Fake Review Detection Using Machine Learning and Deep Learning, Distilled-Bert

Mayur Kadam, Shubham Marewad, Chetan Nemade, Parikshit Mote
Department Computer engineering, ZCOER, Pune, Maharashtra, India

ABSTRACT

In today's digital economy, the reliability of internet reviews is crucial in influencing consumer behavior. However, spam reviews—deceptive or manipulative entries that blur real customer feedback—have become a problem due to the extensive use of user-generated content across social media and e-commerce platforms. Although the nature and implications of such reviews have been examined in previous studies, more sophisticated detection methods are required due to the growing sophistication of spamming techniques. Using natural language processing (NLP) models that can recognize contextual inconsistencies and hidden patterns in review content, this paper explores the use of contemporary machine learning and deep learning approaches for spam review detection[9]. To improve model accuracy, behavioral characteristics like reviewer credibility, posting patterns, and temporal activity are also looked at.The goal of this research is to create a strong, scalable solution to stop spam reviews and preserve the integrity of online feedback systems by combining textual and behavioral insights. The study also underscores how social media is increasingly influencing consumer choices and how sophisticated sentiment analysis tools can offer more profound understandings of consumer intent, ultimately promoting equitable and reliable online marketplaces[5].

***Keywords: Fake Reviews**, Nueral Network, machine learning, deep learning, Behaviour Analysis, Text Analysis, Natural Language Processing, Genuine Reviews, E-Commerce Multi Model System*

1. INTRODUCTION

The change of Web 2.0 transformed the digital ecosystem completely by allowing people to share their thoughts, opinions, and experiences across multiple websites. Some of its most prominent effects include the emergence of online review sites where consumers can voice their opinions regarding products and services. These reviews, showcasing genuine opinions and influencing purchasing decisions, have become essential in a consumer's buying process. Spam reviews pose a major challenge that has arisen as a result of this openness. In the realm of online trust and reputation management, detecting and mitigating these deceptive or fake reviews is paramount, as they mislead consumers, distort public perception, and damage brand image.

image processing and machine learning techniques, including the integration of the Distilled-Bert model, to improve both detection accuracy and system robustness[4].Drawing upon prior work on automatically detecting spam reviews, this paper seeks to explore the further intricacies involved in employing more advanced methods to detect such content. Basic detection methods that use keyword spotting or rule-based algorithms attempt to cleanse content of minimally inauthentic elements; however, they often bypass sophisticated psychologically manipulative spam detections that are contextually rich. For this reason, this study focuses on better detection systems that integrate NLP with machine and deep learning for enhanced precision and broader applicability[9].

1. FAKE REVIEW DETECTION USING DISTILLED-BERT : ACCURACY APPROACH

The reviews provide the necessary feedback to both customers and businesses, and in recent times, there has been an increased dependence on reviews which has also led to the availability of numerous spam reviews. These fraudulent reviews are created with the intent to deceive potential customers or misguide competitors by deceptively inflating ratings. Trust is undermined with the existence of these reviews, business reputations are destroyed, and transparency in the market is disoriented. This manipulative behavior raises the issue of detecting fake reviews in e-commerce, data mining, or AI[3].

Collecting and preprocessing textual review data is one of the most important steps of solving the problem. Data in reviews is most of the time unstructured which means that it needs to be thoroughly cleaned and formatted before analysis. Standardized data is achieved through the use of several techniques like lowercasing, removal of stopwords, tokenization, and stemming for modeling. Furthermore, textual information is transformed into numerically meaningful data using natural language processing (NLP) techniques such as lemmatization along with vectorization, making them easier to analyze through machine learning algorithms[9].Using lexical and behavioral characteristics, traditional machine learning models such as Random Forest, SVM, and Naive Bayes have been used to identify fraudulent reviews. These models, however, frequently have trouble comprehending language's more complex semantics and context[1]. Transformer-based models, like BERT (Bidirectional Encoder Representations from Transformers), have been developed to get around this restriction and provide better performance in comprehending the context of words within sentences. We use DistilBERT, a lighter and faster variant of BERT that is optimized for speed without appreciably sacrificing accuracy, in this work. DistilBERT is perfect for large-scale review analysis because it is 60% faster and 40% smaller while maintaining 97% of BERT's language understanding.We examine the accuracy approach, i.e., how accurately DistilBERT can discriminate between authentic and fraudulent reviews. Our goal is to improve detection accuracy and lower false positives by utilizing the model's contextual awareness and fine-tuning capabilities. Standard metrics like accuracy, precision, recall, and F1-score are used to assess the model's performance after it has been trained on labeled datasets that contain both genuine and fraudulent reviews.We got the accuracy of 89.7% along with precision of 89.8.

2. ADVANCEMENT IN FAKE REVIEW DETECTION

Traditional machine learning techniques are frequently insufficient to accurately identify fake reviews due to the growing sophistication of deceptive content. In order to improve detection capabilities, recent developments in fake review detection have concentrated on utilizing deep learning, transformer-based models, and hybrid systems that integrate contextual, behavioral, and linguistic information[4].Natural language processing has undergone a revolution since transformers were introduced. Among these, the lightweight version DistilBERT and BERT (Bidirectional Encoder Representations from Transformers) have demonstrated impressive results in detecting fake reviews.In order to more accurately interpret word meanings, BERT processes language in both directions, comprehending each word's left and right context.

DistilBERT, which was employed in this study, provides almost the same performance as BERT with less memory and computation time, making it appropriate for real-time applications without significantly compromising accuracy[5].

These models' generalization and robustness across diverse review types and platforms are enhanced by their adaptation to multiple domains and fine-tuning on labeled datasets.

3. PROJECT ARCHITECCTURE

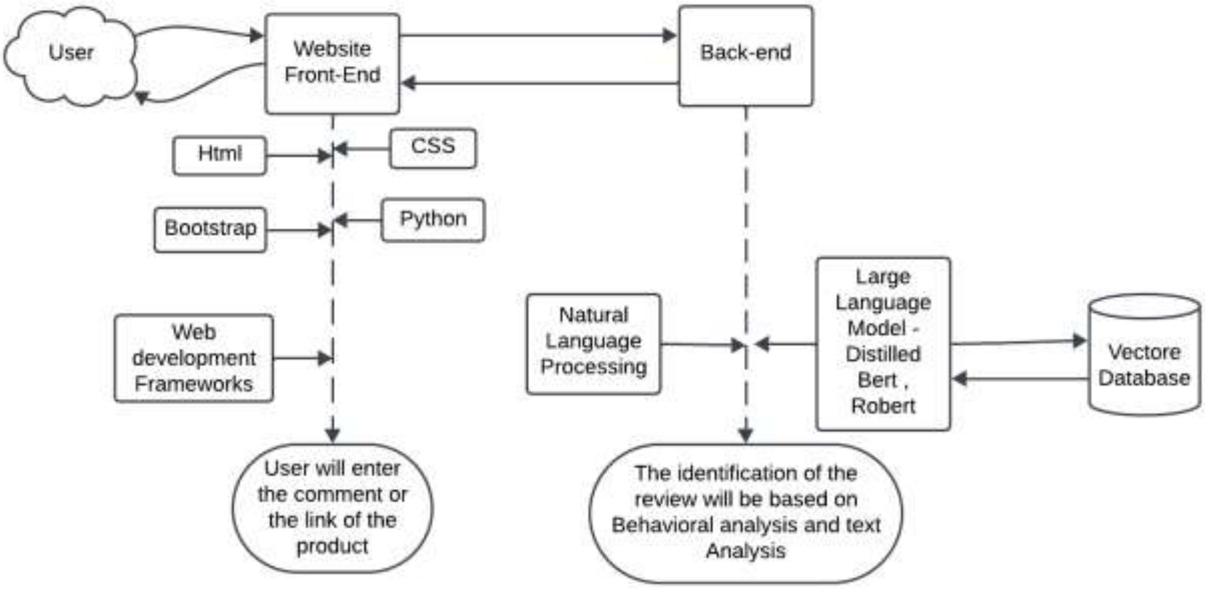

User
Website Front-End
Back-end
Html
CSS
Bootstrap
Python
Web development Frameworks
Natural Language Processing
Large Language Model - Distilled Bert , Robert
Vectore Database
User will enter the comment or the link of the product
The identification of the review will be based on Behavioral analysis and text Analysis

Fig1:Project Architecture of Fake Review Detction

1. User Interaction

The process begins with the user, who serves as the central participant in the system. The user interacts with the website's front-end through a graphical interface where they can either input a comment about a product or paste a link related to it. This input becomes the core data for further analysis. The user may write a review based on their experience or refer to content available online. The system is designed to accommodate various types of user-generated content. This ensures flexibility in how reviews are submitted and analyzed.

2. Website Front-End

The front-end of the website is what the user sees and interacts with. It is built using a combination of technologies like HTML for content structure, CSS for visual styling, and frameworks like Bootstrap for responsive design. Python can also be integrated for dynamic features and processing logic. The front-end acts as the entry point for user input and facilitates a smooth user experience. It is essential that this component is intuitive and accessible on multiple devices. The main goal is to collect user data seamlessly and pass it to the back-end.

3. Web Development Frameworks

Web development frameworks play a vital role in building scalable and maintainable websites. Technologies such as HTML, CSS, Bootstrap, and Python are commonly used in combination to enhance performance and speed up development. These frameworks offer reusable components, templates, and tools that reduce coding effort. For example, Bootstrap provides pre-built UI elements for better design consistency. Python can be used with frameworks like Flask or Django to handle logic. Altogether, these tools help developers create robust user interfaces efficiently.

4. User Input: Comment or Product Link

Once the user is on the website, they are prompted to either write a comment or paste a product link. This input is crucial as it represents the review or sentiment that will be analyzed. If a link is provided, the system can fetch the associated review text automatically. This flexibility allows the platform to handle both direct and indirect inputs. The collected data is then transferred to the back-end for processing. Ensuring correct and clean data input is the first step toward accurate analysis.

5. Back-End

The back-end is the core processing engine that manages the logic and communication between the user interface and the analytical tools. Once the input is received, the back-end initiates several processing tasks including text parsing, NLP, and interaction with machine learning models. It ensures the smooth functioning of the entire pipeline and handles data storage and retrieval. The back-end may be built using Python frameworks like Django or Flask for efficient routing and processing. This layer ensures all functionalities are performed behind the scenes without burdening the user interface.

6. Natural Language Processing (NLP)

Natural Language Processing (NLP) enables machines to understand, interpret, and derive meaning from human language. The user’s comment is analyzed using NLP techniques such as tokenization, sentiment analysis, and entity recognition. These steps help in identifying the tone, intent, and context of the review. NLP is essential for converting unstructured user input into structured data that can be processed by machine learning models. It bridges the gap between human communication and machine understanding. The more accurate the NLP, the better the quality of analysis[7].

7. Review Identification Based on Behavioral and Text Analysis

This step focuses on identifying the nature and authenticity of the review by analyzing behavioral patterns and textual content. Behavioral analysis may consider writing style, frequency of specific terms, and sentiment flow. Text analysis looks into grammar, keywords, and contextual meaning to classify the review. This combined approach helps in identifying fake, biased, or genuine reviews. It also categorizes the sentiment as positive, negative, or neutral. This classification is important for businesses to understand customer feedback effectively[9].

8. Large Language Model - DistilBERT

Once NLP is complete, the processed data is fed into large language models like **DistilBERT** and . These models are pre-trained on massive datasets and are capable of understanding complex linguistic patterns. They provide contextual embeddings that help in accurate classification and interpretation of text. DistilBERT is a lighter version of BERT, optimized for speed, while RoBERTa offers improved performance on classification tasks. These models add intelligence to the system, making it capable of deep text understanding. Their use significantly boosts the system's reliability and precision[4].

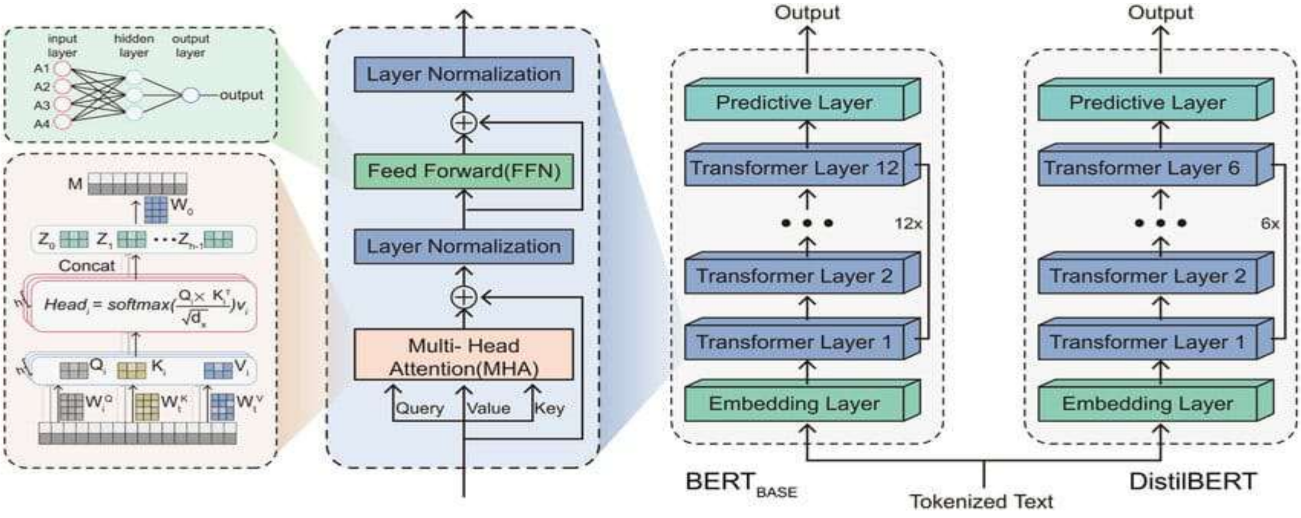


Fig:2 Distilled Bert Daigram

9. Vector Database

The final step involves storing the output of the large language models into a **Vector Database**. Unlike traditional databases, vector databases store high-dimensional vector representations of text. This makes it easier to perform similarity searches, clustering, and semantic comparisons. For example, similar reviews can be grouped or retrieved quickly based on vector proximity. These databases are essential for scalable and real-time text analytics. They enable the system to handle large volumes of data efficiently while maintaining performance.

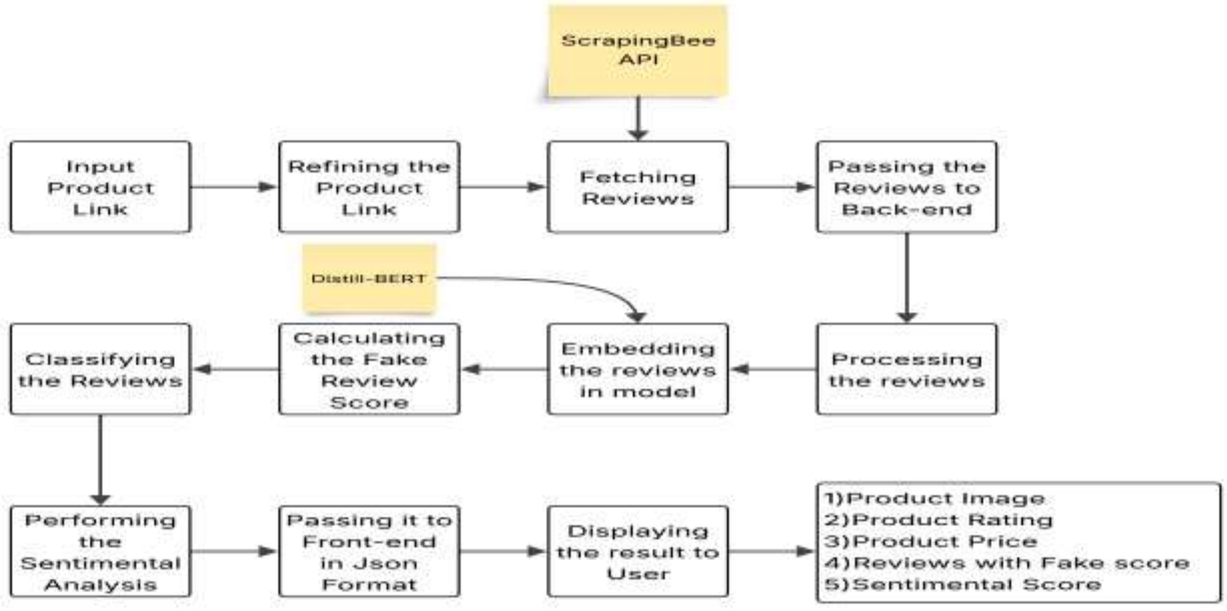

ScrapingBee API
Input Product Link
Refining the Product Link
Fetching Reviews
Passing the Reviews to Back-end
Distill-BERT
Classifying the Reviews
Calculating the Fake Review Score
Embedding the reviews in model
Processing the reviews
Performing the Sentimental Analysis
Passing it to Front-end in Json Format
Displaying the result to User
1)Product Image
2)Product Rating
3)Product Price
4)Reviews with Fake score
5)Sentimental Score

Fig2: Project Working Architecture

1. **Input Product Link:**

The system begins with the user providing a product link from an e-commerce platform. This link acts as the source from which all the relevant reviews will eventually be fetched. It is important for the link to be accurate and valid because any mistake here can break the entire chain of processing. The input needs to lead directly to a product page, avoiding irrelevant pages like homepage links or ad links. Essentially, this step sets the foundation for the entire workflow to follow.

2. **Refining the Product Link:**

After receiving the raw product link, the next task is to refine it. Many times, URLs carry extra tracking information, session IDs, or unnecessary parameters that can hinder further processing. Refining the link involves cleaning it up to retain only the core product identifier or necessary URL structure. This ensures that when the link is later used for scraping, the request remains consistent and doesn't trigger errors. Refining at this early stage boosts the efficiency and reliability of data fetching later.

3. **Fetching Reviews (using ScrapingBee API):**

Once the link is refined, the system uses the ScrapingBee API to fetch the actual reviews from the product page. ScrapingBee acts as a smart scraper that handles JavaScript rendering, bot detection, and CAPTCHA challenges, making it easier to collect data from protected websites. Through this API call, the system extracts user reviews, ratings, and sometimes other product details like name and price. The richness and correctness of the fetched reviews play a critical role in how accurate and valuable the final analysis will be. Thus, this fetching phase is vital for a strong input base.

4. **Passing the Reviews to Back-end:**

After fetching, the collected reviews are passed securely to the backend server where most of the heavy processing happens. This handoff must be fast and reliable to ensure no reviews are lost in transit. The backend acts as the control center, managing the reviews and preparing them for cleaning, analysis, and further processing steps. It also handles any exceptions or errors that might arise due to incomplete or malformed data. A robust back-end system is critical to maintain the workflow’s smoothness and data integrity.

5. **Processing the Reviews:**

In the backend, the raw review data undergoes several cleaning processes. Reviews may contain irrelevant characters, HTML tags, emojis, or incomplete sentences that need to be filtered out. Processing also involves normalizing the text — converting everything to lowercase, removing extra spaces, and sometimes correcting minor spelling errors. Cleaned and normalized text ensures that the later machine learning models work with high-quality inputs. Poorly processed reviews could lead to bad model performance, so this step ensures data quality and consistency.

6. **Embedding the Reviews in Model (using DistilBERT):**

Once the reviews are processed, they are converted into dense numerical vectors, called embeddings, using the DistilBERT model. DistilBERT is a smaller and faster version of BERT, designed for efficient text representation while maintaining high accuracy. The embedding process transforms the review text into a machine-understandable format that captures semantic meaning. These embeddings preserve the context and emotion of the original reviews, making them suitable for the next classification steps. Without good embeddings, even powerful models would struggle to classify the reviews effectively.

7. **Calculating the Fake Review Score:**

With the embedded reviews, the system now moves toward calculating a fake review score for each review. This score estimates the likelihood that a review is fake, based on linguistic patterns, sentiment mismatches, and hidden features learned by the model. This stage leverages the DistilBERT-powered embeddings to make more informed predictions. Higher scores indicate a greater chance of a review being fake, helping users understand the trustworthiness of the content. The accuracy of these scores heavily depends on how well the model has been trained and fine-tuned.

8. **Classifying the Reviews:**

Based on the fake review score, each review is classified as either genuine or fake. This binary or probabilistic classification helps in separating reliable feedback from suspicious or promotional reviews. Proper classification ensures that users receive an honest overall sentiment about a product, rather than being influenced by manipulated reviews. The classification results are also used for summarizing and displaying trust indicators to the user later. A good classification model needs to balance sensitivity (catching fake reviews) and specificity (not mislabeling real ones).

9. **Performing the Sentimental Analysis:**

After classification, sentiment analysis is performed on the genuine reviews. This analysis detects whether a review is positive, negative, or neutral based on the words and tone used. Sentiment scores give users a quick understanding of general customer satisfaction about the product. Even if a review is genuine, knowing whether it's positive or negative adds another layer of decision support for the customer. Sentiment analysis complements fake review detection by painting a complete picture of customer opinion.

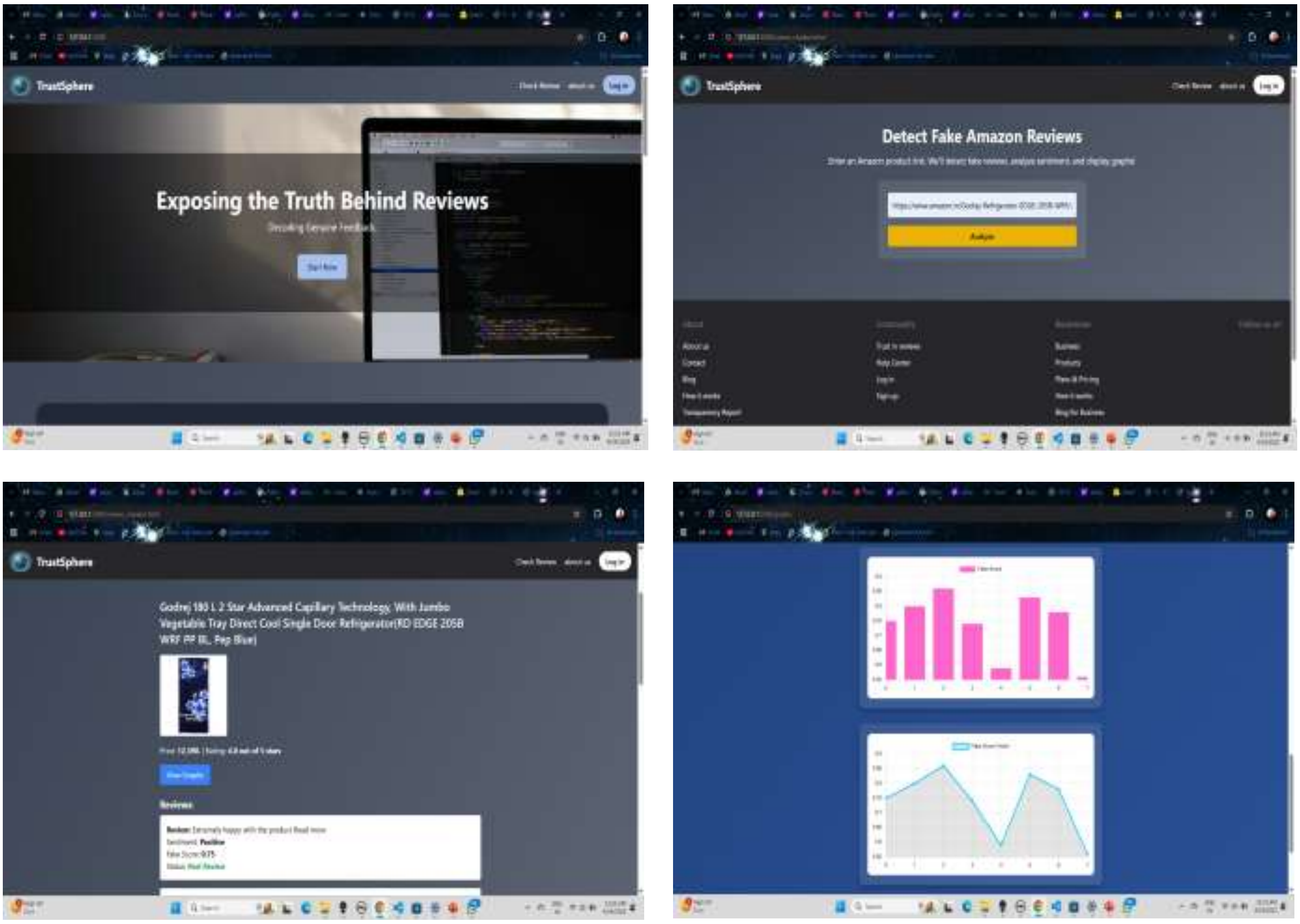
10. **Passing it to Front-end in JSON Format:**

All the processed information — including fake review scores, classifications, and sentiment results — is bundled and sent to the front-end in JSON format. JSON is lightweight, easy to parse, and perfect for web applications. This step enables the user interface to easily extract and present different pieces of information, such as review text, scores, and sentiments. Proper structuring in JSON ensures that the front-end development is flexible and scalable.

11. **Displaying the Result to User:**

Finally, all results are displayed neatly to the user on the front-end interface. Users can see product details like the product image, product rating, price, individual reviews tagged with fake scores, and overall sentiment scores. This visualization helps users make better purchasing decisions by giving them transparent insights into the authenticity and overall feeling about the product. A good UI/UX here ensures that users can quickly understand the output without feeling overwhelmed or confused.

5. IMPLEMTATION DETAIL



Godrej 180 L 2 Star Advanced Capillary Technology With Jumbo Vegetable Tray Direct Cool Single Door Refrigerator(RD EDGE 205B WRF PF BL, Pep Blue)

Review: Extremely happy with the product Read more
Sentiment: Positive
Fake Score: 0.75
Status: Real Review

Fig3: Outputs Of Project

Figure 3 illustrates the output of the the project begins with a straightforward and easy-to-use front-end where users are greeted by a tidy landing page. It gives an overview of the system's goal, which is to reveal the real story behind internet reviews. The user can enter an Amazon product link in the designated textbox on the main functionality page (first image) once they are prepared. When the "Analyze" button is clicked, the system is set up to instantly retrieve the related reviews, guaranteeing a seamless transfer from user input to backend processing.

The backend of the system starts working as soon as the user submits a product link. It uses APIs like ScrapingBee to scrape the reviews and then runs them through a DistilBERT model that has been trained. Two important outputs are computed by this model: a sentiment score for every review and a fake review score. After the analysis is finished, users are shown a comprehensive result page that includes the list of reviews and product details (like the product name, image, and price), as shown in the third screenshot. To make it easier for users to assess the reviews' credibility, the platform displays the sentiment polarity (positive, negative, or neutral) and fake score for each review.

The platform provides visual data representations through charts, as seen in the fourth image, to further enhance the output's insight. The system creates graphs that display sentiment trends or fake score averages, as well as the distribution of fake review scores throughout the dataset. Users can quickly spot patterns in the reviews, like spikes in possibly fraudulent reviews or changes in sentiment, with the aid of these graphical insights. Through the integration of visual analytics and text-based outputs, the system improves transparency and trust in product evaluation.

All things considered, the architecture exhibits a smooth fusion of strong backend machine learning analysis, user-friendly visualization, and front-end user interaction. The project offers a comprehensive end-to-end solution for identifying fraudulent Amazon reviews, from entering a product link to evaluating reviews and ultimately presenting clear results. Together, the user-friendly interface, automated backend, and educational outputs guarantee that users can make more intelligent and knowledgeable purchasing decisions.

6. CONCLUSION

This project highlights notable developments in review analysis and fraud detection by providing a thorough assessment of a fake review detection system for Amazon products. By incorporating cutting-edge machine learning techniques—most notably, the DistilBERT model for review embedding and classification—one of the work's main results is an enhanced capacity to evaluate review authenticity. Even when dealing with noisy, unstructured data, the system's end-to-end architecture, which combines automated data scraping, sophisticated preprocessing, and sentiment analysis, guarantees high reliability. By allowing users to easily visualize fictitious scores and sentiments through interactive graphs, the user-centric front-end further improves accessibility.This project has significant wider ramifications. The capacity to automatically detect fraudulent reviews presents a potent tool for boosting platform integrity and gaining customer trust as e-commerce continues to rule the retail market. These kinds of systems, which combine deep learning and user-friendly design, can be extremely important for protecting online marketplaces, assisting with well-informed purchasing decisions, and preserving fair competition in the digital economy.

7. ACKNOWLEDGEMENT

We would like to express my heartfelt gratitude to all those who contributed to the success of this research project on Fake Review Detection Using Distelled-Bert Model First and foremost, we extend my sincere appreciation to Our research supervisor Prof. Balaji A. Chaughule for their invaluable guidance, support, and encouragement throughout the process. Their expertise and insights were instrumental in shaping this work.

8. REFERENCES

[1] Dr. Atika QaziDr, Dr. Najmul Hasa, Dr. Rui Mao, "Machine Learning-Based Opinion Spam Detection: A Systematic Literature Review", vol.[8], no. 1, January 2024.

[2] M. F. MRIDHA, ASHFIA JANNAT KEYA"A Comprehensive Review on Fake News Detection With Deep Learning", journal of networks, vol.[10], no. 1, November 10, 2021.

[3] RAMI MOHAWESH, SHUXIANG XU, SON N. TRAN " Fake Reviews Detection: A Survey", journal of networks, vol.[12], no. 1, April 2021.

[4] Junwen Lu Xintao Zhan ,*, Guanfeng Liu , Xinrong Zhan 1 and Xiaolong Deng "BSTC: A Fake Review Detection Model Based on a Pre-Trained Language Model and Convolutional Neural Network" 9 May 2023.

[5] Rahul Kumar, Shubhadeep Mukherjee, Nripendra P. Rana " Exploring Latent Characteristics of Fake Reviews and Their Intermediary Role in Persuading Buying Decisions" 24 May 2023.

[6] K. Pooja, Pallavi Upadhyaya "What makes an online review credible? A systematic reviewof the literature and future research directions" 5 Dec 2022.

[7] Abhijeet A Rathore, Gayatri L Bhadane, Ankita D Jadhav, Kishor H Dhale" Fake Reviews Detection Using NLP Model and Neural Network Model " 5 May 2023.

[8] Ahmed M. Elmogy, Usman Tariq,Atef Ibrahim4 "Fake Reviews Detection using Supervised Machine Learning"2021.

[9] Ms. Rajshri P. Kashti, Dr. Prakash S. Prasad "Enhancing NLP Techniques for Fake Review Detection " Feb 2019.

[10] Naznin Sultana,Prof. Sellappan Palaniappan " Deceptive Opinion Detection Using Machine Learning Techniques " 8 Feb 2020.