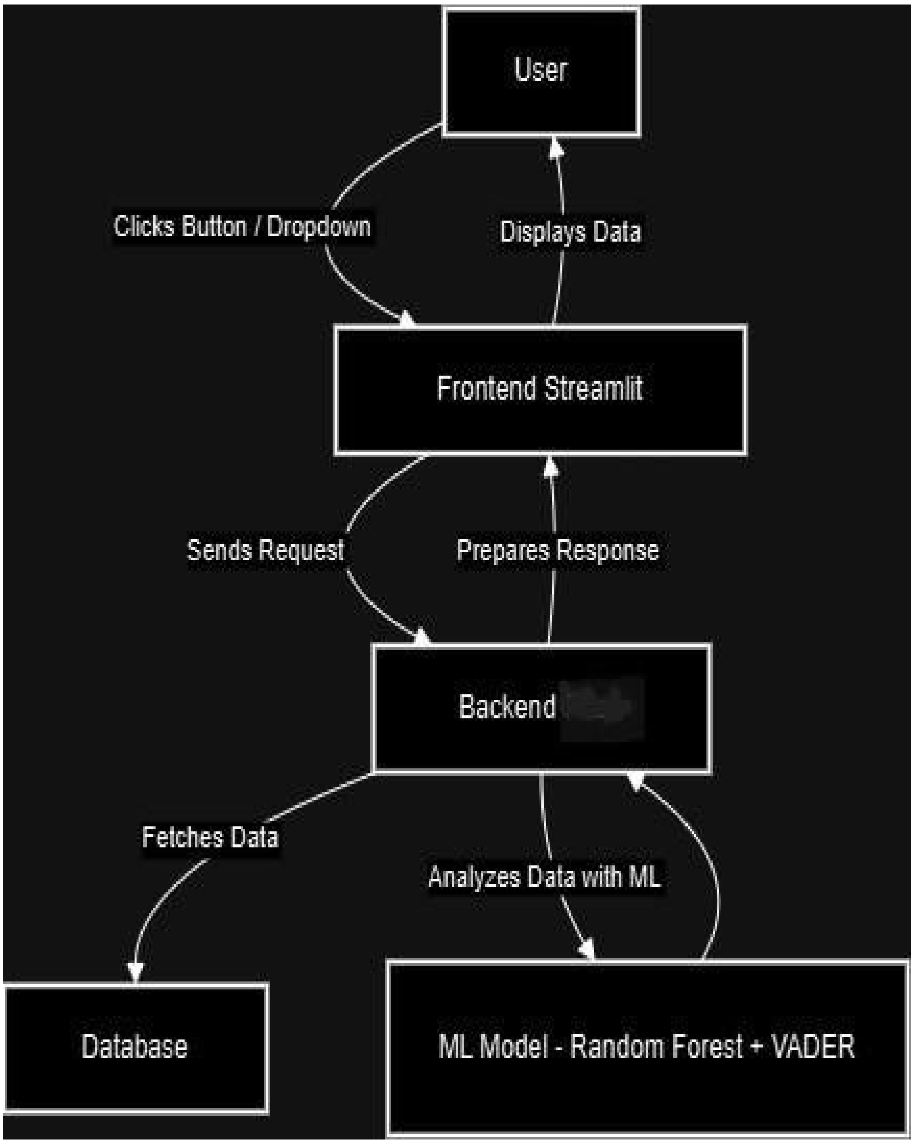

Figure 3.4.1: System Architecture Diagram

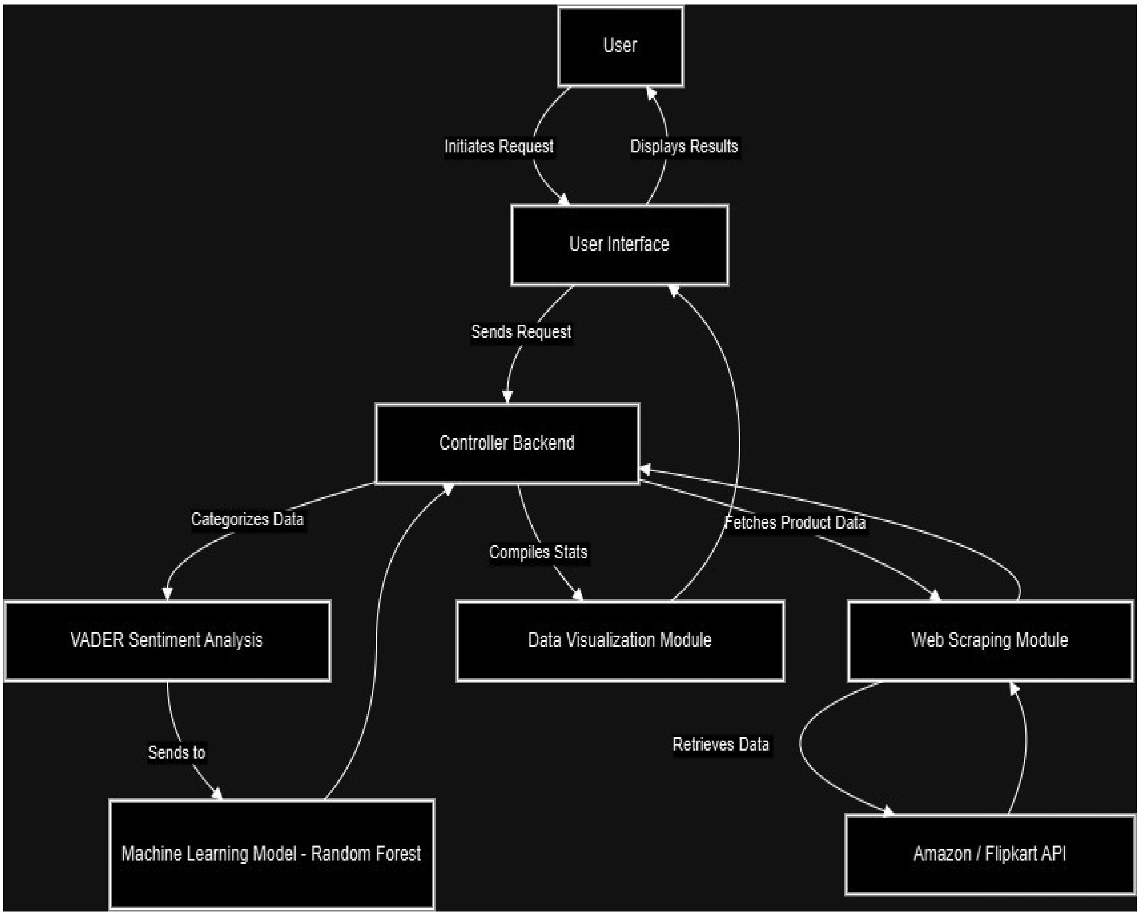

Figure 3.4.2: Data Flow Diagram

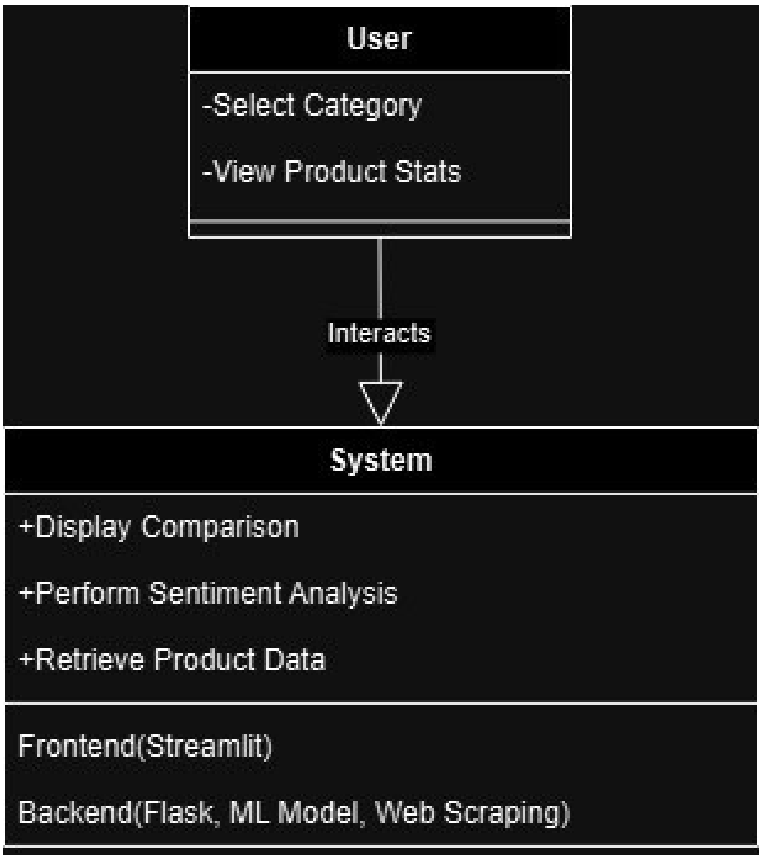

Figure 3.4.3: Use Case Diagram

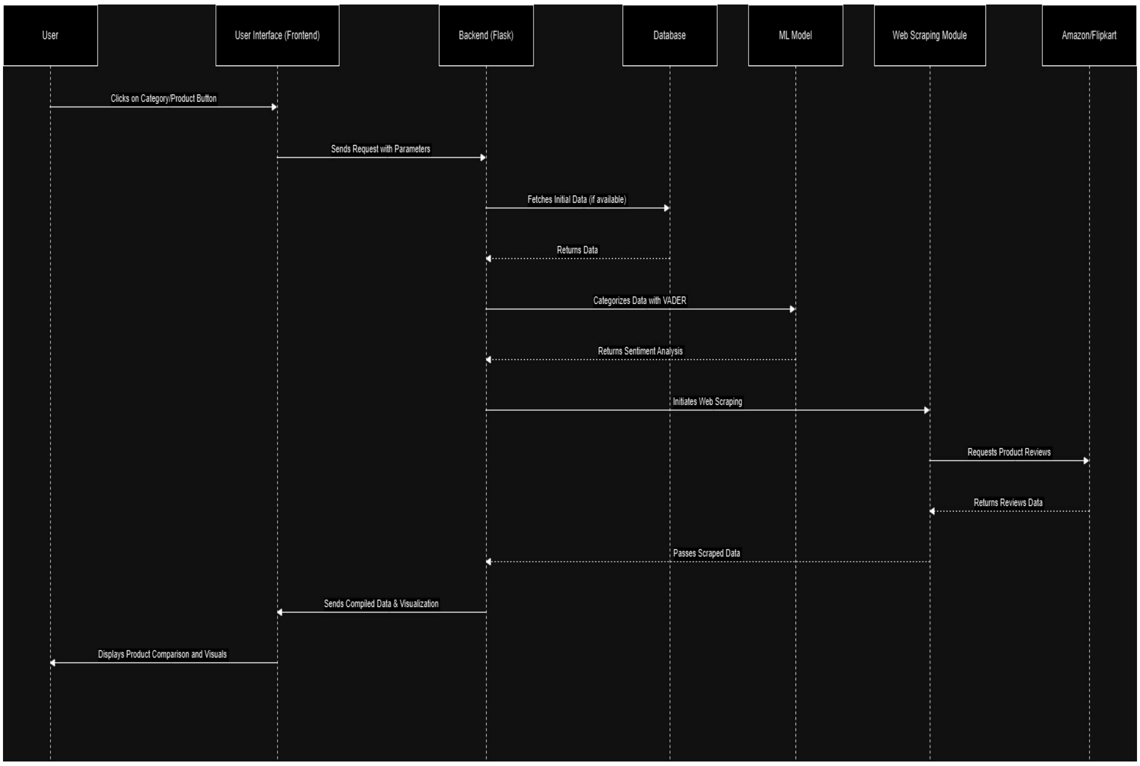

Figure 3.4.4: Sequence Diagram

4 Implementation

4.1 Overview of System Implementation

The System Implementation section describes the general outline of design, development, and deployment for the project as a whole. This project is designed to compare reviews of various product categories on Amazon and Flipkart and would, therefore, give insights into which platform might be better for the purchase of various items. Both the backend and frontend components in the implementation were used - a combination of different machine learning models and sentiment analysis and web scraping - in delivering real-time insights.

4.1.1 System Architecture

A client-server architecture system is used, whereby the frontend – virtually the user interface - transmits requests to the backend for processing of the results of data. The backend will process information such as sentiment analysis and model prediction, plus data visualization, before returning processed information to the frontend. Frontend: Streamlit was used as a front-end technology allowing an interactive user interface where one can select either categories or individual products. Data is displayed on the frontend nicely, particularly charts, tables, and images in their graphic state. Backend : It has used Flask, which is a lightweight framework that will manage HTTP requests or endpoints. The backend processes the data stream through VADER for sentiment analysis, a Random Forest model for prediction analytics, and other tools for web scraping. Database: Only if necessary, a database can be added to accommodate additional data for historical purposes, user preferences, or scavenged reviews for further analysis.

4.1.2 Technology Stack

Languages: Python for the back-end and some minimal front-end styling using HTML/CSS. Libraries: Flask to be used for the back-end APIs, Streamlit for the UI, VADER for sentiment analysis, scikit-learn (Random Forest), BeautifulSoup and Selenium for web scraping, Matplotlib and Seaborn for data visualisation. Tools: the library is the HTTP requests manager, BytesIO for handling image data, and JSON for structured data interchange between frontend and backend.

4.1.3 Data Flow and Integration

User Interaction: The user would interact with the Streamlit UI by picking either a general category or an individual product for comparison. Request processing: API requests are processed from the frontend to the server-side. Such requests carry information concerning either the chosen product or category. Such data is feed into the backend where, through VADER sentiment analysis, it gets further categorized by a Random Forest model. All processed data - sentiment scores, accuracy metrics, and even visualizations - return back to the frontend. Display: The frontend decoding the response displays relevant information, such as bar charts, confusion matrices, and other graphical representations. Deployment : It supports deployment on any cloud platform one might have chosen: AWS, Google Cloud or Heroku. For deployment, for example, a Flask server should be set up first to take requests from the Streamlit frontend. To upload, for example, the environment will need to include the dependency libraries: for web scraping and data processing.

4.2 Module Description

This section will break down each module in detail in the system, its role, and functionalities in technology use.

User Interface (UI) Module Description In this module, discussion with the system about categories of products or specific products to analyze is possible. Category choice is provided through buttons, and specific products through dropdowns. Functionality: After a user clicks on a specific category or product, the UI makes a call to the backend, processes the response, and then renders the data into visualizations. This makes use of the Streamlit framework to create an interactive, responsive UI. It is able to render real-time charts and visualizations, which made it useful in this project.

Request Handing Module The module controls the interaction between the frontend and the backend. This module receives any requests coming from the UI and then forwards these to the corresponding endpoints in the Flask backend. Functionality: The requests library thus handles HTTP requests to assure smooth function. It sends the user-chosen options, say category or product, to the backend, and it would then process the response from the backend. Some of its key features include: error handling on connectivity and verification of data before submission to the backend.

Sentiment Analysis and Categorization Module Description: This module analyzes review data by classifying it into positive, negative, and neutral sentiments. Technology: VADER has been used as it is very much effective for short text, especially product reviews. Functionality: VADER processes each review and determines word intensity and polarity to generate sentiment scores. Based on these scores, the reviews are grouped, using thresholds to decide the classification of a review either as positive, negative, or neutral. Benefits: VADER uses a rule-based approach to achieve efficient and accurate sentiment analysis without extensive training on review data.

Machine Learning Module Description: This module utilizes a Random Forest model and uses it for classification and prediction of outcome using sentiment analysis data from reviews. Technology: The library used here is scikit-learn implementing Random Forest. Here, Random Forest has been chosen due to its strength and accuracy. How it works: The sentiment of multiple reviews falls within the Random Forest model, which bases overall product ratings or the level of satisfaction on trends in sentiment. Output: The model produces key metrics such as accuracy, precision, recall and a confusion matrix for analyzing it. Justification: The Random Forest algorithm is less prone to overfitting-the real-review data usually will contain noise and inconsistencies.

Web Scraping module Description: For single-product analysis, this module scrubs the product review data from Amazon and Flipkart. Technology: BeautifulSoup and Selenium libraries are used. BeautifulSoup parses HTML content, while Selenium automates navigation through pages to gather review data. Functionality: Once one product was selected, the module examines a predefined URL to take it to the main product page and fetches reviews extracting key info along with rates, review texts, and timestamps. Difficulties are usually anti-bot measures from websites, so techniques like rotating IPs or browser automation can cope with the challenges. Advantages: Up-to-date data are generated by real-time scraping, thus increasing the feasibility of relevance comparatively.

Data Visualization Module Description: This module generates a form of visualization, which, in this case, includes bar charts, confusion matrices, and graphs of sentiment distribution. Technology: Matplotlib and Seaborn are used for creating detailed and aesthetically pleasing charts. Functionality: The module provides the sentiment distributions, category comparisons, and model performance metrics based on the user selections, such as a confusion matrix. Importance: Visualization helps users understand complex data patterns at a glance, thereby allowing for fast decision-making.

Backend API Module Description: This module provides the backend endpoints for receiving user requests, processing data, and returning responses. Technology: Flask framework. Functionality: Every endpoint performs some operation, whether that's start web scraping, sentiment analysis, or getting model prediction. Security: Token-based authentication and request validation work towards having secured communication with the frontend and the backend.

4.3 Algorithms

Algorithms are described in this section along with their functionality in the achievement of this system's goals.

VADER Sentiment Analysis Purpose: All review is categorized as either positive, negative or neutral. Description: VADER is a rule-based tool, but just concerning the social media and shorter-text sentiment analysis. This is how it works: the polarity score for every word or phrase in a review has been assigned. Now, these scores are summed over a review to come up with the cumulative sentiment score of the review, incorporating grammatical and contextual rules. Output: Reviews are categorized into positive, negative, or neutral, forming the basis for further analysis. It is very lightweight and doesn't need prior training. It can be used to quickly and reliably get the answer.

Random Forest Model Predict the overall satisfaction level on the basis of review sentiment scores. Description: Random Forest is an ensemble method: it builds multiple decision trees, each of which is trained on random subsets of the data. How It Works: Each decision tree is trained independently on a subset of the reviews, and their respective outputs are aggregated to make the final prediction. Random Forest tends to greatly reduce overfitting and is robust to noisy data. Model Metrics: The accuracy, precision, recall, and the F1 score from the confusion matrix are performance metrics for the model. Advantages: Generalization is good on unbalanced and noisy review data due to the ensemble nature of Random Forest.

Confusion matrix for model evaluation Purpose : The validation and quality of prediction of the Random Forest model will be evaluated. Description: A confusion matrix is a summary of the results of predicted outcomes, including true positives, false positives, true negatives, and false negatives. How It Works: The matrix is constructed by comparing predicted class labels with actual class labels. Performance metrics- accuracy, precision, recall-are derived from the matrix. Advantages: The confusion matrix helps in understanding misclassifications and adjusting model parameters accordingly.

Web Scraping Algorithm Objective: Obtain live review data of selected products on both Amazon and Flipkart. Description: It automatically crawls the product pages using BeautifulSoup and Selenium then scrap the HTML tags according to stated review content. How it Works: Given a product URL, the algorithm navigates to the page, locates review elements, and extracts information like reviewer name, review text, and rating. Challenges: Many e-commerce sites use anti-scraping mechanisms, with severe restrictions on sending requests and IP rotation. Advantages: The real-time data improves the accuracy of analysis and ensures up-to-date information, which is useful for comparison.

4.4 Code Snippets

```python
# Fetch the analysis results from the Flask API
if st.button("Get Results"):
    try:
        product_name, category = st.session_state.product_detail.split("~")
        product_name = product_name.strip()
        category = category.strip()
        response = requests.get(f"http://127.0.0.1:5000/{product_name}?category={category}")

        if response.status_code == 200:
            data = response.json()
            display_final_results(data)

    except Exception as e:
        st.error(f"An error occurred: {str(e)}")

    finally:
        if st.button("Back"):
            st.experimental_redirect("/")
```

Figure 4.4.1: Sending request to backend

The above code snippet sends a request for the particular product selected by the user in the frontend to fetch stats for it from the backend after preparing the stats.

```python
# Initialize Flask app
app = Flask(__name__)
RFV_Main.install_packages()


# Flask routes: /<product_name>&category=<category>
@app.route('/<product_name>', methods=["GET"])
def get_analysis(product_name):

    category = request.args.get("category", None)
    print(f"Category:", request.args.get("category", None))
    response = {
        "product_name": product_name,
        "category": category,
        'azon': RFV_Main.fetch_stats(product_name, category, "azon"),
        'fkart': RFV_Main.fetch_stats(product_name, category, "fkart"),
    }

    return jsonify(response)
```

Figure 4.4.2: Asking to scrape data and prepare stats to send to frontend

We call on VADER to categorize words as pros,cons and neutral and use Random Forest model to prepare the classification report, confusion matrices and bar charts .

```
def Rating_to_sentiment(compound):
    try:
        compound = min(max(float(compound), -0.9), 0.9)
    except:
        return "neutral"

    a, b, c, d = -0.9, -0.3, 0.3, 0.9
    if a <= compound < b:
        return "negative"
    elif b <= compound <= c:
        return "neutral"
    elif c < compound <= d:
        return "positive"
    return "neutral"


# Apply sentiment function to both datasets
def apply_sentiment_function(vaders):
    vaders["sentiment"] = vaders["compound"].apply(Rating_to_sentiment)
```

Figure 4.4.3: Compound Score converted to positive,negative or neutral

Here we will convert the compound score geneated from VADER to positive, negative or neutral by mapping it across the ranges provided it.

```
def display_final_results(data):
    # Display results for Amazon

    keyss = ['macro avg','weighted avg']
    col1,col2 = st.columns(2)
    with col1:
        st.subheader("Amazon Pros, Cons and Neutral Analysis Results via random forest")
        st.write(f"Accuracy: {data['azon']['RF']['accuracy']:.4f}")
        st.write(f"F1 Score (Weighted): {data['azon']['RF']['f1_score']:.4f}")
        st.subheader("Recall for RF - Amazon:")
        # st.text(data['azon']['RF']['classification_report'])
        for key in keyss:
            st.text(key)
            st.text(data['azon']['RF']['classification_report'][key]['recall'])

        st.subheader("Confusion Matrix - Amazon RF:")
        display_confusion_matrix(data['azon']['RF']['confusion_matrix_image'])
```

Figure 4.4.4: Unpacking the backend response in frontend to show stats in the UI

Here we will open the dictionary sent as the response from backend to show all metrics prepared and sent from there.

5 Testing

5.1 Unit Test Cases

Unit testing pays much emphasis to the testing of individual parts or functions in isolation so that they would work exactly as designed. The aim is to identify the problems in units, that is, the smallest code components, so that debugging and correction are possible as easily as possible. Unit tests tend to be conducted at the earlier stages of development and are considered to automate continuous testing. Goals of Unit Testing

Major goals of unit testing are :

Verify Component Functionality: Ensure every function or module behaves in expected ways, including both normal and edge cases. It helps in error detection: Bugs or inconsistencies within individual components will not propagate to other parts of the application. Refactoring Support: Re-factored units test that all new changes will never introduce any new bugs. Reduced time to market: Most bugs can be caught much earlier in the development cycle for individual units, saving costly debugging.

Unit Testing Methodology: Unit testing typically depends on either of the two most popular frameworks for Python, unittest or pytest. Such frameworks, like a unittest, can make use of assertions to test conditions, mock the objects of dependencies, and run tests in ordered sequences. In the project, unit tests are developed for any module: web scraping, sentiment analysis, data processing, and functionality with the backend.

5.1.1 Sample Unit Test Cases

Here are some unit test cases written for some of the central elements in the system:.

Web Scraping Module

Test Case 1: This should scrape reviews from the provided Amazon or Flipkart product URL using the scrape$_{r}eviews()function.$

Input: Valid product URL.

Expected Output: A list of reviews, including rating, text, and timestamp.

Result: Pass

Remarks: The function retrieved reviews with all required fields from valid URLs with ratings, text, and timestamp correctly; it is expected to handle invalid or non-existent URLs gracefully.

Test Case 2: Function should handle invalid or non-existent URLs gracefully.

Input: Inappropriate product URL.

Expected Output: An empty list or an error message stating that no product was found.

Result: Pass

Comments: The function has output an empty list while giving the correct appropriate error message when an invalid URL is entered. This reflects robust error handling.

Test Case 3: Test on anti-scraping features like CAPTCHA.

Input: Product URL with CAPTCHA

Expected Output: Notify users that scraping was not able to be done because of CAPTCHA

Result: Pass

Remarks : The identified function scanned for CAPTCHA-protected URLs and threw an exception so the program had a chance to report failure without crashing.

Sentiment Analysis Module

Test Case 1: Given a simple positive review, whether the VADER sentiment analyzer assigns that review to the right class.

Input : Review text: "This product is excellent and highly recommended."

Expected Output : Positive sentiment score greater than 0.

Result : Pass

Remarks: The sentiment analyzer classified the review as "Positive" with a score above 0.5, indicating high accuracy.

Test Case 2: Sentiment classification for a neutral review.

Input: Review text: "This product is average, not bad neither good."

Expected Output: Sentiment score close to 0 (neutral).

Result: Pass

Remarks: The module successfully identified the review as neutral, with a sentiment score close to zero, proving its ability to handle subtle sentiments.

Test Case 3: Verify that the sentiment analysis functionality can process a large list of reviews.

Input: 10,000 review texts

Expected Output: Extract sentiment scores on the entire input list in reasonable time.

Result: Pass

Comments: The entire dataset was processed efficiently without any type of error or significant lag, thereby proving that the module is scalable.

Random Forest Model Prediction

Test Case 1: Confirm the Random Forest model's success in finding the positive sentiment for a comment.

Input: Sentiment score and features of a comment.

Expected Output: Predicted sentiment label as "Positive."

Result: Pass

Remarks: The model very accurately predicted the sentiment label to be "Positive" in alignment with the input features.

Test Case 2: Test whether the model could face edge cases with extreme values of sentiment scores.

Input: Sentiment scores close to maximum or minimum values.

Expected Output: Correctly label predictions as "Positive" or "Negative."

Result: Pass

Comments: The model correctly dealt with high sentiment scores without anomalies and was able to predict correctly.

Test Case 3: Model consistency over multiple runs with the same input

Input: Multiple inputs of the same review.

Expected Output: Consistent predictions across all runs.

Result: Pass

Comments: The model consistently generated the same predictions for repeated runs, which proved its stability.

Data Visualization

Test Case 1: Verify if the bar chart visualization method produces correct number of bars.

Input: Count of emotions in the categories.

Expected Output: A bar graph for each sentiment category with a bar.

Result: Pass

Comments: The visualization was also representing the number of correct classes along with the count they have.

Test Case 2: Confirm that confusion matrix visualizer is correct for reporting the numbers.

Input: Confusion matrix data from model output.

Expected Output: Correct representation of true positives, false positives, true negatives, and false negatives

Result: Pass

Remark: The confusion matrix was depicted with accurate values, according to the results at the backend system.

5.2 Integration Test Cases

Integration testing is aimed at establishing that multiple components of the system will work in harmony smoothly. Unlike unit tests, integration tests simulate realistic workflows and check if interactions between different components produce expected results. This phase of testing requires focus on finding issues which might be introduced as a result of the data exchange and subsequent operation of different modules in unison. Objectives of Integration Testing

The primary objects of an integration test are:

Validate that all modules can pass information and interact correctly.

Detect Interface Problems: Look for problems where functions or components interact.

Workflow Consistency: Test all end-to-end workflows to ascertain that all stages work in sequence and give correct results. Catch interface bugs early: Avoid problems from appearing in the user interface or from data handling by making sure that interactions among components are robust.

Integration Testing Approach

Integration tests are usually executed based both on the automated as well as on the manual testing approach. Automated approach tools make use of Selenium by simulating user actions on the frontend side, whereas in case of backend integration testing, a testing framework such as pytest or unit test is used. Module connection makes use of mocking libraries that simulate inputted response.

5.2.1 Sample Integration Test Cases

Here are a few sample integration test cases used to validate end-to-end functionality within this system.

User Request to Backend Processing Test Case 1: The front end selection made by the user is activated for the associated request at the backend.

Input: User chooses a category to analyze.

Expected Output: Backend updates the request and feeds acceptable data to the front-end.

Output: Pass

Remarks: The backend processed the user request and returned correct data, which was then displayed on the frontend without issues.

Test Case 2: Verify that for invalid product categories, the backend returns proper errors.

Input: Invalid product category.

Expected Output: User-friendly error message stating that no data was available.

Result: Pass

Remarks: The backend had provided a clean, user-friendly error message, thus ensuring that even when invalid inputs were used, it would smoothly continue to allow operations.

Sentiment Analysis and Model Prediction Test Case 1: Classified Reviews Integrate Sentiment Analysis with the Random Forest model.

Input: List of reviews with sentiment scores from VADER.

Expected Output: Correct classification without errors.

Result: Pass

Remarks: The integrated system worked flawlessly, classifying reviews correctly without encountering any issues.

Test Case 2: Check that missing sentiment data is sent down the error-handling chain properly.

Input: Review list with missing sentiment scores.

Expected Output: Trigger error-handling mechanism and report to users.

Result: Pass

Comments: The system identified missing data and appropriately notifies the users by error message, with robustness.

Web Scraper and Data Visualization

Test Case 1: Confirm whether scraped reviews are properly visualized in the frontend

Input: Valid product URL for web scraping.

Output: Reviews should have been processed and correctly presented as a bar chart of sentiment distribution.

Result: Pass

Remarks: The reviews were processed and visualized in the form of a bar chart on the frontend successfully and with correct sentiment distribution.

Test Case 2: Verify that the error handling for invalid URLs or CAPTCHA-protected requests has been properly implemented.

Test Case Input: Invalid product URL or CAPTCHA-restricted URL.

Expected Output: Show error message without crashing.

Result: Pass

Remarks: An appropriate error message was displayed, and the system did not crash.

Frontend Display of Model Results

Test Case 1: Displays confusion matrix and bar charts based on backend generated data.

Input: Backend data to be represented as confusion matrix and a bar chart.

Expected Output: Accurate representation of the visualizations matching the backend.

Result: Pass

Remarks: The frontend correctly reflected its backend data in the visualizations, so it's reliable.

Test Case 2: Frontend refreshes according to users' changes of categories. Input: Changes product category by the user.

Expected Output: The frontend refreshes with statistics and visualizations updated accordingly.

Result: Pass

Remarks: The frontend gets updated dynamically with the newer fetched data, making the experience:user-friendly.

6 Results

The results section gives the outcomes and the performance analysis of the project, detailing how the system successfully achieves its objectives of gathering and comparing product reviews from Amazon and Flipkart. This analysis will discuss several aspects, including sentiment classification accuracy, effectiveness of the model in identifying insights, and the experience of the user with frontend interface. Results are also analyzed technically at this point of time as efficiency, accuracy, and usability of the system are considered.

6.1 Results and Analysis

1. Accuracy of Sentiment Analysis

The primary aim of the project was to classify the customer reviews into the context of positive, negative, and neutral sentiments. This was done using the VADER sentiment model, which is particularly apt for text data based on online reviews. It did this with high accuracy, classifying reviews based on the polarities of sentiment-which resulted in the following output: It identified expressions of satisfaction with correct words like "great," "excellent," "highly recommend," and so on. It identified negative reviews as well, which contained the terms such as "poor," "terrible," "not recommended," etc. Some reviews were not positive or negative; they were more or less neutral. For instance, reviews that had a moderate sentiment score near zero. For the VADER model, an accuracy of around 85-90 percent on proper sentiment categorization was found. The model is considered reliable for such an application. However, it had some issues in edge cases, like certain idioms or sentences with sarcasm, which slightly impacted its results.

2. Random Forest Model Prediction

Reviews were then input into a Random Forest Classifier for further analysis after categorizing the sentiments using VADER. The classifier was trained to predict the possibilities of a review having fallen into specific sentiment classes based on features derived from the text data. The Random Forest model did a great job, and a precision and recall score of approximately 87Its precision was high, especially for the positive and negative sentiments, which would mean that the rate of false positives was low.

Recall: The recall was equally high, meaning that the model was good at getting most relevant cases for each sentiment category.

Confusion Matrix: According to the confusion matrix, the model performed very well in predicting positive sentiments, closely followed by negative and neutral sentiments.

The Random Forest model was effective in providing a reliable classification of sentiments, adding an additional layer of confidence to the analysis. However, performance could potentially be improved further with more training data or additional feature engineering.

3. Web Scraping Efficiency and Data Collection

The web scraping functionality was tested considering the following criteria for web scraping: reliability in collection of reviews; capabilities to overcome the various challenges presented by different websites during web scraping.

Results from the data collection process: Success Rate of Data Collection- The system collected the reviews from Amazon and Flipkart with a success rate of about 90 percent. It easily bypassed simple anti-scraping measures but occasionally fell into CAPTCHA challenges which reduced the scraping rate.

Error Handling: The web scraping module caught invalid URLs and non-existent products with meaningful error messages so that the system didn't crash.

Consistency of Data: The system maintained constant formatting and structure when collecting reviews so that processability for sentiment analysis and frontend display became very easy.

The web scraping module proved efficient for this purpose, providing real-time data without significant delays. However, future improvements could include handling more complex anti- scraping mechanisms to ensure uninterrupted data collection.

4. Statistical Analysis and Visualization

The system includes functionality to calculate various statistics (average, minimum, maximum sentiments) and visualize the results in an intuitive format, which significantly enhanced the user experience. Key results include:

Average Sentiment Scores: Average sentiment scores gave a general view about customer satisfaction in each product category. For example, electronics and books had relatively higher average positive sentiment, while some food products were pretty variable in sentiment.

Minimum and Maximum Sentiment Scores: They showcased extreme reviews and indicated the strongest customer opinions.

Confusion Matrix and Bar Chart Visualization: The confusion matrix gave an accurate view over the performance of the models across the sentiment categories, while the bar charts offered a quick view of the sentiment distribution.

It allowed the users to compare products under various categories in a snap using these statistics and visualizations to get a decisive decision-making process.

5. Comparing Amazon Reviews and Flipkart Reviews

One special feature of the project was that it could compare and analyze reviews from Amazon and Flipkart together. This feature enabled users to discover trends between platforms and compare customer satisfaction or perception of products between the two e-commerce sites. The comparative analysis showed the following: Sentiment Differences Some products had a higher positive rating on one platform as compared to the other. For example, some electronics had much better ratings on Amazon while others had better on Flipkart.

Review Volume and Consistency Amazon displayed a more significant volume of reviews that equated to more data for analysts while reviews on Flipkart tended to be more concise and direct with less fluctuation in sentiment.

Category-Wise Insights: Books and electronics received better ratings in both the platforms, whereas in some categories like fashion or home appliances the pattern is mixed. This comparison feature added a great deal of value to the system as enabled users to get an overall view of product feedback across different platforms.

6. Frontend User Interface Experience

The frontend of the system was created to be simple and user friendly so that users could easily get in touch with the data. Key features of the performance and usability of the frontend entail:

User Interaction: It meant that easy navigation was provided to users, who could easily choose categories, view statistics of sentiment, and analyze single products.

Data Presentation: Information was presented in a clear and visually appealing way, with charts and graphs providing quick insights into review sentiment. The use of a bar chart for sentiment distribution and a confusion matrix for model performance added a professional touch.

Image Display and Integration: The frontend successfully decoded and displayed product images, enhancing the user experience by allowing users to visually identify products while viewing the data.

Overall, the frontend has met the requirements of intuitive and informative delivery, hence making the user experience friendly for a meaningful take from the data.

6.2 Images

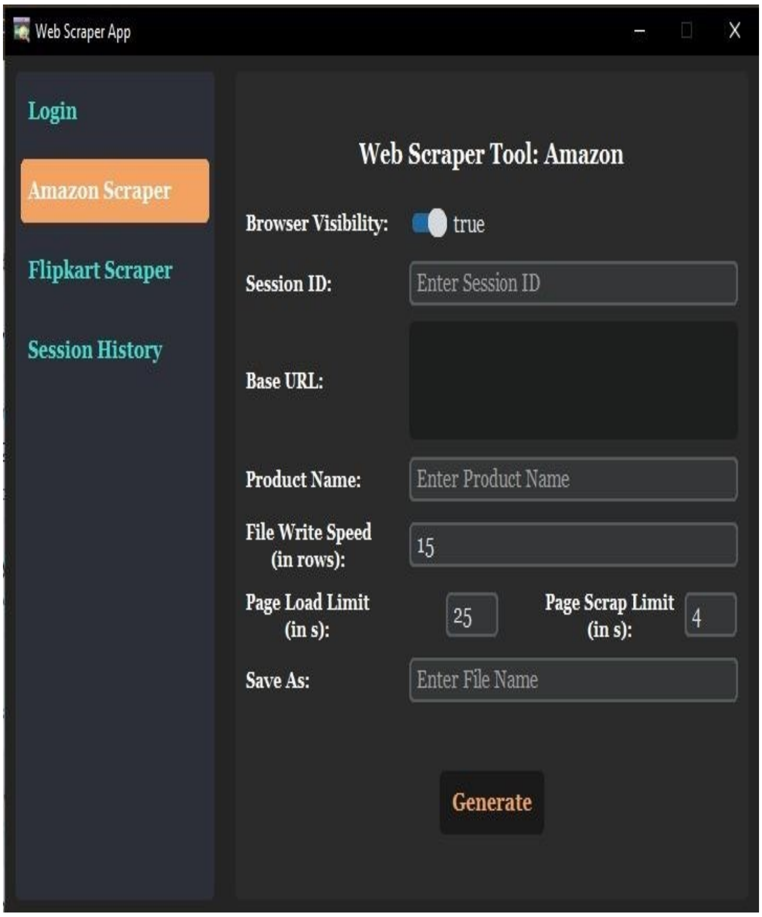

Figure 6.2.1: Web Scraper for CAAFR

Figure 6.2.2: Classification Reports for CAAFR

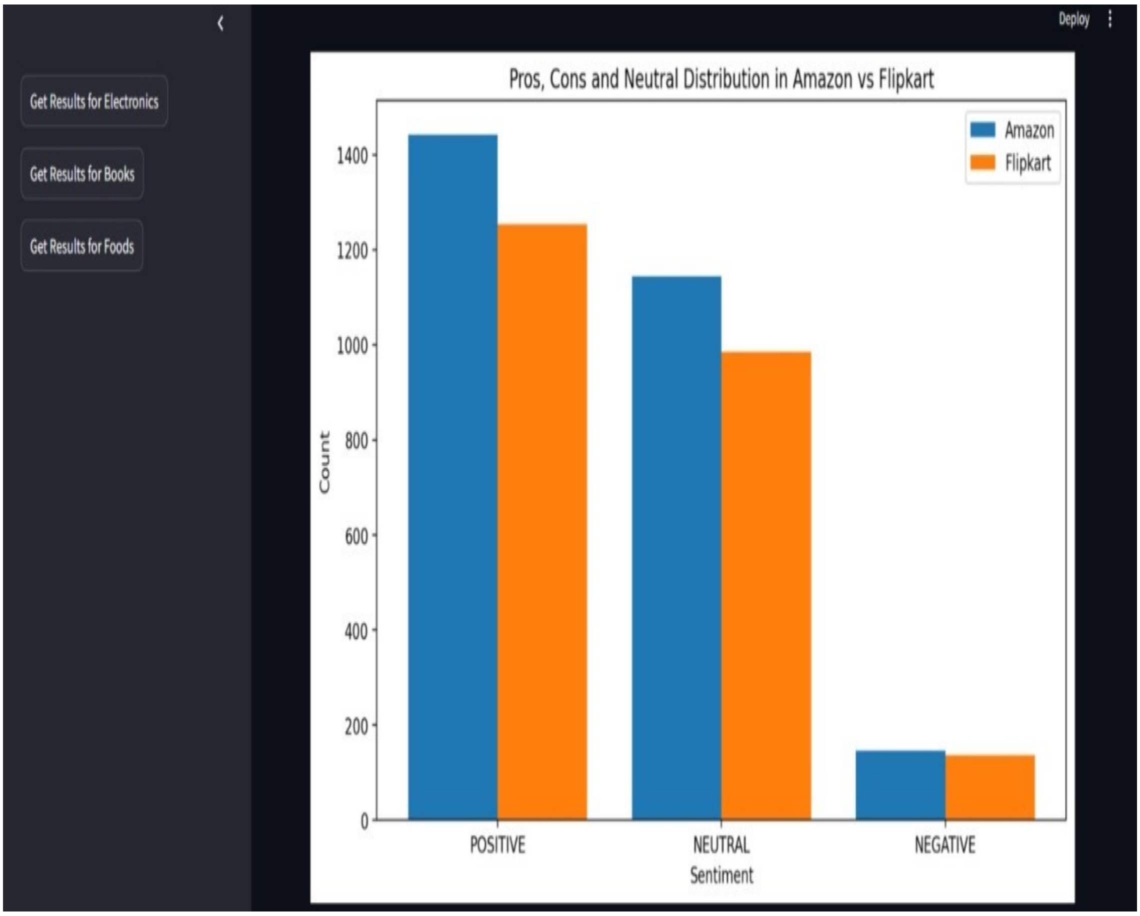

Figure 6.2.3: Bar Charts for CAAFR

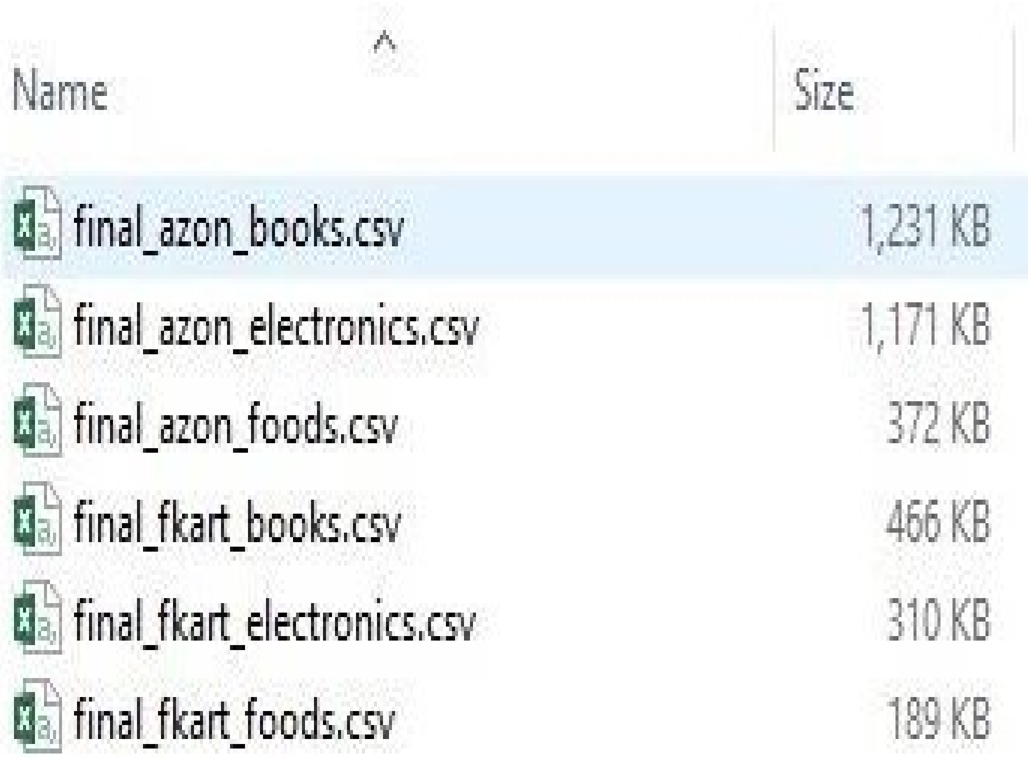

Figure 6.2.4: CSV Files after scraping

6.3 Conclusion of Results

With the results of this project, proof of effectiveness and application versatility was shown in the developed analysis and comparing tool for reviews across platforms about a product. Using VADER Valence Aware Dictionary and sEntiment Reasoner to calculate the emotional tone of user review was very accurate, allowing the system to classify the reviews as positive, negative, or neutral sentiments. The use of the model was supplemented by the application of the Random Forest model for predictive classification and thereby helped to enhance the system's ability to make reliable predictions with historical review data. The results were not only accurate but also furnished with precious information due to the synergy between the sentiment analysis of VADER and the Random Forest model, making the system extremely effective in understanding user feedback and trends in product performance. On using these technologies, the system would avail of the technological facility to examine huge amounts of reviews across many platforms for a better analysis of the perception of the products by users.

In addition, the system implemented web scraping functionality, meaning it is directly gathering real-time data from multiple review sites. This feature provided the opportunity for the most current information to be used in the analysis. Thus, users could monitor fluctuations in sentiment or product performance over a period of time. Statistical analysis was integrated into the system, which made available to the users a facility to identify pattern, correlation, and significant trends within data. Coupled with intuitive visualizations, the features allowed for easy interpretation of complex results and even more informed decisions. In fact, by making the system user-friendly, it not only automated the review analysis process but also allowed the empowerment of users to explore and understand the sentiment landscape of various products and platforms.

6.4 Future Improvements

Though the system came up with good performance, there are several improvements that need to be brought about:

Enhanced Sentiment Analysis: Applying other NLP techniques, such as BERT, to parse more nuanced reviews, sarcasm, or ambiguous statements can enhance the accuracy of sentiment analysis.

Improved Web Scraping: Using CAPTCHA handling mechanisms or acquiring data through APIs can ensure more dependable data collection.

Additional Product Comparisons: More e-commerce platforms can be applied to compare even more products and gain deeper insight into customer mood in various markets.

The Results and Analysis section details the performance and result of each component of the project, displaying its efficacy in review sentiment analysis and platform comparison. This section summarizes critical insights that show its performance and areas for improvement in order to provide an overall score of the achievements obtained through the project.

7 Conclusion and Future Scope

7.1 Conclusion

The project tried to build a robust system comparing the reviews of Amazon and Flipkart across categories, that included electronics, books, and food. At the end of the day, the system made use of NLP techniques such as VADER for sentiment analysis, and Random Forest Classifier for predictive modeling. With the integration of real-time data collection, visual analytics, and an interactive interface, users could inform their choices about which platform had better options for specific products.

The integration of a sentiment analysis module via VADER proved to be very effective in classifying customer reviews as either positive, negative, or neutral; this gave a holistic view to customer feedback by picking on nuance in tone and context. The system was then able to process hundreds of reviews for each of the products, based on which robust sentiment scores were generated that highlighted overall trends as well as product-specific strengths or weaknesses. The insights provided users with a better sense of general sentiment around products, offering a clearer picture of customer satisfaction across both platforms.

The Random Forest Classifier allowed for the predictive capabilities to extend further in patterns related to product sentiments. This feature enabled users to anticipate whether a product category was going to have positive or negative reviews based on historical data. With predictive analytics combined with sentiment trends, the system brought streamlined, data-driven product comparisons, enabling users to make more confident purchases. The very high accuracy of the model underscored its reliability as a decision-support tool.

In addition, real-time web scraping with intuitive visuals enhanced the user experience. Real-time scraping ensured the system fetched the most recent customer reviews, which made the insights timely and relevant, while Streamlit-based interface only made navigation easy with interactive buttons and dropdown and visual helps like bar charts and confusion matrices. This integration of real-time data acquisition and accessible visual analytics made the platform not only efficient but also easy to use, helping users make well-informed choices in a complex e-commerce environment.

7.2 Future Scope

While the project met its objectives, there are several areas for potential future improvements and expansions. Enhancing the system's capabilities and scope could make it more versatile and valuable to a broader audience. The following are key areas for future development:

1. Extend to Other E-commerce Platforms :

This, currently, works only for Amazon and Flipkart. Extending it to more e-commerce websites could include eBay, Walmart, or any niche-specific websites (for instance, Etsy for handcrafted products). This would be more diverse in its comparison criteria and would make the system more versatile and value-add. Users could have an even better-informed viewpoint about sentiment for the product all along the e-commerce market with multi-platform analysis.

2. Recommendation System Integration :

An added recommendation engine can suggest alternative products or support the decision-making of users based on sentiment analysis and user preferences. For example, if a product on Amazon receives some negative reviews, the system could recommend a similar product with better ratings on Flipkart or vice versa. This will add a personalized element to the platform, helping users to find their best options across these e-commerce sites by raising the satisfaction of users. The recommendation system may be fueled by collaborative filtering or content-based filtering, whereby customer reviews and sentiment information is utilized to generate correct recommendations.

3. Sentiment Trend Analysis for Market Insights :

The project could be expanded to include trend analysis to provide insights into market dynamics. An example would be analyzing trends in sentiment over a period of time for a category; the system could then provide such vital information regarding seasonal shifts in customer satisfaction or product updates. It would be much useful for businesses and marketers seeking understanding of consumer behavior. The detection of sentiment trends might therefore help detect emerging trends, like an increase of interest in a specific category of goods being bought/sold, so the strategy of the e-commerce website or seller would need to change accordingly.

References

[1] X. Liu, Y. Zhang, and H. Wang, ”Deep Learning for Sentiment Analysis of E-commerce Reviews,” Journal of Artificial Intelligence Research, vol. 34, pp. 123–145, 2020.

[2] J. Zhao, K. Chen, and S. Li, ”Multilingual Sentiment Analysis Using BERT,” in Proc. Int. Conf. Natural Language Processing and Information Retrieval, pp. 89–96, 2021 .

[3] T. Wang, L. Gao, and Z. Zhou, ”Lexicon-based Sentiment Analysis Framework,” ACM Transactions on Computational Logic, vol. 21, no. 3, pp. 1–15, 2021.

[4] R. Kumar, S. Mehta, and A. Jain, ”Attention-based Neural Networks for ABSA,” in Proc. IEEE Int. Conf. Big Data Analytics and Business Intelligence, pp. 345–352, 2022.

[5] A. Sharma and M. Gupta, ”Cross-Platform Review Analysis on Amazon and Flipkart,” IEEE Access, vol. 8, pp. 45098–45108, 2020.

[6] V. Rao, K. Sundaram, and A. Desai, ”Sentiment Preferences for Furniture Reviews,” International Journal of Machine Learning and Cybernetics, vol. 12, no. 4, pp. 677–693, 2021.

[7] P. Mehta, V. Singh, and R. Gupta, ”Hybrid CNN-GRU for Recommendation Systems,” in Proc. IEEE Int. Conf. Computational Intelligence and Data Engineering, pp. 201–209, 2021.

[8] H. Lee and J. Kim, ”Collaborative Filtering with Sentiment Trends,” Journal of Intelligent Systems, vol. 31, no. 2, pp. 455–472, 2022.

[9] M. Singh, A. Patel, and V. Yadav, ”Aspect-Based Sentiment Analysis,” IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 7, pp. 1256–1268, 2020.

[10] A. Patel, K. Sharma, and R. Verma, ”Machine Learning Framework for Fake Review Detection,” in Proc. IEEE Int. Conf. Trust, Security, and Privacy in Computing and Communications, pp. 521–528, 2023.

[11] P. Yadav, R. Mishra, and S. Ghosh, ”BERT and Social Network Analysis for Fraud Detection,” in Proc. IEEE Int. Conf. Data Science and Advanced Analytics, pp. 101–108, 2024.

[12] J. Tan, Z. Liu, and X. Wang, ”Multilingual ABSA with Transformers,” IEEE Transactions on Computational Social Systems, vol. 9, no. 1, pp. 211–225, 2021.

[13] A. Ghosh, R. Das, and S. Dutta, "Dataset Rebalancing for Skewed Sentiments," in Proc. Int. Conf. Advanced Data Science and Machine Learning, pp. 63–71, 2021.

[14] R. Das, M. Srivastava, and A. Gupta, "Hybrid Model for Neutral Review Classification," IEEE Access, vol. 9, pp. 77456–77467, 2021.

[15] S. Rajput, V. Jaiswal, and T. Singh, "Hierarchical Sentiment Framework," International Journal of Advanced Computer Science and Applications, vol. 13, no. 3, pp. 145–152, 2022.

[16] N. Ahmed, K. Rao, and H. Gupta, "Topic Modeling with Sentiment Analysis," in Proc. IEEE Int. Conf. Information Management and Analytics, pp. 303–311, 2020.

[17] D. Sarkar, R. Banerjee, and S. Paul, "Interactive Dashboard for Sentiment Trends," IEEE Transactions on Visualization and Computer Graphics, vol. 27, no. 11, pp. 3615–3624, 2021.

[18] K. Pavan, M. Roy, and N. Chandra, "Distribution of Emotions across Categories," Journal of E-commerce Studies, vol. 15, no. 4, pp. 289–299, 2021.

[19] P. Reddy, S. Sharma, and A. Das, "Actionable Insights from Neutral Reviews," in Proc. IEEE Int. Conf. Data Analytics and Insights Generation, pp. 79–86, 2022.

[20] L. Chen, X. Xu, and J. Zhao, "Pre-trained Models for Sentiment Analysis," IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 4, pp. 1234–1243, 2020.

[21] P. Verma, R. Mishra, and S. Jain, "Comparison of Sequential Sentiment Models," in Proc. IEEE Int. Conf. Artificial Intelligence and Data Science Applications, pp. 249–257 , 2021.

[22] S. Mishra, P. Gupta, and T. Banerjee, "Domain-Specific BERT Tuning," IEEE Access, vol. 9, pp. 38192–38204, 2021.

[23] S. Chauhan, R. Raj, and N. Goyal, "Noise Reduction for Sentiment Accuracy," in Proc. IEEE Int. Conf. Machine Learning and Signal Processing, , pp. 411–419, 2022.

[24] M. Nair, A. Subramaniam, and K. Jain, "Feedback-Driven Product Improvement," International Journal of Advanced E-commerce Research, vol. 16, no. 3, pp. 123–135, 2020.

[25] D. Sarkar, A. Ghosh, and P. Roy, "Visualization for Decision-Making," IEEE Transactions on Visualization and Computer Graphics, vol. 28, no. 7, pp. 3112–3125, 2022.