School of Computing, Engineering and Digital Technologies

Middlesbrough TS1 3BA

Natural Language Processing and Machine Learning-Based Sentiment Predictor for Amazon Devices

Submitted in partial requirements for the degree of MSc Data Science

January 10th, 2025

Tran Thuy Vy

S3369268

Email: S3369268@live.tees.ac.uk

Supervisor: Dr. Francis Ho

Acknowledgment

I would like to express my sincere gratitude to my supervisor, Dr. Francis Ho, for his invaluable guidance, feedback and support throughout the course of this project. His expertise and mentorship have been pivotal in shaping the direction and outcome of this research.

I would also like to extend my appreciation to Teesside University for providing the opportunity to pursue this program and for the resources and knowledge that have been instrumental in the successful completion of this work.

Lastly, I am grateful to my family and friends for their continuous support and encouragement throughout my academic journey. Their support has been a significant factor in enabling me to achieve my goals.

Abstract

Sentiment analysis has become an essential tool for understanding customer feedback, particularly in e-commerce, where product reviews play an important role in decision-making. This project aimed to develop a sentiment analysis tool for Amazon device reviews using advanced Natural Language Processing (NLP), Machine Learning (ML), and Deep Learning (DL) techniques to classify sentiments as positive, neutral, or negative.

The project involved key data preprocessing steps like tokenization, lemmatization, and Term Frequency-Inverse Document Frequency (TF-IDF) for feature extraction. Six models were developed and tested: five Machine Learning models: LightGBM, Naïve Bayes, Logistic Regression, Random Forest, LinearSVC and one Deep Learning model, Long Short-Term Memory (LSTM). LightGBM was the best-performing model which achieved an accuracy and F1-score of 0.80, handling imbalanced datasets and high-dimensional features effectively.

The project produced ta user-friendly web application for real-time sentiment classification and a Power BI dashboard to visualize sentiment distribution, product performance, and review trends. While the tool effectively classified clear positive and negative sentiments, it faced challenges with mixed or complex sentiments, particularly in the neutral class.

This project demonstrates how sentiment analysis can extract useful insights from customer reviews to improve products and customer satisfaction. Future work will focus on expanding the dataset, using advanced embedding methods and improving the model to address these challenges and enhance its accuracy.

Table of Contents

1. Introduction

The growth of e-commerce platforms has changed how people interact with products and services, making customer reviews a key source of information for both businesses and buyers. Amazon, as a global leader, offers a wide range of products and gathers millions of customers reviews every day. These reviews contain valuable feedback about customer preferences, product quality, and satisfaction. However, the large volume of unstructured textual data presents significant challenges for manual analysis, creating the need for automated tools to extract meaningful insights.

This project focuses on sentiment analysis, a branch of Natural Language Processing (NLP), Machine Learning (ML) and Deep Learning (DL) to evaluate customer reviews of Amazon devices such as Alexa, Echo, Kindle Tablets and Fire TV. Sentiment analysis also known as opinion mining, involves categorizing textual data into positive, neutral or negative sentiments based on the emotions and opinions expressed (Pang & Lee, 2008). By applying advanced NLP, ML and DL techniques, this study aims to overcome the limitations of manual analysis, such as inefficiency and subjectivity, while addressing the challenges of traditional ML models in dealing with complex language patterns like sarcasm, ambiguity, and mixed sentiments.

A key feature of this project is the development of a web-based sentiment analysis tool that allows users to input individual or batch reviews for automatic classification. This tool aims to provide a user-friendly solution that simplifies sentiment analysis and makes it accessible to a broader audience, including businesses and individual users. Furthermore, the project integrates a Power BI dashboard to visualize key statistics, such as the distribution of sentiments, the number of reviews analyzed, top-performing products based on sentiment and key sentiment factors. This visual representation enhances the comprehensibility of the results and supports data-driven decision-making.

The motivation for this project comes from the need for businesses to understand customer opinions to stay competitive in a fast-changing market. Analyzing customer feedback enables businesses to identify issues, refine their products and enhance customer satisfaction. This project employs Machine Learning (ML) and Deep Learning (DL) techniques to develop and evaluate models for sentiment analysis, ultimately selecting the most effective model for accurate sentiment classification.

This introduction provides an overview of the key areas covered in the following sections, including the challenges of manual review analysis, the research questions guiding this study and the objectives aimed at addressing these challenges. By applying advanced NLP and ML techniques, this project seeks to contribute to the field of sentiment analysis and offer a practical solution for analyzing and understanding customer feedback effectively.

1.1. Research Questions

The following research questions guide this study:

1. What are the most effective NLP, Machine Learning, and Deep Learning techniques for analyzing sentiment in Amazon device reviews?

2. How do machine learning models compare to deep learning models, such as LSTM-based RNNs, in terms of sentiment classification accuracy and F1-score?
3. What are the key factors driving positive and negative sentiment for Amazon devices?
4. How can sentiment analysis insights improve product recommendations and enhance customer satisfaction?
5. What challenges arise in developing sentiment analysis tools, and how can these challenges be effectively mitigated?

1.2. Research Objectives

The primary objectives of this research are as follows:

1. To design and implement a sentiment analysis model using advanced NLP and ML techniques for classifying Amazon device reviews into positive, neutral, or negative sentiments.
2. To develop and evaluate five machine learning models (Logistic Regression, Naïve Bayes, Random Forest, Linear Support Vector Classifier, and Light Gradient Boosting Machine) and one Deep Learning model Long Short-Term Memory (LSTM) for their performance in sentiment analysis.
3. To compare the performance of ML models with the DL model, focusing on metrics like accuracy and F1-score and select the best-performing model for deployment.
4. To integrate the chosen model into a user-friendly web-based sentiment analysis application for sentiment classification.
5. To design a Power BI dashboard that delivers comprehensive statistical insights, including sentiment distribution, top products by sentiment, review trends over time and key sentiment factors. This dashboard helps users better understand sentiment data, providing actionable insights for more informed decision-making.

2. Literature Review

The study of sentiment analysis has evolved significantly with the advancements in Natural Language Processing (NLP), Machine Learning (ML) and Deep Learning (DL). This literature review explores existing research and methodologies relevant to analyzing customer reviews, particularly in the context of Amazon devices. It provides an overview of traditional ML techniques, advanced deep learning approaches, and ensemble methods such as LightGBM. Additionally, key challenges in sentiment analysis are discussed, along with gaps in current research that this project seeks to address. By examining these topics, this review establishes the foundation for the development and evaluation of sentiment classification models in this study.

2.1. Introduction to Sentiment Analysis

Sentiment analysis also known as opinion mining is a key application of Natural Language Processing (NLP) that aims to identify, extract, and classify opinions or emotions expressed in text. It typically involves categorizing text into positive, neutral, or negative sentiments based on the context and tone of the language (Pang & Lee, 2008). This technique has become an essential tool for analyzing user-generated content, particularly in domains like e-commerce, where understanding customer feedback can inform product development, marketing strategies, and service improvements (Cambria et al., 2013).

With the rise of platforms like Amazon, customer reviews have increased significantly, offering businesses useful insights into customer satisfaction and preferences. However, the large volume and unstructured format of these reviews make manual analysis difficult and inefficient (Medhat, Hassan, & Korashy, 2014). Sentiment analysis addresses this challenge by automating the processing and interpretation of reviews, allowing businesses to quickly and accurately gain actionable insights.

This overview explains the basics of sentiment analysis and its role in NLP. The next sections will explore traditional machine learning models and advanced methods in more detail.

2.2. Machine Learning Models

Machine learning (ML) models have been fundamental in sentiment analysis due to their efficiency and effectiveness in text classification (Sebastiani, 2002). These models convert unstructured text into structured numerical data using techniques such as Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW). This section discusses five ML models used in this project, including both traditional approaches and the advanced Light Gradient Boosting Machine (LightGBM).

2.2.1 Traditional Machine Learning Models

Logistic Regression (LR) is a widely used algorithm for text classification and sentiment analysis due to its simplicity and efficiency. It uses a linear decision boundary to predict sentiment classes based on input features. LR works well for datasets where sentiment distinctions are clear but struggles with capturing complex relationships, such as negations or sarcasm (Ng & Jordan, 2001).

Naïve Bayes (NB) is a probabilistic model that classifies text by calculating the likelihood of features for each sentiment class. Its simplicity and ability to handle large amounts of data make it effective for tasks like sentiment analysis. However, its assumption of feature independence can result in inaccuracies, particularly in text with strong contextual relationships (Zang, 2004).

Random Forest (RF) is an ensemble method that uses multiple decision trees to classify text data. It improves accuracy and reduces overfitting by combining predictions from different trees. While effective for text classification with complex features, it can be slow with large datasets (Breiman, 2001).

Linear Support Vector Classifier (LinearSVC) is a type of Support Vector Machines (SVM) that uses a linear kernel for classification task. It is highly effective for text classification, especially in high-dimensional data generated by BoW or TF-IDF. It separates sentiment classes by maximizing the margin with a linear decision boundary. Although accurate and reliable, it can be affected by noisy data and needs careful tuning of parameters (Joachims, 1998)

2.2.2 Advanced Machine Learning Model: Light Gradient Boosting Machine

LightGBM is an advanced ensemble algorithm known for its speed, scalability, and accuracy in sentiment analysis. Unlike traditional ML models, it uses a leaf-wise tree growth strategy, which enables it to capture complex interactions between features. LightGBM handles large and imbalanced datasets effectively, making it a strong choice for text classification tasks (Ke et al., 2017).

LightGBM has several strengths that make it effective for sentiment analysis. It handles high-dimensional features from methods like TF-IDF and Bag of Words efficiently, making it suitable for sparse text data (Ke et al., 2017). It also performs well on imbalanced datasets, effectively managing cases where some sentiment classes are less common (Dube & Verster, 2023). Another advantage is its ability to highlight important features, showing which words or phrases impact predictions the most (Alzamzami et al., 2020). These strengths make LightGBM a reliable choice for text classification, especially with large or complex datasets.

While LightGBM addresses many limitations of traditional ML models, it is less effective at capturing sequential relationships in text, such as word order or context (Naji et al., 2024). These challenges are better handled by deep learning models, which are discussed in the next section.

2.2.3 Strengths and Limitations

Traditional ML models are efficient and effective for text classification tasks where sentiment distinctions are well-defined. Their reliance on feature extraction methods like TF-IDF makes them suitable for structured datasets. However, they often struggle with complex sentiment patterns, such as sarcasm, mixed emotions, or context-dependent phrases. LightGBM improves these limitations by offering better performance and reliability, but it still cannot capture sequential dependencies, which are important for understanding complex text.

The next section introduces the Long Short-Term Memory (LSTM) model, a deep learning approach designed to address these limitations by focusing on the sequential nature of text data.

2.3. Deep Learning Model: Long Short-Term Memory

Deep learning has greatly improved sentiment analysis by addressing the challenges faced by traditional machine learning methods. Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN), are particularly effective for analyzing sequential data, such as text (Young et al., 2018). LSTM models were introduced by Hochreiter and Schmidhuber (1997) to overcome the vanishing gradient problem in traditional RNNs, making it possible to understand long-term dependencies in data.

In sentiment analysis, capturing sequential relationships is key for understanding word context (Socher et al., 2013). For example, the phrase "not bad" relies on the order of words to convey meaning, which traditional models often miss. LSTMs overcome this limitation by retaining important context across sequences, making them ideal for sentiment analysis tasks (Goodfellow et al., 2017).

LSTMs are highly effective in sentiment analysis as they can understand word order and context, making them ideal at handling complex text like sarcasm or negations (Hochreiter & Schmidhuber, 1997). Their flexibility allows them to handle varying input lengths, making them adaptable to different text sizes. However, LSTMs also have limitations. They require significant computational resources, especially for large datasets, and perform best with diverse data, which may not always be available. Without proper adjustments, they can also overfit, especially when the dataset is small (Zhang et al., 2021).

In this project, LSTM is used alongside traditional ML models to classify sentiment. Its ability to handle context makes it valuable for text with complex patterns, and its performance will be compared with models to identify the best approach.

2.4. Key Challenges in Sentiment Analysis

Despite advancements, sentiment analysis still faces many challenges that make accurate classification difficult. These challenges are common when dealing with complex or unstructured text data.

2.4.1 Ambiguity and Mixed Sentiments

Text often contains ambiguous or mixed sentiments, making it difficult to classify accurately. For example, a review might include both positive and negative statements, such as "The product quality is great, but the delivery was delayed." These mixed emotions can confuse models, especially those lacking contextual understanding (Zhao et al., 2016).

2.4.2 Sarcasm and Figurative Language

Sarcasm and phrases with hidden meanings are confusing for models to understand. Phrases like "Oh great, another delay" carry negative sentiment despite using positive words (Maynard & Greenwood, 2014). Traditional models often struggle with these complexities, highlighting the need for advanced techniques to capture implied meanings effectively (Poria et al., 2018).

2.4.3 Imbalanced Datasets

Imbalanced datasets where one type of sentiment is less common are a frequent challenge in real-world applications. For example, neutral or negative sentiments often occur less frequently than positive ones (Japkowicz & Stephen, 2002). This imbalance can cause models to favor the more common sentiment. Methods like oversampling, undersampling or class weighting can help address this problem, but they are not always effective (Buda et al., 2018).

2.4.4 Context Dependency

The sentiment of a phrase often depends on its context, which makes classification harder. For example, "not bad" has a positive sentiment even though it includes the negative word "not." Models that miss these details can give inaccurate results, showing the importance of sequential and contextual analysis (Hochreiter & Schmidhuber, 1997).

2.4.5 Data Quality and Noise

Real-world datasets often contain noise, such as spelling errors, abbreviations, and slang. These issues can degrade the performance of sentiment analysis models, especially those relying on word-based features (Eisenstein, 2013). Preprocessing techniques, like tokenization and text normalization, can address these problems but may not fully resolve them (Pradha et al., 2019).

These challenges show where current sentiment analysis models need improvement. Solving these issues is essential to make sentiment analysis more accurate and reliable in diverse situations.

2.5. Summary of Literature Review

This literature review has explored the foundations and advancements in sentiment analysis, focusing on the use of machine learning (ML) and deep learning (DL) models. Traditional ML models like Logistic Regression, Naïve Bayes, Random Forest, and LinearSVC have shown efficiency and effectiveness with structured datasets that have clear sentiment patterns. LightGBM, an advanced ML model, improves performance by handling imbalanced datasets better but struggles to capture the order and context of words.

Deep learning models, especially Long Short-Term Memory (LSTM) networks, have further improved sentiment analysis by understanding context and word sequences, making them suitable for complex language. However, challenges such as handling ambiguity, sarcasm, imbalanced datasets, and domain-specific language persist, highlighting areas for continued research and innovation.

This study combines insights from both ML and DL methods to address these challenges and improve sentiment analysis. The next sections will outline the methodology and experimental design, building on the findings from this review.

3. Methodology

This section outlines the processes and techniques used in this project, from data collection and preprocessing to model development, evaluation, and deployment.

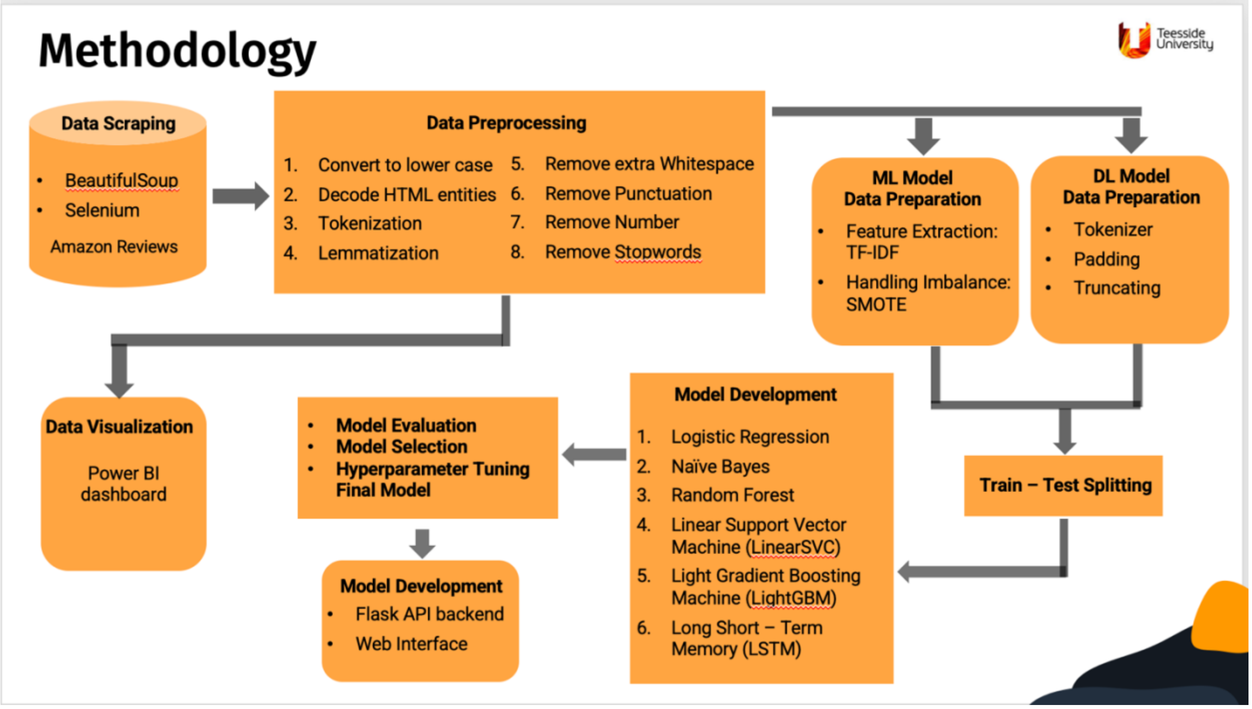

FIGURE 1. METHODOLOGY DIAGRAM

3.1. Data Collection

The dataset for this project was collected by scraping reviews for **Amazon Devices** from the [Amazon Devices Review Page](). The process employed **BeautifulSoup** and **Selenium**, two widely recognized Python libraries for efficient web scraping.

BeautifulSoup is a Python library designed for parsing HTML and XML documents. It allows for easy extraction of specific elements, such as text, links, or metadata, from web pages. Its ability to navigate, search, and modify HTML trees makes it an essential tool for data scraping projects (Richardson, 2007).

Selenium is a web automation tool primarily used for testing web applications but also widely employed for web scraping. It enables dynamic interaction with web pages, such as clicking buttons or filling out forms, which is useful for extracting data from sites requiring user interaction (Avasarala, 2014). In this project, Selenium was used to handle Amazon's review pages, which load dynamically as users scroll or interact with the page. It ensured that all reviews were fully loaded before extracting data.

The following steps were performed to collect the data:

3.1.1 Dynamic Web Page Handling with Selenium

Selenium was used to interact with Amazon's website, load review pages, and handle dynamic content.

3.1.2 Extracting Data with BeautifulSoup

BeautifulSoup parsed the loaded pages to extract details such as: ReviewID, Review content, Review title, Ratings, Review dates, ProductID, Product link.

```
# Import libraries
from bs4 import BeautifulSoup
from selenium import webdriver

# Initialize the ChromeDriver service for Selenium
webdriver_service = Service("/usr/local/bin/chromedriver")

# Create a new Chrome browser instance
driver = webdriver.Chrome(service=webdriver_service, options=chrome_options)

driver.get(review_link)  #Load each review page
time.sleep(5)  # Wait for content to load

review_divs = driver.find_elements(By.CSS_SELECTOR, 'div[data-hook="review"]')
time.sleep(2)

soup = BeautifulSoup(driver.page_source, 'html.parser')
```

FIGURE 2. SELENIUM AND BEAUTIFULSOUP CODE USED IN THE PROJECT

3.1.3 Data Storage

The scraped data was stored in a structured file: **amazon_reviews.xlsx**

1	Review_id	Reviewer	Rating	Date	Review title	Review content	Product_id	Product_title	Product_link
2	R192QJ45JRSLTC	Chris	5	13/07/2024	Didn't think I needed it until I	The Blink Subscription Basic Plan offers essential feature	B08JHCVHTY	Blink Subscript	https://www.am
3	RLJN0G2I0CRNC	Amazon Custo	5	12/10/2024	Worth every penny!	I have had Blink cameras at my house for years now! The	B08JHCVHTY	Blink Subscript	https://www.am
4	R19D78F9YK0DVA	uniquely uniq	5	11/10/2024	I'm quite satisfied	I've been using the Blink Subscription Plus Plan with the	B08JHCVHTY	Blink Subscript	https://www.am
5	R2W7QUYHDCN6CB	Lyndi Dawn M	4	26/09/2024	Very Nice Added Security	Really like having these cameras, they give us a strong se	B08JHCVHTY	Blink Subscript	https://www.am
6	RM9R0N4N310DC	ccoulson90	5	21/10/2024	Great Item	The Blink Subscription Plus Plan is a fantastic investmen	B08JHCVHTY	Blink Subscript	https://www.am
7	R2CULQ1YL9X8OH	4 Lil Lambs	3	27/08/2024	Doesn't do much .. But nothin	I'm not impressed with the blink system at all. It spends most of it's time offline and missing	B08JHCVHTY	Blink Subscript	https://www.am
8	RM6T053CTY7A1	A Y G	5	1/9/2024	Provides peace of mind	We have had a couple of blink cameras for a couple of ye	B08JHCVHTY	Blink Subscript	https://www.am
9	R3S06Q8N1V4R48	Patricia Lindle	4	23/10/2024	Great customer service.	We found the batteries need replaced 2-3 times a year. It	B08JHCVHTY	Blink Subscript	https://www.am
10	R1RL7VKZWV950H	Samrat Roy	5	3/7/2024	The functionality and security	The Blink Subscription Plus Plan is an upgrade for users c	B08JHCVHTY	Blink Subscript	https://www.am
11	RVSOXBAUTA8ER	Emma Benado	4	17/10/2024	pretty good	Your basically forced to purchase it. My husband and i h	B08JHCVHTY	Blink Subscript	https://www.am
12	R2G8P9ZTCXT0T3	KWLVT	4	31/08/2024	Great but Pricy	I tried the Blink subscription plan for a month to evaluat	B08JHCVHTY	Blink Subscript	https://www.am
13	RR0TQSI9VG9BK	Pittie Mom	5	27/10/2024	Blink outdoor camara subscri	When you purchase the wireless outdoor blink cameras	B08JHCVHTY	Blink Subscript	https://www.am
14	R2BAYCDCPH2O09	Karen	4	27/10/2024	Blink doorbell	I like the basic plan, however, about every two weeks th	B08JHCVHTY	Blink Subscript	https://www.am
15	R2QHLZC53OIL9W	J. D. Wright	4	18/08/2024	Police won't do anything with	I have a next door neighbor who ran his mouth to me to	B08JHCVHTY	Blink Subscript	https://www.am
16	R2OW86CJ35Y6TQ	Amazon Custo	5	11/10/2024	Blink awesome	Purchased Blink security cameras on a whim to keep up	B08JHCVHTY	Blink Subscript	https://www.am
17	R1PLFRV4CI5DMK	Jess Tilton	4	5/10/2024	It helps	You definitely do not need a subscription to use your Bli	B08JHCVHTY	Blink Subscript	https://www.am
18	ROF00RJXGCWML	Stridar	4	22/10/2024	Works well and affordable.	Excellent function during the day … night time I have to	B08JHCVHTY	Blink Subscript	https://www.am
19	R1LOMMPXVLX5H4	Wise Mom	4	12/10/2024	Love Blink Cameras	I have had Blink cameras for many years now and origina	B08JHCVHTY	Blink Subscript	https://www.am
20	R3AK0MRAPUYO0I	Michael	4	19/10/2024	Must have	We've had this for some years now, and have been VERY	B08JHCVHTY	Blink Subscript	https://www.am
21	R2F1YA5ODAFI1C	K Otto	5	13/10/2024	Excellent plan for price	Have had for a year, EXCELLENT!GREAT VALUE for price.V	B08JHCVHTY	Blink Subscript	https://www.am
22	R339B3EVJYU2F3	Tarrah L	4	13/10/2024	Blink is great, doesn't always r	Works great most of the time. Picture quality is good. H	B08JHCVHTY	Blink Subscript	https://www.am
23	R2284TYE5EIHEH	C S	5	27/10/2024	Great Product!	Product is great for sensors and the things the camera ca	B08JHCVHTY	Blink Subscript	https://www.am
24	R2DYOVZTTRXOMG	Ashley Ranson	5	26/10/2024	I love it	It's so nice to know what is happening in my front yard a	B08JHCVHTY	Blink Subscript	https://www.am
25	R399JLWS60SRSC	Ginny	5	19/10/2024	I love these	I am so happy with these cameras. I can see what's happ	B08JHCVHTY	Blink Subscript	https://www.am

FIGURE 3. SAMPLE OF SCRAPED DATASET

The scraped dataset contains **15,862** rows and **9** columns, with each row representing a unique review and the columns including details like: Review Dd, Reviewer name, Rating, Date, Review Title, Review Content, Product ID, Product Title, Product Link. This dataset is the basis for preprocessing and analysis in the project.

3.2. Data Preprocessing

3.2.1 Data Cleaning and Preprocessing

To ensure clean and consistent data, several data preprocessing steps were applied:

- **Handle Null Values**
  Rows with significant missing data were removed or imputed to ensure data integrity.
- **Create New Columns**
  Review Length column was added to calculate the character count of each review. Word Count column was added to compute the total number of words in each review.
- **Combine Title and Content**
  The *review_title* and *review_content* columns were merged into a single column to create a unified text input for analysis.
- **Generate Sentiment Column**
  Sentiment column was created to classify reviews as positive, neutral, or negative, based on predefined thresholds.

```python
def sentiment_categorization(row):

    ''' Assigns a sentiment value based on the user's rating.'''

    if row['Rating'] < 3.0:
        val = 'Negative'
    elif row['Rating'] > 3.0:
        val = 'Positive'
    else:
        val = 'Neutral'

    return val
```

FIGURE 4. FUNCTION TO CLASSIFY REVIEWS

- **Handle Date Column**
  review_date column was split into separate columns for **day**, **month**, and **year** for analysis of review trends.

3.2.2 Text Mining

The following NLP techniques were applied in this project to preprocess text data:

- **Lowercasing**
  All text was converted to lowercase for consistency across the dataset.
- **Decoding HTML Entities**
  HTML tags and encoded entities were removed to clean the text.

- **Tokenization**

  Tokenization is the process of breaking text into smaller units called tokens, such as words or sentences (Jurafsky, 2000). It is an essential step in NLP, making text easier to analyze and process. In this project, tokenization was applied at the word level, this allows each word in a review to be processed separately for better analysis and model performance.

- **Lemmatization**

  Lemmatization is the process of reducing words to their base or root form, while ensuring the resulting word is meaningful and grammatically correct (Jurafsky, 2000). For example, the word "running" is lemmatized to "run." This step helps standardize words in the dataset, improving the consistency and accuracy of text analysis. In this project, lemmatization was applied to handle variations in word forms effectively.

```python
# Initialize lemmatizer
lemmatizer = WordNetLemmatizer()

# Function to convert NLTK POS tags to WordNet POS tags
def get_wordnet_pos(tag):
    if tag.startswith('J'):
        return wordnet.ADJ
    elif tag.startswith('V'):
        return wordnet.VERB
    elif tag.startswith('N'):
        return wordnet.NOUN
    elif tag.startswith('R'):
        return wordnet.ADV
    else:
        return wordnet.NOUN  # Default to noun if POS tag is unknown

# Function to tokenize, remove punctuation marks and lemmatize with POS tagging
def tokenize_and_lemmatize(text):

    # Step 1: Tokenize the text
    tokens = word_tokenize(text)

    # Step 2: Get POS tags for each token
    pos_tags = pos_tag(tokens)

    # Step 3: Lemmatize each token based on its POS tag
    lemmatized_tokens = [lemmatizer.lemmatize(token, get_wordnet_pos(tag)) for token, tag in pos_tags]

    return " ".join(lemmatized_tokens)

# Apply the function to the 'Full_review' column
process_reviews['Full_review'] = process_reviews['Full_review'].apply(tokenize_and_lemmatize)
```

FIGURE 5. CODE FOR TOKENIZATION & LEMMATIZATION

- **Removing Punctuation**

  Punctuation marks such as periods, commas, and special characters were removed, as they do not carry meaningful information for sentiment classification (Jurafsky, 2000). This step ensures cleaner text and prevents unnecessary noise during analysis.

- **Removing Numbers**

Numerical values that are irrelevant to the analysis in this project were removed to focus on meaningful textual content. This step improves the overall clarity and relevance of the dataset for sentiment classification.

- **Stopword Removal**

  Stopwords are common words like “and”, “the” and “is” that appear frequently but add little value to sentiment analysis. Removing stopwords reduces noise in the data and helps the model focus on more important terms that influence sentiment (Jurafsky, 2000).

Before:

```
Output review example:
doesn't do much .. but nothing works without it. i&apos;m not impressed with the blink system at all. it spends most of it&apos;s time
offline and missing. currently, i have 2 mini blinks ... which work the best - if they are plugged in. currently, one is missing, and the
other is often unplugged because it is annoying if someone is working in the backyard. we have 2 doorbells. one never seems to be working,
but the front door at least rings through. i rarely get pictures from it anymore.

we have 2 spotlights ... one requires the little white box to work that stores the pictures and uploads to the blink plan ... however, if
the white box fails, you have no way to contact or store the pictures. it&apos;s been ages since the white storage box quit working, and i
can&apos;t find the replacement. my husband says he&apos;ll get around to installing the second spotlights, but he&apos;s been saying that
for 2 years now.

overall, i would not recommend the blink as a system. i&apos;m torn between starting over with a ring system and trying to get the blink
system working again. super frustrated with the hidden costs and unreliability. when it works, it&apos;s nice ... though it is always
delayed, so we never get faces or actions ... frequently it records backs as people leave or nothing at all, as the person has already
gone.

we got our first blink almost 5 years ago ... we got two mini blinks for our apartment. so we got doorbells when we got our own house, just
over 2 years ago, since nobody could hear the knocking on the front door or garage door. they worked great for about 6 months.

so, i&apos;m stuck. can&apos;t hear the doorbell without the plan. the plan is super overpriced just to hear the doorbell.
```

After:

```
Output review example:
not much but nothing work without not impress blink system spend time offline miss currently mini blink work best plug currently one miss
often unplugged annoy someone work backyard doorbell one never seem work but front door least ring rarely get picture anymore spotlight one
require little white box work store picture uploads blink plan however white box fails no way contact store picture age since white storage
box quit working not find replacement husband say get around instal second spotlight but say year overall would not recommend blink system
tear start ring system try get blink system work super frustrate hidden cost unreliability work nice though always delay never get face
action frequently record back people leave nothing person already go get first blink almost year ago get two mini blink apartment get
doorbell get house year ago since nobody could hear knocking front door garage door work great month stick not hear doorbell without plan
plan super overprice hear doorbell
```

FIGURE 6. PROCESSED REVIEW EXAMPLE AFTER TEXT PREPROCESSING

3.3. Data Visualization

After data preprocessing, a separate Power BI dashboard was created to visualize the sentiment analysis results using the cleaned dataset. This dashboard effectively bridges the gap between raw data and meaningful insights, it presents the results in a clear and interactive way. It includes visualizations such as sentiment distribution, top-performing products by sentiment, review trends over time and key sentiment factors. These elements provide a comprehensive understanding of the data and support informed decision-making.

3.4. Data Preparation

Data preparation involves transforming the preprocessed dataset into a format suitable for Machine Learning (ML) and Deep Learning (DL) models. This step ensures that the models can effectively learn from the data.

3.4.1 Machine Learning Workflow

a) Feature Extraction

Feature extraction is the process of converting raw text data into meaningful numerical features that ML models can use. In this project, **TF-IDF (Term Frequency-Inverse Document Frequency)** was used as the feature extraction technique to transform text reviews into numerical representations.

TF-IDF assigns a weight to each word in a review based on its importance within the review and across the dataset. This helps the machine learning models focus on meaningful words while ignoring common words that appear frequently in all reviews (Ramos, 2003).

N-grams (Unigrams and Bigrams): Capturing both individual words and word pairs to better understand word context and relationships.

```
# Convert text data into TF-IDF features
vectorizer = TfidfVectorizer(
    max_features=5000,
    ngram_range=(1,2),
    min_df=1                # Includes terms that appear in at least 1 document.
    )

# Generate the TF-IDF feature matrix
X_tfidf = vectorizer.fit_transform(X)
```

FIGURE 7. CODE SNIPPET DEMONSTRATES THE IMPLEMENTATION OF TF-IDF AND N-GRAM

b) Train-Test Split

After applied TF-IDF, the dataset was split into training and testing sets with an 80:20 ratio.

c) Handling Imbalance: SMOTE

Class imbalance occurs when certain sentiment categories (positive, neutral, negative) are overrepresented or underrepresented in the dataset. In this project, the dataset showed noticeable class imbalance, as seen in the distribution:

```
Class distribution before train-test split:
Sentiment
Positive    8327
Neutral     4020
Negative    3444
Name: count, dtype: int64

Class distribution in training set before SMOTE:
Sentiment
Positive    6629
Neutral     3236
Negative    2767
Name: count, dtype: int64
```

FIGURE 8. DATASET SENTIMENT DISTRIBUTION

This imbalance can cause ML models to favor the majority class (positive), leading to biased predictions and poor performance on minority classes (neutral and negative). To address this, the **Synthetic Minority Oversampling Technique (SMOTE)** was applied to balance the training data.

SMOTE is a resampling method that generates synthetic samples for the minority classes by interpolating between existing data points. Unlike simply duplicating samples, SMOTE creates new data points, enhancing diversity in the training set and reducing overfitting (Chawla et al., 2002).

In this project, SMOTE was used to balance the training data, making sure all sentiment categories were fairly represented. SMOTE was applied only to the training set so that the test set stayed true to the original dataset, ensuring an unbiased evaluation of the model's performance.

```python
# Apply SMOTE to the training set
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)
```

FIGURE 9. CODE SNIPPET OF SMOTE IMPLEMENTATION

```
Class distribution in training set after SMOTE:
Sentiment
Positive    6629
Negative    6629
Neutral     6629
Name: count, dtype: int64
```

```
# Check the size of the original and resampled datasets
print("Original training set size:", y_train.shape[0])
print("Resampled training set size:", y_train_resampled.shape[0])
print("Number of synthetic samples added:", y_train_resampled.shape[0] - y_train.shape[0])
```

```
Original training set size: 12632
Resampled training set size: 19887
Number of synthetic samples added: 7255
```

FIGURE 10. SENTIMENT DISTRIBUTION AFTER APPLYING SMOTE

After applying SMOTE to the training set, the class distribution was perfectly balanced, this ensures equal representation of all sentiment categories.

3.4.2 Deep Learning Workflow

This workflow involved several key steps to transform the raw data into a format suitable for Long Short-Term Memory (LSTM) model.

a) Encode Sentiment Labels to Integers

To prepare the sentiment labels for the DL model, they were first converted into integer values using the **LabelEncoder** class from the **sklearn** library. This step ensures that the categorical sentiment labels ("Positive," "Neutral," "Negative") are represented in a numerical format suitable for LSTM model training and evaluation (Ghazvini et al., 2024).

The mapping of sentiment labels to integers was as follows:

- Positive: 2
- Neutral: 1
- Negative: 0

```python
from sklearn.preprocessing import LabelEncoder

# Encode sentiment labels into integers
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

# Display the count of each label
pd.Series(y_encoded).value_counts()
```

```
2    8327
1    4020
0    3444
Name: count, dtype: int64
```

FIGURE 11. CODE SNIPPET DEMONSTRATING SENTIMENT LABEL ENCODING

b) Tokenization

Tokenization is the process of converting text into numerical sequences, where each word is assigned a unique integer (Chollet, 2021). In this project, the **Tokenizer** from the **keras.preprocessing.text** module was used to prepare the text data for the LSTM model.

```python
# Tokenizes the top 10,000 words and handles out-of-vocab tokens
tokenizer = Tokenizer(num_words=10000, oov_token="<OOV>")

# Fit the tokenizer on text data
tokenizer.fit_on_texts(X)

# Convert text to sequences
X_seq = tokenizer.texts_to_sequences(X)

# View a sample of the tokenized sequences
print("Sample tokenized sequence:", X_seq[:5])  # Display the first 5 sequences
```

FIGURE 12. CODE SNIPPET FOR TOKENIZATION FOR LSTM MODEL

```
Sample tokenized sequence: [[2, 55, 29, 239, 267, 387, 457, 355, 332, 2002, 82, 305, 267, 47, 197, 15, 245, 1099, 425, 738, 2319, 615, 387,
355, 998, 447, 267, 15, 678, 616, 563, 197, 4334, 966, 82, 678, 616, 457, 355, 237, 678, 616, 76, 957, 893, 267, 15, 1065, 1671, 2410,
6755, 1, 875, 2088, 660, 15, 1020, 894, 3091, 292, 400, 76, 957, 267, 44, 220, 73, 690, 3416, 1367, 82, 4064, 732, 47, 197, 85, 825, 1100,
146, 110, 3092, 2899, 457, 355, 332, 4065, 151, 305, 36, 457, 678, 616, 109, 3185, 9595, 355, 245, 2002, 197, 82, 1128, 1079, 2815, 387,
1024, 1374, 355, 549, 267, 15, 279, 1065, 1224, 465, 447, 710, 163, 128, 1, 1901, 605, 267, 44, 245, 1901, 605, 726, 15, 211, 231, 203,
1273, 292, 373, 76, 957, 28, 32, 305, 732, 47, 286, 1746, 197, 9596, 1, 895, 267, 387, 457, 355, 3093, 4976, 998, 615, 267, 15, 678, 616,
133, 292, 76, 957, 245, 103, 197, 983, 28, 2411, 1714, 3537, 57, 7843, 47, 197, 128, 173, 520, 173, 1128, 678, 616, 601, 292, 400, 76, 957,
443, 267, 44, 1374, 267, 15, 9597, 9598, 605, 726, 197, 3538, 1, 47, 197, 133, 1150, 615, 520, 457, 355, 202, 1480, 364, 9595, 1, 236, 603,
329, 678, 616, 292, 7844, 267, 387, 457, 355, 251, 577, 305, 1902, 2002, 678, 616, 133, 292, 82, 267, 47, 197, 15, 245, 738, 197, 1150,
1099, 425, 1128, 71, 28, 2996, 1714, 3537, 57, 998, 47, 197, 128, 109, 402, 2364], [166, 116, 2089, 267, 15, 110, 80, 1099, 425, 896, 4628,
1847, 15, 166, 906, 2659, 267, 167, 114], [261, 1012, 7, 267, 387, 149, 355, 1079, 7845, 91, 217, 261, 1012, 139, 267, 387, 149, 355, 639,
577, 36, 738, 6756, 616, 2365, 447, 267, 15, 1079, 7845, 28, 3092, 4629, 461, 338, 39, 6757, 85, 387, 3291, 1368, 36, 3539, 197, 109, 3185,
3291, 4066, 2320, 379, 267, 16, 355, 3292, 3186, 2816], [87, 103, 197, 38, 9, 15, 61, 216, 623, 594, 197, 256, 104, 558, 1274, 15, 10, 290,
95, 252, 999, 6, 1286, 3, 88, 4, 1, 5, 76, 34, 4, 11, 111, 200, 6758, 901, 215, 400, 6759, 3540, 4630, 5969, 146, 15, 5, 27, 34, 497, 15,
44, 12, 11, 243, 3700, 70, 15, 1715, 2003, 1, 41, 25, 237, 1080, 23, 8, 15, 20, 29, 159, 396, 225, 92, 317, 3, 20, 138, 88, 127, 1120, 681,
18, 2, 282, 37, 126, 20, 80], [11, 264, 267, 387, 149, 355, 604, 1714, 374, 57, 998, 47, 197, 3094, 7845, 245, 1099, 425, 109, 1397, 1079,
1121, 355, 332, 1066, 76, 616, 220, 300, 292, 73, 1671, 1349, 29, 403, 727, 292, 379, 305, 471, 149, 28, 691, 357, 1555, 2366, 82, 1369,
92, 405, 77, 395, 85, 1746, 434, 18, 217, 387, 355, 103, 1434, 328, 267, 15, 99, 1065, 47, 1, 85, 138, 4631, 1901, 375, 123, 374, 1495,
197]]
```

FIGURE 13. SAMPLE OF FIRST 5 TOKENIZED SEQUENCES

c) Padding and Truncating

Padding and truncating were applied to standardize the length of all sequences for the LSTM model. Padding added zeros to shorter sequences, while truncating reduced longer sequences to a fixed maximum length. These steps ensured consistent input dimensions, which is important for efficient batch processing in deep learning (Chollet, 2021).

d) Train-Test Split

The processed data was split into training and testing sets with an 80:20 ratio.

3.5. Model Development

In this project, six models were developed to perform sentiment analysis on the preprocessed dataset. These include five ML models and one DL model. The aim was to evaluate the performance of each model and select the best one for deployment.

- 5 ML models:
  - Logistic Regression (LR)
  - Naïve Bayes (NB)
  - Random Forest (RF)
  - Linear Support Vector Classifier (LinearSVC)
  - Light Gradient Boosting Machine (LightGBM)
- DL model:
  - Long Short-Term Memory (LSTM)

The models were trained on the preprocessed dataset, and their hyperparameters were fine-tuned to improve performance. For machine learning models, methods like grid search and cross-validation were employed to find the best parameter combinations. For the LSTM model, key parameters such as the number of LSTM units, dropout rate, and learning rate were adjusted to optimize the model's architecture and performance.

```python
# Parameter grid
param_grid_lgbm = {
    "learning_rate": [0.01, 0.1],  # Controls the contribution of each tree during boosting
    "n_estimators": [50, 100],     # Number of boosting stages (trees) to train
    "max_depth": [-1, 20],         # Maximum depth of the trees (-1 means no limit)
    "num_leaves": [31, 50],        # Maximum number of leaves per tree (controls tree complexity)
    "min_data_in_leaf": [10, 20]   # Minimum number of samples required in a leaf node
}

# Initialize Model
lgbm_model = LGBMClassifier(
    random_state=42,
    force_row_wise=True,    # Optimizes memory usage by processing data row-wise
    min_split_gain=0.001    # Minimum gain required to split a node
    )

# Define callbacks to enhance training
callbacks = [
    early_stopping(stopping_rounds=10),  # Stops training if no improvement for 10 rounds
    log_evaluation(period=10)            # Logs training progress every 10 iterations
]

# Initialize GridSearchCV for hyperparameter tuning
grid_search_lgbm = GridSearchCV(
    estimator=lgbm_model,
    param_grid=param_grid_lgbm,
    cv=cross_validation,
    scoring=f1_weighted_scorer,         # Weighted F1-score as the evaluation metric for hyperparameter search
    verbose=True,
    n_jobs=-1
)

# Fit the GribSearchCV on training data
grid_search_lgbm.fit(
    X_train_resampled,
    y_train_resampled,
    eval_set=[(X_test, y_test)],                # Validation set for monitoring during training
    eval_metric=custom_f1_metric_lightgbm,      # Custom F1-score metric for validation
    callbacks= callbacks)                       # Callbacks for early stopping and logging
```

FIGURE 14. SAMPLE OF CODE USED TO BUILD LIGHTGBM MODEL

```python
# Define sequence length
sequence_length = 64  # Example sequence length
model = Sequential([

    # Input Layer to specify the input shape as a sequence of tokens
    Input(shape=(sequence_length,)),  # Explicitly define the input shape

    # Embedding Layer to convert words into dense vectors
    Embedding(input_dim=10000, output_dim=64),
    Dropout(0.3), # Prevents overfitting by randomly dropping 30% of connections during training.

    # LSTM Layer for sequential data processing: Processes sequential data and learns dependencies in the sequence.
    LSTM(units=64, return_sequences=False),  # return_sequences=False because we don't need outputs at every step
    Dropout(0.3),

    # Dense Output Layer with softmax for multi-class classification: Produces probabilities for 3 sentiment classes
    Dense(units=3, activation='softmax')  # 3 sentiment classes: Negative, Neutral, Positive
])
```

FIGURE 15. SAMPLE OF CODE USED TO BUILD LSTM MODEL

3.6. Model Evaluation

Accuracy and **F1-score** were chosen as the primary metrics to evaluate overall performance and the balance between precision and recall across sentiment classes.

3.7. Model Deployment

Model deployment is the final step in applying sentiment analysis to real-world scenarios. This phase focuses on making the best-performing model accessible for practical use and enable real-time predictions on new customer reviews.

By developing an API and a user-friendly web interface, the deployed system allows businesses to gain immediate insights from customer feedback, supporting data-driven decision-making.

- **Selecting the best model:** The model with the highest accuracy and F1-score was chosen.
- **Developing the API:** A Flask API was built to make the model accessible. Users can input new reviews and instantly receive sentiment predictions. Flask was selected for its simplicity and ease of use.
- **Building a Web Interface:** A simple and interactive web interface was created to connect with the API. Users can enter review text, and the interface will display the sentiment prediction (positive, neutral, or negative).

Flask is a lightweight and flexible Python web framework that is widely used for building APIs and web applications. It is known for its simplicity and is an excellent choice for deploying machine learning models in production environments (Grinberg, 2018).

```
from flask import Flask, request, jsonify, send_file, render_template

app = Flask(__name__)

@app.route("/test", methods=["GET"])
def test():
    app.logger.info("GET request received on /test")
    return jsonify({"message": "Flask is working!"}), 200

# Route for rendering the frontend
@app.route("/")
def home():
    return render_template("index.html")  # Render the index.html file from the templates directory


@app.route("/predict", methods=["POST"])
def predict():

    # Select the predictor to be loaded from Models folder
    trained_model = joblib.load(r"models/lightgbm_model.pkl")
```

FIGURE 16. SAMPLE OF CODE USED TO BUILD API USING FLASK

4. Evaluation and Results

This section presents the evaluation of six developed models and their performance on the test dataset. The results of various metrics, including accuracy and F1-score, are analyzed to determine the effectiveness of each model in sentiment classification. The goal is to identify the model that provides the most reliable predictions across the sentiment categories of positive, neutral, and negative.

The table below summarizes the accuracy and F1-score for each model:

Model	Accuracy	F1-Score
LightGBM	0.79	0.79
Naive Bayes	0.77	0.78
Logistic Regression	0.75	0.75
LSTM	0.75	0.75
Random Forest	0.74	0.74
LinearSVC	0.74	0.74

TABLE 1. SUMMARY OF ACCURACY AND F1-SCORE FOR SIX MODELS

4.1. Key Observations

Accuracy and **F1-score** were chosen as the primary metrics to evaluate overall performance and the balance between precision and recall across sentiment classes.

- **Best Model: LightGBM**
  - LightGBM achieved the highest accuracy and F1-score (0.79), thid makes it the best model for sentiment analysis.
- **Naïve Bayes Performed Well**
  - Naïve Bayes also performed strongly, with an F1-score of 0.78. Despite being a simple model, it handled the TF-IDF features effectively and worked well with the high-dimensional data.
- **Similar Results for Logistic Regression and LSTM**
  - Both Logistic Regression and LSTM achieved the same accuracy and F1-score (0.75).
  - Logistic Regression performed well because the dataset was straightforward and structured.
  - On the other hand, LSTM, while designed to capture sequential patterns and context, didn’t perform as well because the dataset was relatively small. Deep learning models like LSTM typically require large amounts of data to

fully capture complex relationships and show their advantages, which was not the case here.

- **Lower Scores for Random Forest and LinearSVC**
  - Random Forest and LinearSVC had the lowest scores, with both accuracy and F1-score at 0.74.
  - Random Forest struggled with the high-dimensional data, while LinearSVC might have been affected by noise in the dataset, this makes them less effective compared to other models.
- **Machine Learning vs Deep Learning Models**
  - Traditional ML models, especially LightGBM and Naïve Bayes, outperformed the deep learning model (LSTM) in this project. This shows that for smaller datasets with well-prepared features like TF-IDF, simpler models can perform better than complex ones like LSTM, which require larger datasets to fully utilize their capabilities.

In summary, LightGBM showed the best performance, with the highest accuracy and F1-score, making it the most effective model for this project. Based on these results, LightGBM was chosen as the final model. The next subsection covers the final hyperparameter tuning to further enhance its performance.

4.2. Final Hyperparameter Tuning of LightGBM

After performing final hyperparameter tuning, the performance of the LightGBM model improved across several metrics, as shown in the table below.

Metric	Before Tuning	After Tuning
Negative F1-Score	0.74	0.75
Neutral F1-Score	0.7	0.71
Positive F1-Score	0.86	0.87
Accuracy	0.79	0.8
Macro Avg F1-Score	0.77	0.77
Weighted Avg F1-Score	0.79	0.8

TABLE 2. PERFORMANCE METRICS AFTER FINAL HYPERPARAMETER TUNING OF LIGHTGBM

The tuning process optimized the model to produce more accurate and balanced predictions across all sentiment categories.

Key Improvements:

- Negative F1-Score: Increased from 0.74 to 0.75, showing better classification of negative sentiments.
- Neutral F1-Score: Improved slightly from 0.70 to 0.71, showing that the adjustments helped the model better distinguish neutral reviews from positive and negative ones.
- Positive F1-Score: Increased from 0.86 to 0.87, maintaining strong performance in identifying positive reviews.
- Accuracy: Improved from 0.79 to 0.80, reflecting a better overall prediction capability.
- Weighted Avg F1-Score: Improved from 0.79 to 0.80, indicating improved performance while accounting for class imbalance.

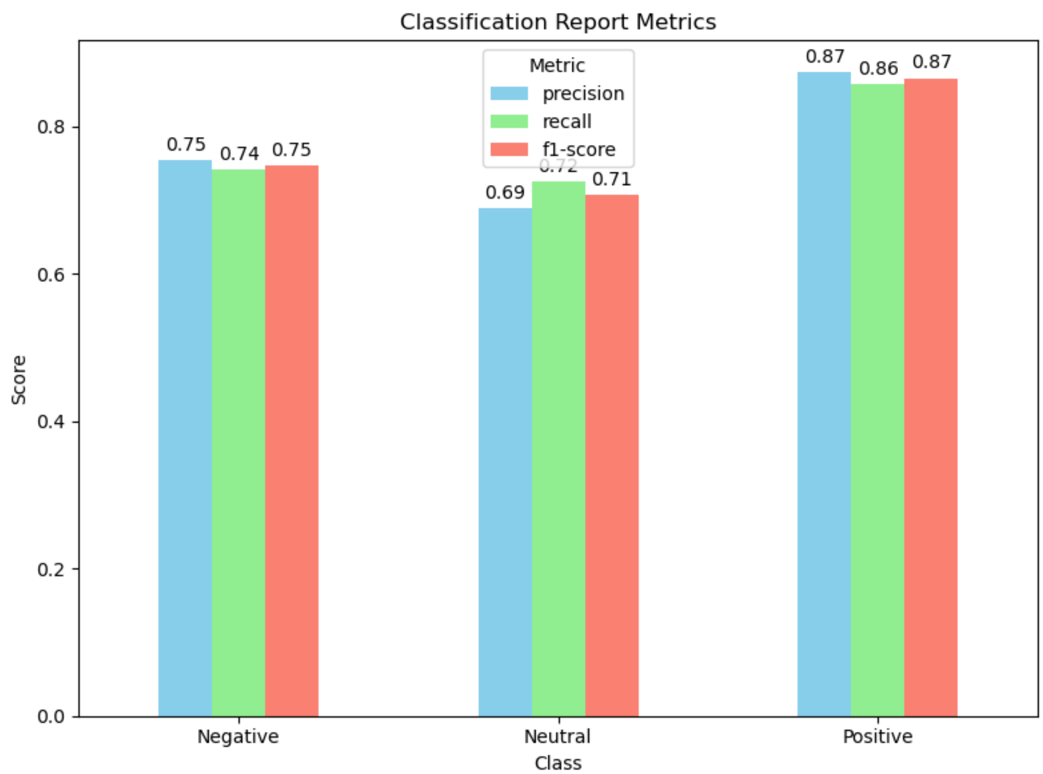

FIGURE 17. CLASSIFICATION REPORT METRICS OF FINAL TUNED LIGHTGBM MODEL

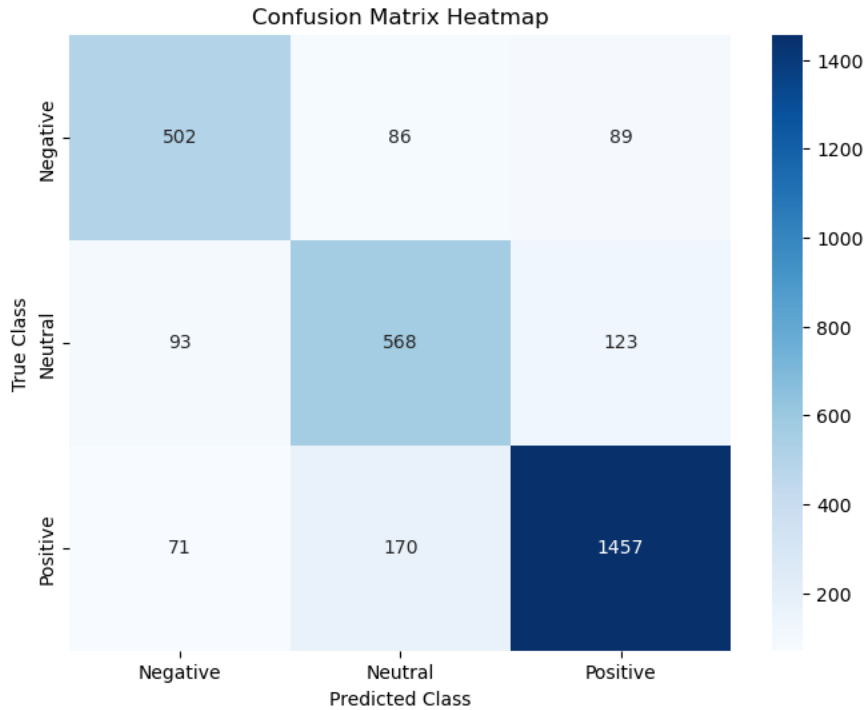

FIGURE 18. CONFUSION HEATMAP OF TUNED LIGHTGBM MODEL PERFORMANCE ON TEST DATA

The final hyperparameter tuning resulted in meaningful improvements, particularly in handling the neutral and negative sentiment categories. These refinements enhanced LightGBM's reliability and suitability for deployment, confirming its position as the best model for this sentiment analysis task.

4.3. Model Validation on New Reviews

To evaluate the generalization capability of the final LightGBM model, it was tested on 10 new reviews that were not part of the original dataset. This step was conducted to simulate real-world conditions and determine how well the model performs when analyzing fresh data.

By validating the model on new reviews, the aim is to assess its accuracy and reliability in classifying sentiments beyond the original dataset.

	Review	Prediction	Negative proba	Neutral proba	Positive proba
0	This product is perfect!	Positive	0.0750	0.0670	0.8580
1	I don't recommend this product, it doesn't work.	Negative	0.6954	0.2637	0.0410
2	The laptop is okay, nothing special.	Neutral	0.0878	0.8020	0.1102
3	I love my blink cameras and it's handy having our subscription plan setup through Amazon to make it easy to purchase the plan!	Positive	0.0018	0.0076	0.9906
4	Terrible experience with this device.	Negative	0.8524	0.1057	0.0419
5	The product is ok. Camera can be better. Little heavy.	Neutral	0.0646	0.8570	0.0784
6	I love it! Best purchase ever. I like it because I can see all around my front yard.	Positive	0.0084	0.0125	0.9791
7	Product is ok ok only. Main cons is volume is low for calls . Especially in MI phones. So consider this	Negative	0.6469	0.1717	0.1814
8	Breakdown after 3 weeks. Don't buy, really lousy customer service, no refund or gift code!!!	Negative	0.9044	0.0272	0.0684
9	Pc that supports games like League Of Legends with the RTX 3080 ti GE and the ryzen 9 3900X.	Positive	0.1248	0.3833	0.4919

FIGURE 19. MODEL PREDICTIONS ON NEW REVIEWS

Predicted class	Negative	Neutral	Positive
Real class			
Negative	3	0	0
Neutral	1	2	0
Positive	0	0	4

FIGURE 20. CONFUSION MATRIX FOR FINAL MODEL PREDICTIONS ON NEW REVIEWS

Insights from Validation:

- Strengths
  - The model accurately classified most reviews (90%), particularly those with clear sentiment, such as strongly positive or negative language.
  - Demonstrates strong confidence (class proba) and class separation for most reviews
- Weaknesses
  - Struggles with ambiguous or mixed sentiments, as shown by the misclassification of review #7
  - Slight overlap between neutral and negative classes

In conclusion, the LightGBM model achieved 90% accuracy on the validation set, effectively classifying most reviews, especially those with clear positive or negative sentiments. While it struggled with mixed sentiments and showed some overlap between neutral and negative classes, the overall results confirm its reliability for real-world sentiment analysis.

However, further fine-tuning in future work is needed to improve predictions for the neutral class and reduce overlap with other classes. This will help the model better handle subtle sentiment differences and enhance overall performance.

5. Web Application Performance

This section evaluates the performance and usability of the web application developed to deploy the final LightGBM model. The web application provides a user-friendly interface for real-time sentiment analysis and enable users to input new reviews and receive sentiment predictions.

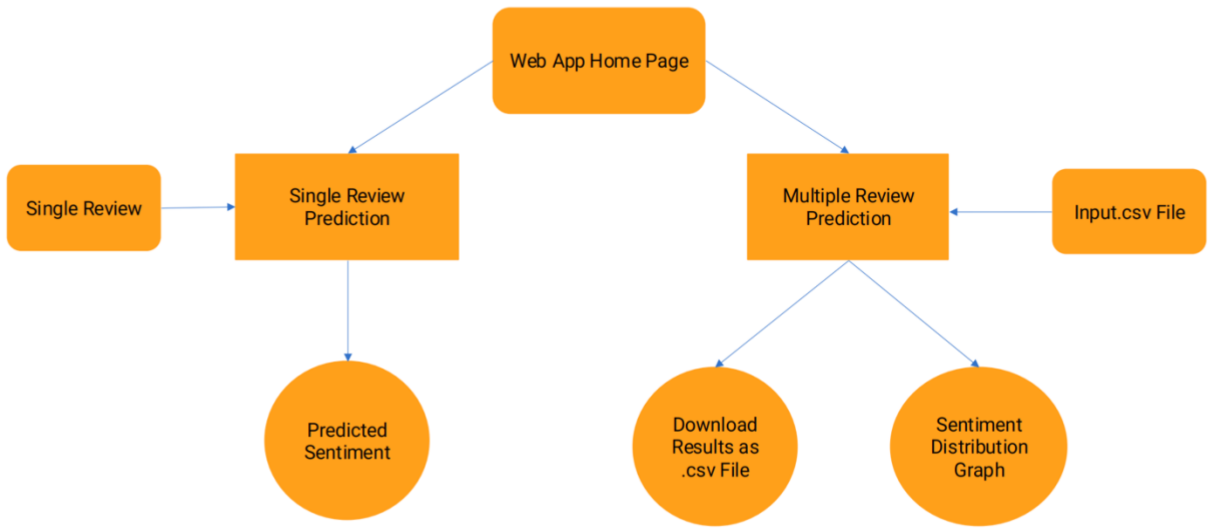

FIGURE 21. WEB INTERFACE WORKFLOW FOR SENTIMENT PREDICTION

The web application provides two main functionalities for sentiment analysis:

- **Single Review Prediction:**

  Users can input a single review on the home page, and the app will display the predicted sentiment (positive, neutral, or negative).

- **Multiple Review Predictions:**

  Users can upload a **.csv** file containing multiple reviews. The app processes the reviews and provides two outputs:

  - Downloadable Results: A **.csv** file containing the predicted sentiments for all reviews.
  - Sentiment Distribution Graph: A pie chart representation of the sentiment distribution across the uploaded reviews.

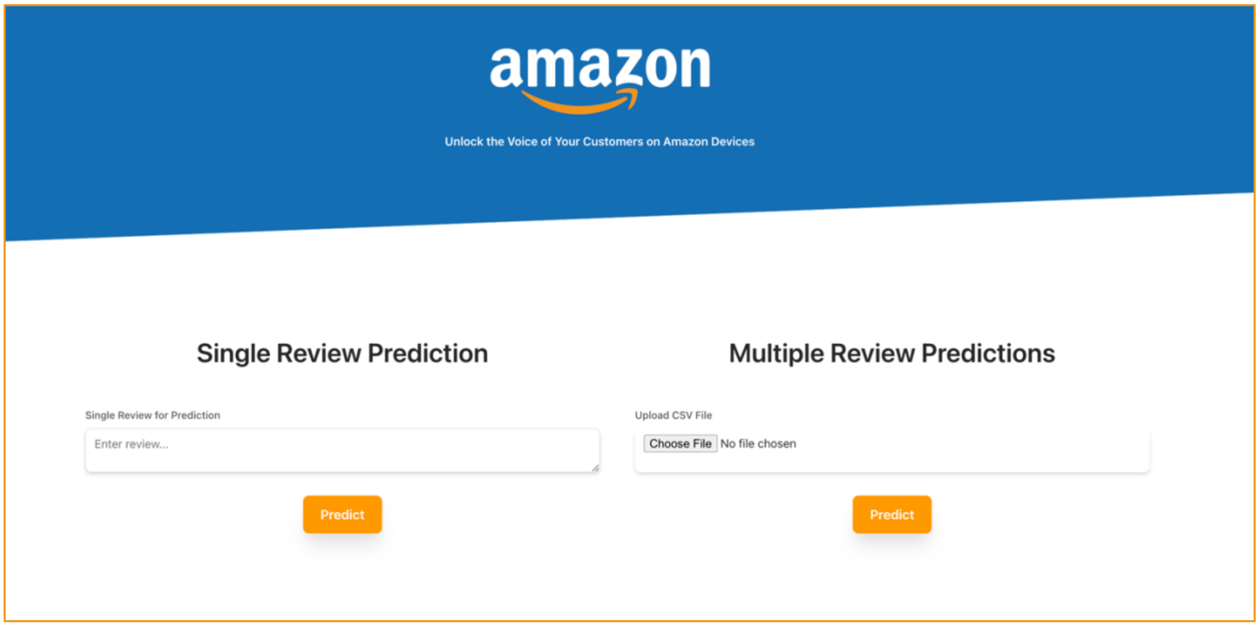

FIGURE 22. WEB APPLICATION HOME PAGE

5.1. Single Review Prediction

The single review prediction feature allows users to input a review and receive its sentiment classification (positive, neutral, or negative) after clicking the **Predict** button.

How It Works:

- **Input:** Users type a review into the text box on the web app interface (Step 1)
- **Output:** After clicking the **Predict** button (Step 2), the predicted sentiment is displayed in the **Prediction Result** text box.

This feature provides a quick and user-friendly way to analyze the sentiment of individual reviews in real time.

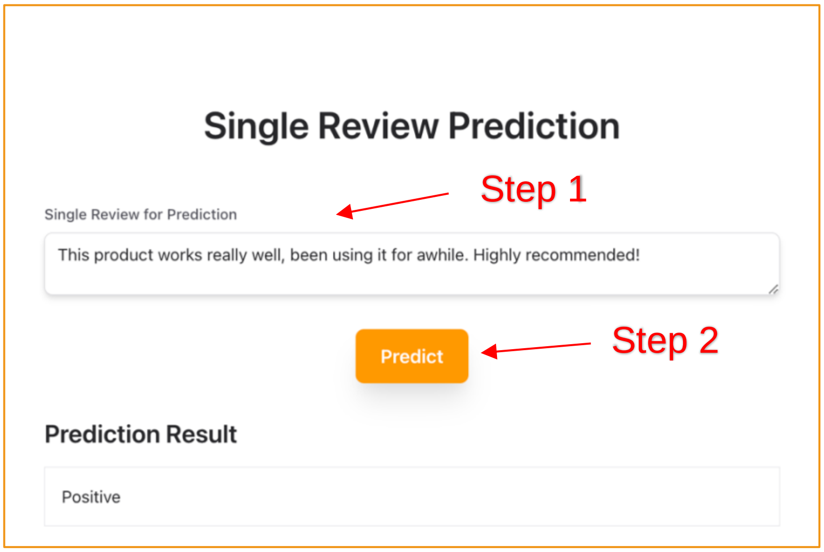

FIGURE 23. SINGLE PREDICTION FOR POSITIVE REVIEW

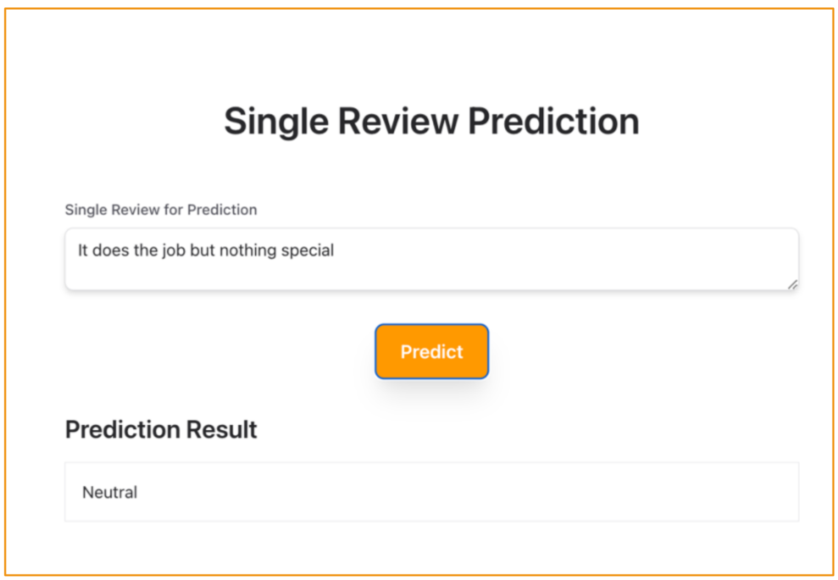

FIGURE 24. SINGLE PREDICTION FOR NEUTRAL REVIEW

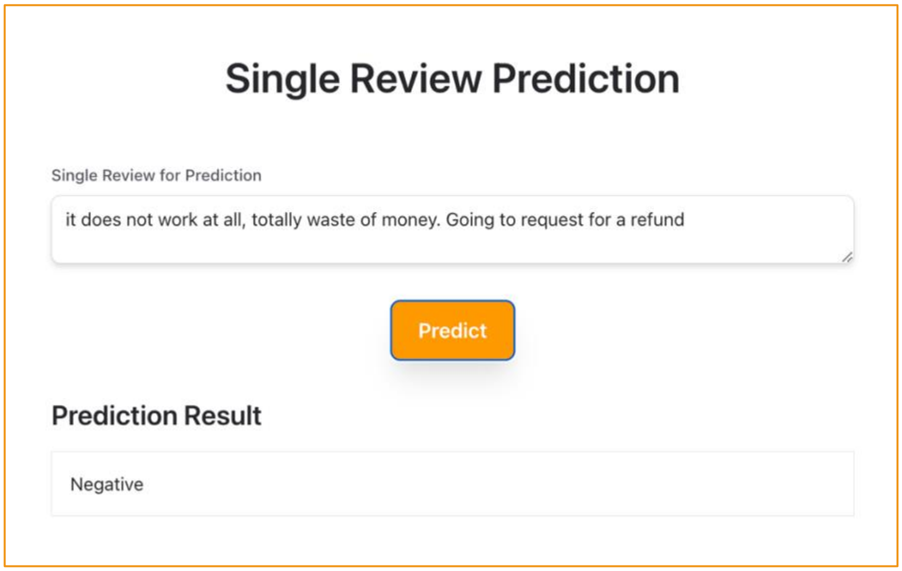

Figure 25. Single prediction for negative review

5.2. Multiple Review Predictions

The multiple review prediction feature enables users to analyze the sentiment of several reviews at once by uploading a **.csv** file containing the reviews.

How it works:

- **Input:** Users upload a **.csv** file with a column named **"Review"** containing the text reviews.
- **Output:**
  - **Downloadable Results:** A .csv file containing the original reviews along with their predicted sentiments.
  - **Sentiment Distribution Graph:** A pie chart showing the proportion of positive, neutral, and negative reviews.

This feature is ideal for bulk sentiment analysis, it offers quick and efficient processing for large datasets, while providing both detailed results and visual insights into sentiment trends.

The image below shows an example of an input CSV file with reviews are listed in the 'Review' column.

Review
Highly recommend it to anyone looking for something reliable and well-made.
The quality of this item is absolutely awful.
It's okay, but I probably wouldn't buy it again.
Fantastic! I loved every aspect of this.
Terrible experience. I want a refund.
Not bad, but not great either.
Customer service was unresponsive and unhelpful. Very disappointing experience.
I can't believe how poorly this was made.
Decent product for the price.
This is the best purchase I've ever made!

FIGURE 26. EXAMPLE OF AN INPUT .CSV FILE

The following image displays the web app interface for the multiple review prediction feature. Users can upload a CSV file containing reviews by clicking the **Choose File** button and then select **Predict** to analyze the sentiments. Once processed, the sentiment distribution is visually represented in a pie chart, showing the percentage of positive, neutral, and negative reviews. Additionally, users can download the sentiment prediction results as a CSV file by clicking the **Download Results** button.

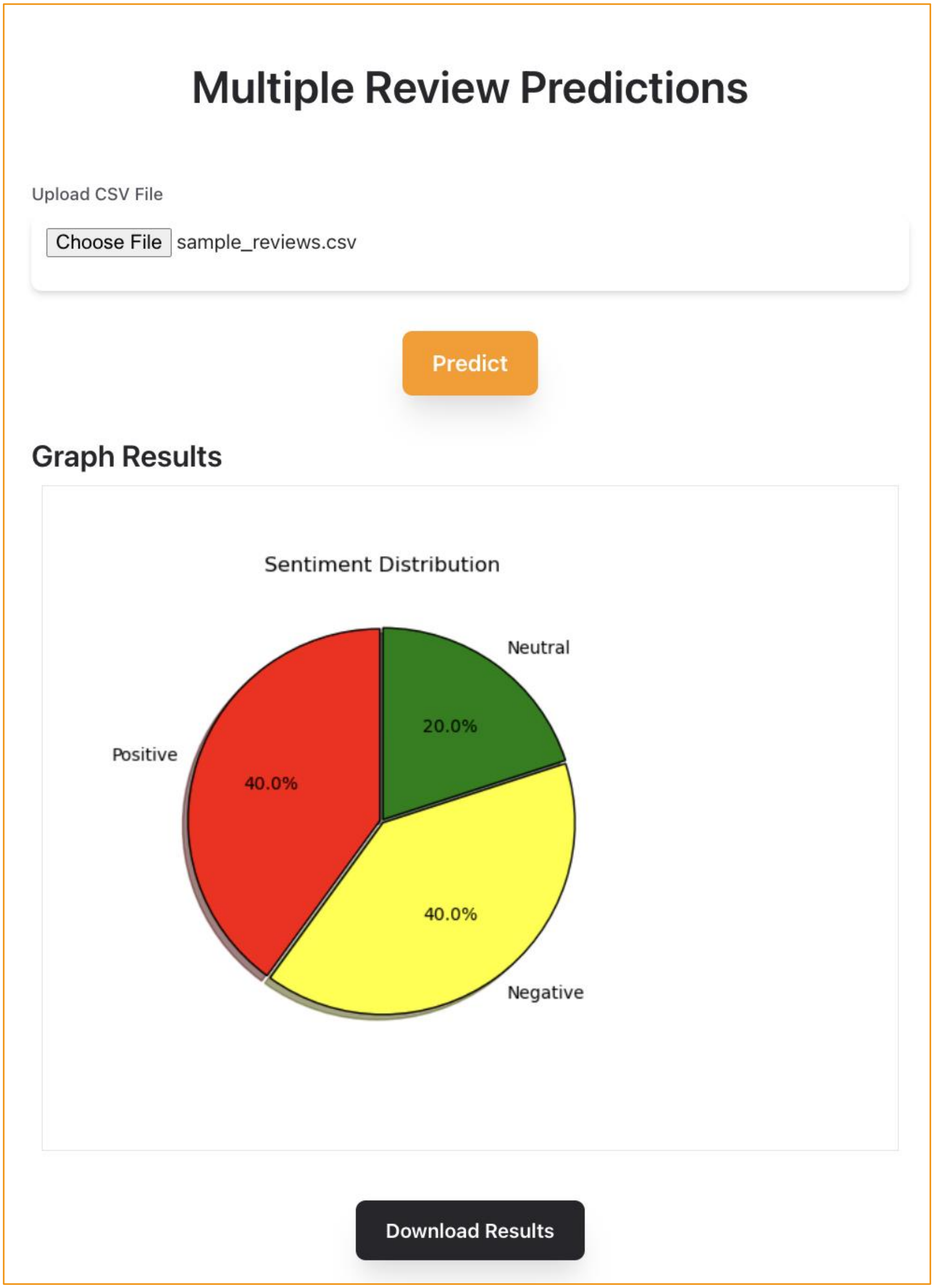

FIGURE 27. MULTIPLE REVIEW PREDICTIONS INTERFACE

Review	Prediction
Highly recommend it to anyone looking for something reliable and well-made.	Positive
The quality of this item is absolutely awful.	Negative
It's okay, but I probably wouldn't buy it again.	Neutral
Fantastic! I loved every aspect of this.	Positive
Terrible experience. I want a refund.	Negative
Not bad, but not great either.	Neutral
Customer service was unresponsive and unhelpful. Very disappointing experience.	Negative
I can't believe how poorly this was made.	Negative
Decent product for the price.	Positive
This is the best purchase I've ever made!	Positive

FIGURE 28. EXAMPLE OF THE DOWNLOADED PREDICTION FILE

The image displays a sample of the downloaded CSV file generated from the multiple review prediction feature. It contains two columns: the Review column, which includes the original text of the reviews, and the Prediction column, showing the predicted sentiment for each review (Positive, Neutral, or Negative).

Based on the downloaded results and the pie chart, the model accurately classifies reviews into positive, neutral, or negative sentiments. This function provides both a clear textual output and visual representation, making it easy for users to interpret overall sentiment trends in the dataset.

6. Data Visualization Dashboard

A Power BI dashboard was developed to provide a comprehensive overview of the sentiment analysis results. This dashboard offers interactive and visually appealing insights to help users better understand review data and uncover patterns.

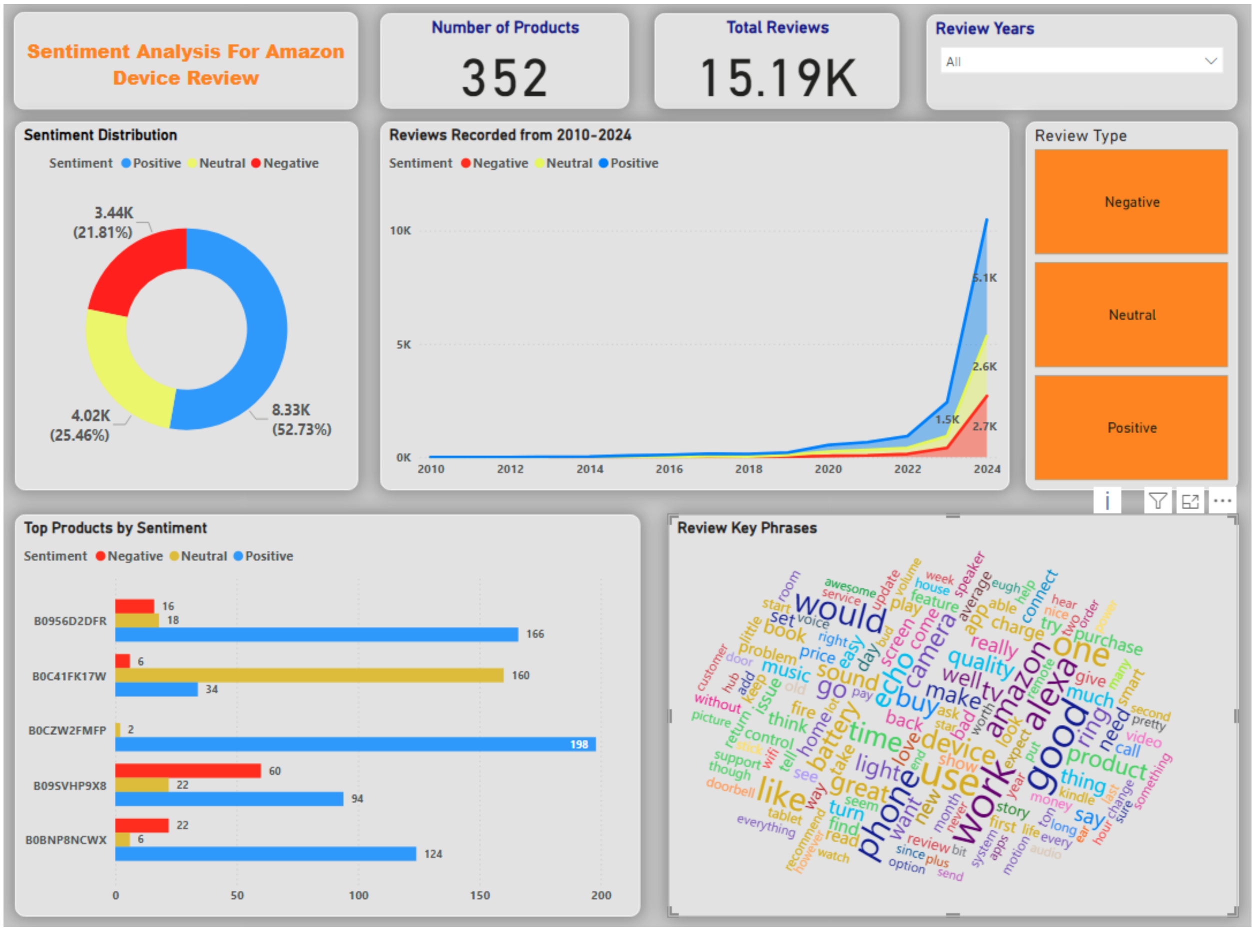

FIGURE 29. SENTIMENT ANALYSIS POWER BI DASHBOARD FOR AMAZON DEVICE REVIEW

This dashboard delivers actionable insights by combining interactive visuals with statistical analysis. It serves as a valuable tool for understanding customer feedback, identifying strengths and weaknesses in products and supporting data-driven decision-making. Below is a breakdown of the key visualizations included in the dashboard:

6.1. Sentiment Distribution

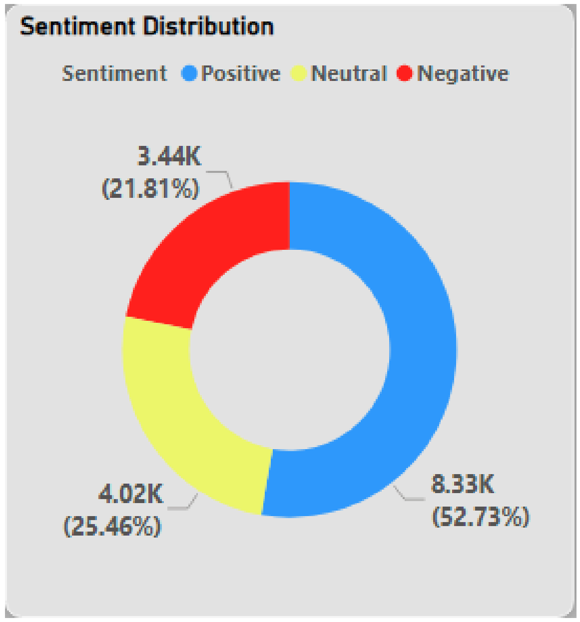

FIGURE 30. SENTIMENT DISTRIBUTION

This pie chart shows the proportion of positive, neutral, and negative sentiments across all reviews. It provides a high-level overview of customer sentiment and reveals that **52.73%** of reviews are positive, while **25.46%** are neutral and **21.81%** are negative. This helps to assess the overall customer satisfaction with Amazon devices.

6.2. Review Trends Over Time

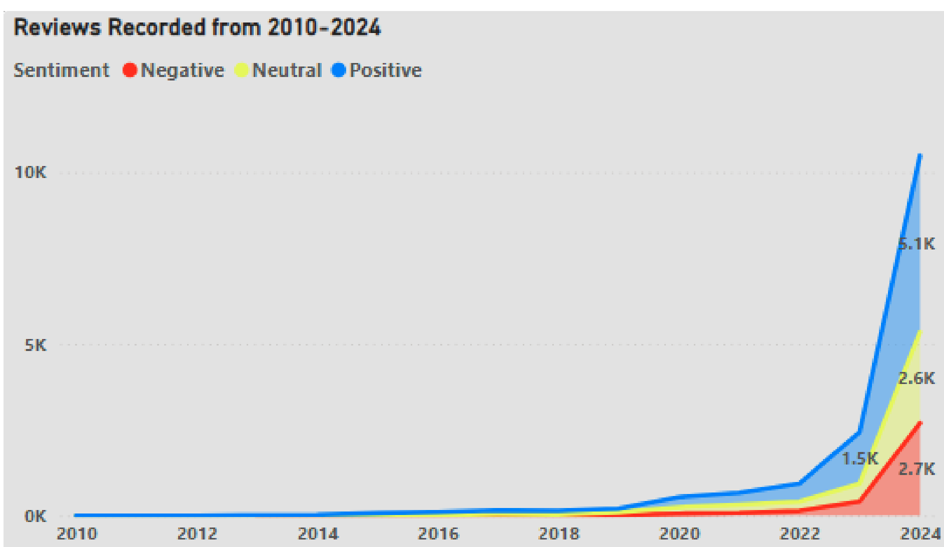

FIGURE 31. REVIEW TRENDS OVER TIME

The line chart visualizes the number of reviews recorded from **2010 to 2024**, categorized by sentiment (Positive, Neutral, Negative).

The review count showed a gradual increase between **2010 and 2015**, with Positive reviews consistently leading.

From **2016 to 2020**, the reviews grew steadily, with significant spikes in 2020, reaching **290 Positive**, **190 Neutral** and **58 Negative** reviews.

A sharp surge occurred between **2021 and 2024**, culminating in 2024 with **5,140 Positive**, **2,648 Neutral**, and **2,702 Negative** reviews.

This visualization highlights customer engagement trends, identifying periods of high interaction and potential areas for improvement.

6.3. Top Products by Sentiment

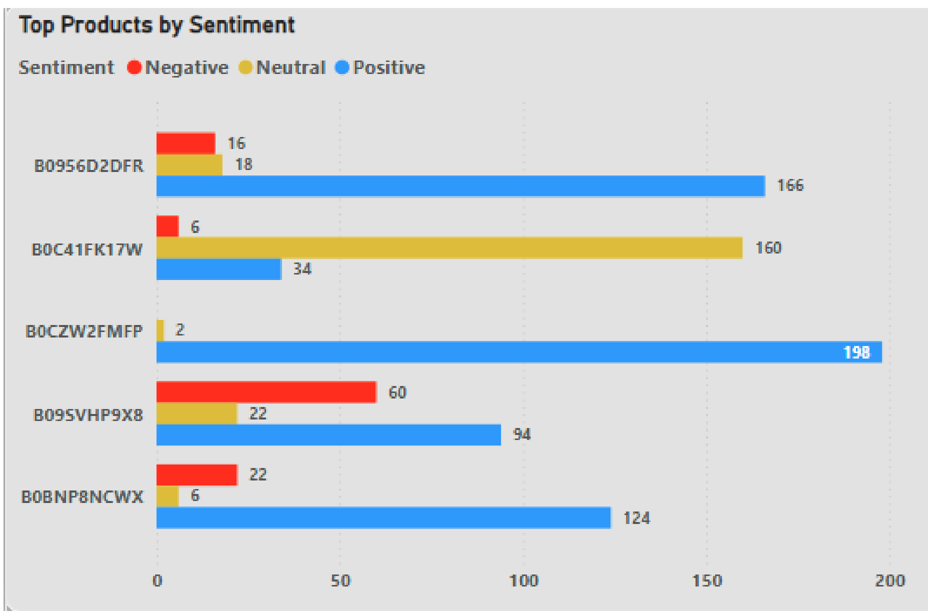

FIGURE 32. TOP PRODUCTS BY SENTIMENT

This bar chart ranks the top 5 products based on their sentiment classifications and provides a clear comparison of positive, neutral, and negative feedback for each product. For example, **product BOCZW2FMFP** stands out with the highest number of positive reviews (**198**), this shows strong customer satisfaction. In contrast, **B0956D2DFR** has a more balanced distribution, with **166 positive**, **18 neutral** and **16 negative reviews**. This indicates mixed customer experiences. Similarly, **B0C41FK17W** also shows a mix of sentiments, with **160 positive**, **34 neutral** and **6 negative reviews**.

This visualization provides useful insights into how customers view different products. It helps businesses easily spot top-performing products with mostly positive reviews and identify those needing improvement due to more neutral or negative feedback. These insights can support product improvements, marketing plans, and better customer service to meet customer needs.

6.4. Review Key Phrases

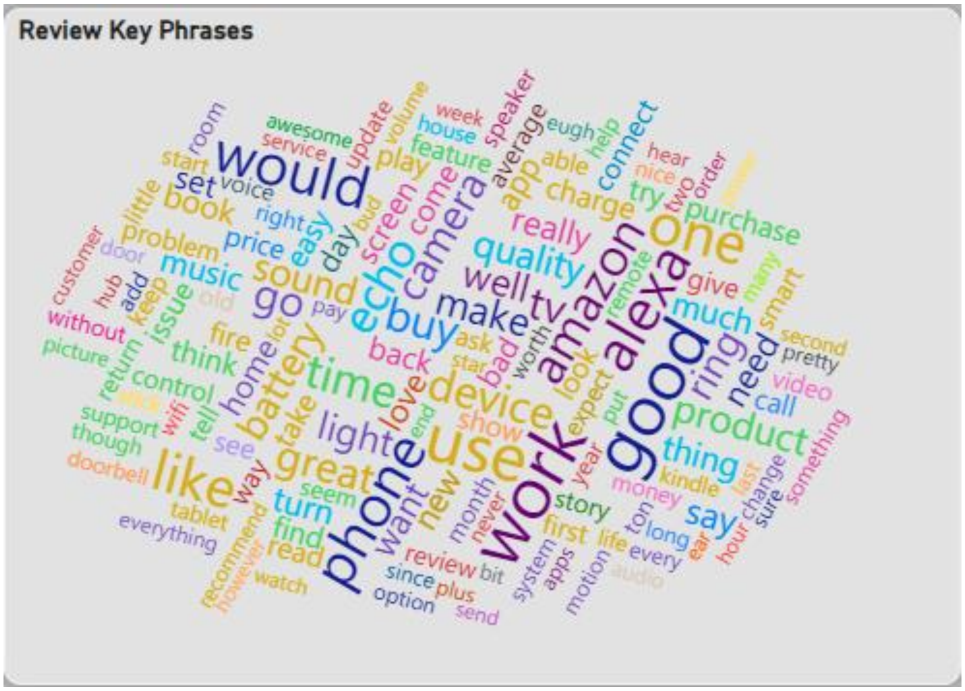

FIGURE 33. REVIEW KEY PHRASES ACROSS SENTIMENTS

A word cloud visual displays the most frequently mentioned key phrases from the positive, negative and neutral reviews. This visualization provides a quick and intuitive understanding of what drives customer sentiment.

6.4.1 Positive Reviews

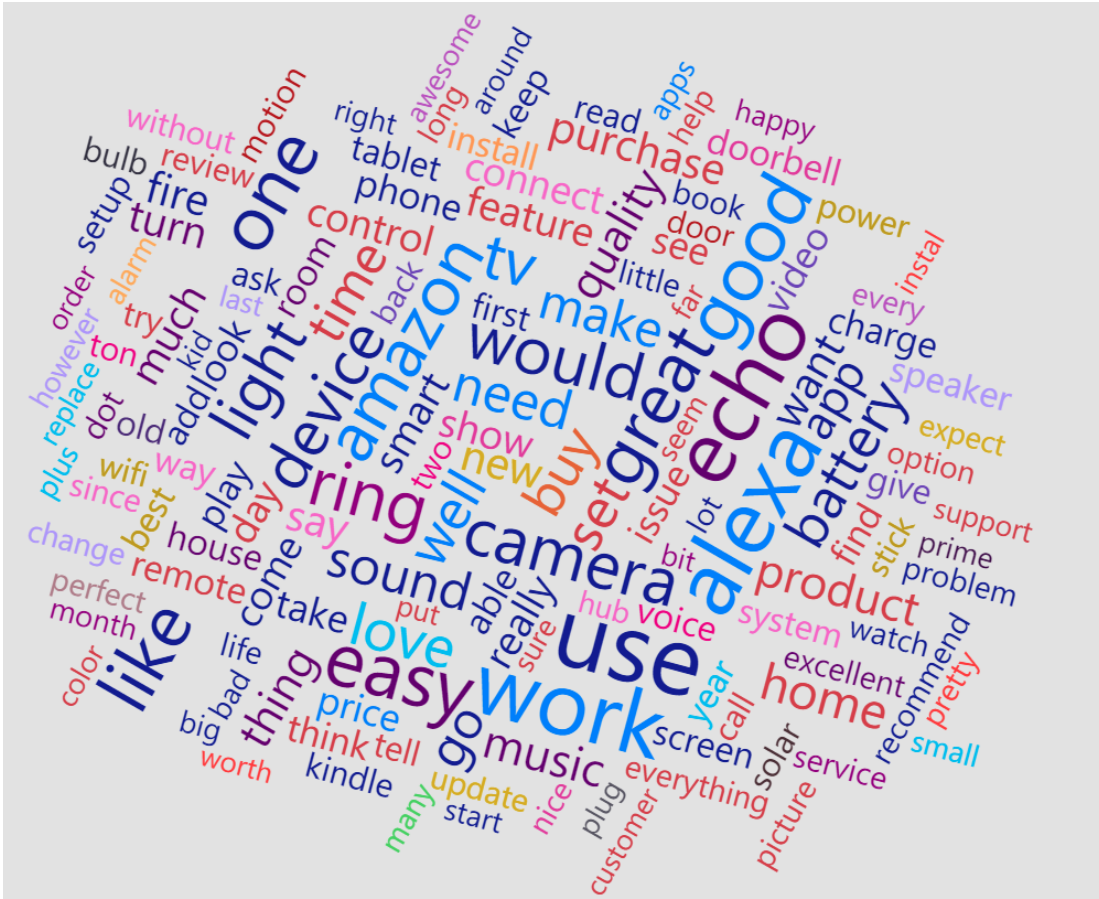

FIGURE 34. REVIEW KEY PHRASES FOR POSITIVE SENTIMENT

Phrases such as **"good", "great" and "love"** reveal a strong positive sentiment among customers, indicating general satisfaction with Amazon devices.

The frequent mention of usability-related terms like **"easy", "use"** and **"work"** suggests that users highly value the simplicity and functionality of the products.

Specific product names such as **"alexa", "echo"** and **"amazon"** dominate the word cloud, this highlights the popularity of Alexa-enabled devices.

Additionally, entertainment-related terms like **"music"** and **"TV"** also appear frequently, suggesting that media and streaming functionalities are significant contributors to customer satisfaction.

6.4.2 Negative Reviews

FIGURE 35. REVIEW KEY PHRASES FOR NEGATIVE SENTIMENT

The word cloud for negative reviews highlights several recurring issues. The dominant presence of **"phone", "work"** and **"battery"** suggests that many negative reviews revolve around device performance and power-related issues. Words like **"problem", "bad"** and **"disappointed"** indicate customer frustration and dissatisfaction with product quality or expectations not being met.

Phrases such as **"return", "replacement"** and **"money"** suggest that customers faced defective products or were dissatisfied enough to seek refunds or exchanges. Additionally, terms like **"charge", "screen"** and **"connect"** point to specific technical or hardware-related issues, while **"support"** and **"service"** may imply dissatisfaction with customer service experiences. Words like **"slow", "waste"** and **"poor"** further emphasize performance concerns.

6.4.3 Neutral Reviews

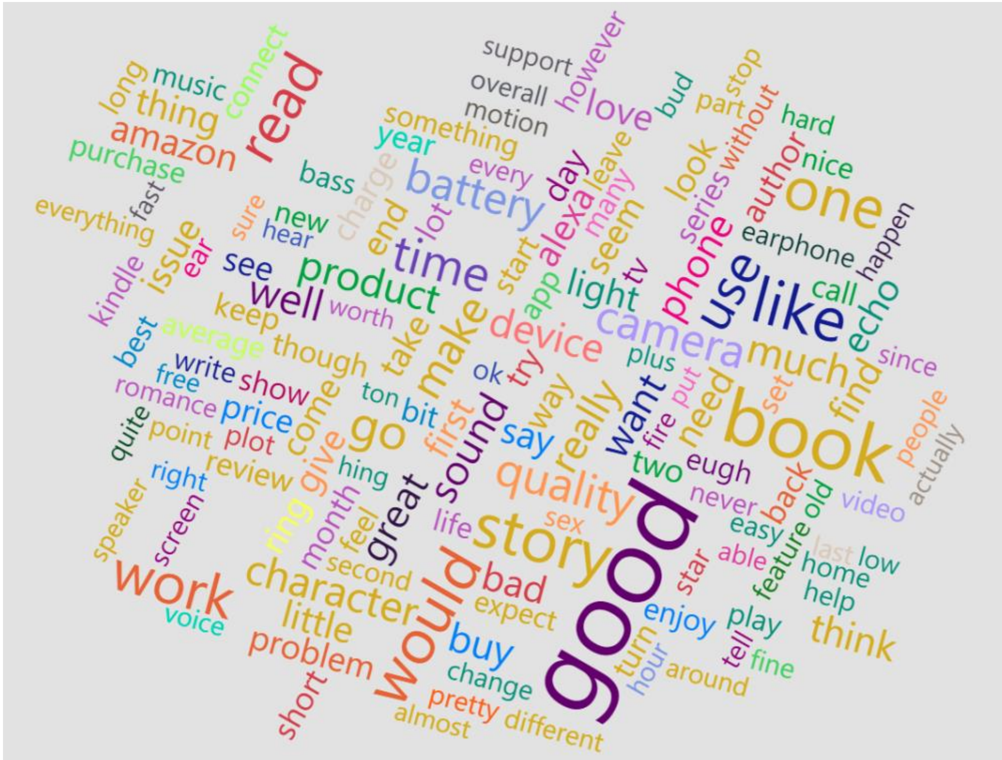

FIGURE 36. SENTIMENT KEY PHRASES FOR NEUTRAL SENTIMENT

The neutral reviews word cloud highlights more balanced customer experiences. Words like **"good", "quality"** and **"product"** suggest that customers acknowledge acceptable performance and quality without expressing significant enthusiasm or dissatisfaction. Terms like **"work", "use"** and **"like"** show a practical and matter-of-fact tone, indicating that these devices met basic expectations but did not necessarily exceed them. Additionally, phrases such as **"price", "average"** and **"worth"** reflect a neutral evaluation of the product's value for money.

6.5. Summary Cards

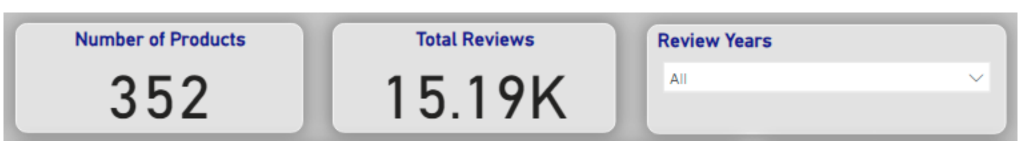

FIGURE 37. SUMMARY CARDS

At the top of the dashboard, summary cards provide quick insights:

- **Number of Products**: Displays the total number of unique products analyzed (**352**).
- **Total Reviews**: Shows the total number of reviews processed (**15.19K**).
- **Review Years**: Offers a filter to explore data across different time periods.

6.6. Sentiment Filter

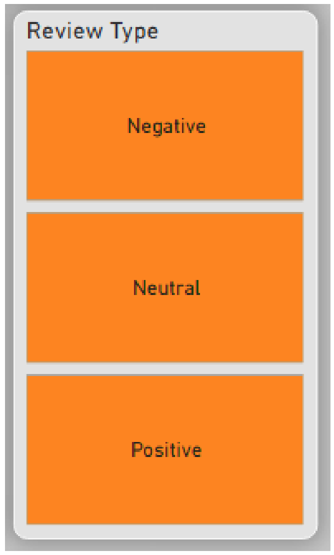

FIGURE 38. SENTIMENT FILTER

This interactive filter allows users to explore specific sentiment categories (Positive, Neutral or Negative) across the entire dashboard. It enhances the user experience by enabling focused analysis on particular sentiment.

7. Answers to Research Questions

In this section, the research questions posed earlier are answered based on the findings from the entire process, including data collection, preprocessing, model development, evaluation, and deployment. The answers provide a comprehensive view of the insights gained throughout the project.

7.1. Research question 1

What is the most effective NLP, Machine Learning, and Deep Learning techniques for analyzing sentiment in Amazon device reviews?

<u>Answer:</u>

The project identified several effective techniques across NLP, ML, and DL.

For NLP, text preprocessing steps such as tokenization, lemmatization and stopword removal were important for standardizing the data. Additionally, TF-IDF was highly effective for feature extraction, converting text into numerical features that captured the importance of words in sentiment classification.

In terms of ML, LightGBM outperformed other models due to its ability to handle imbalanced data, sparse features and large datasets efficiently. The use of SMOTE (Synthetic Minority Oversampling Technique) further improved performance by balancing sentiment classes in the training set.

For deep learning, LSTM was valuable for capturing the sequential nature of text data but required larger datasets for better performance. Overall, the combination of **text preprocessing, TF-IDF, SMOTE, and LightGBM** proved to be the most effective approach for this sentiment analysis task.

7.2. Research question 2

How do machine learning models compare to deep learning models, such as LSTM-based RNNs, in terms of sentiment classification accuracy and F1-score?

Answer:

In this project, ML models generally performed better than the DL model (LSTM) in terms of accuracy and F1-score. LightGBM particularly achieved the highest accuracy (0.80) and F1-score (0.79–0.80), this shows its ability to effectively handle high-dimensional features and imbalanced datasets. Other ML models, such as Naïve Bayes and Logistic Regression also performed well with F1-scores of 0.78 and 0.75 respectively.

The LSTM model with the capable of capturing sequential patterns in text achieved an accuracy and F1-score of 0.75 which is comparable to Logistic Regression but lower than LightGBM. This result suggests that the small dataset limited the LSTM's ability to fully utilize its sequential learning capabilities, which typically require larger datasets for optimal performance.

Overall, ML models, especially LightGBM, provided better performance and required less computational effort compared to LSTM, this result makes them more suitable for this sentiment analysis task.

7.3. Research question 3

What are the key factors driving positive and negative sentiment for Amazon devices?

Answer:

The analysis revealed several key factors that influence both positive and negative sentiment in Amazon device reviews.

Positive sentiment is largely influenced by factors such as functionality, ease of use, and performance. These aspects are reflected in frequently used words like "**work**", "**use**" and "**great**". Customers often show great satisfaction with how well the devices work together and their quality, as seen in terms like "**love**" and "**fantastic**"

On the other hand, **negative sentiment** is primarily driven by issues related to hardware reliability, battery life, and customer service. Common words like "**phone**", "**battery**" and "**charge**" highlight technical problems. Additionally, terms such as "**refund**" and "**return**" suggest dissatisfaction with service policies or product failures. These insights provide valuable guidance for improving product design and customer support.

7.4. Research question 4

How can sentiment analysis insights improve product recommendations and enhance customer satisfaction?

Answer:

Sentiment analysis provides actionable insights that can significantly improve product recommendations and enhance customer satisfaction:

- **Personalized Recommendations**: Positive reviews reveal what customers like, this allows businesses to recommend products that match their preferences. For example, customers who value ease of use can be recommended similar user-friendly devices.
- **Addressing Customer Concerns:** Negative reviews highlight common issues, such as hardware problems or poor battery life. Recommending alternative products with better features can help address these concerns and improve satisfaction.
- **Better Customer Support:** Sentiment analysis can identify areas where customer service needs improvement like refund policies or response times. Fixing these issues creates a better overall experience for customers.
- **Product Improvements:** Feedback from sentiment analysis helps businesses identify areas for improvement, this will be helpful for feature updates or the development of new products based on customer needs.

7.5. Research question 5

What challenges arise in developing sentiment analysis tools, and how can these challenges be effectively mitigated?

Answer:

Developing sentiment analysis tools involves several challenges, including data imbalance, limited data access, and difficulties in handling mixed or subtle sentiments. These challenges were addressed through various techniques to ensure accurate and reliable model performance.

- **Imbalanced Dataset**

  The dataset collected from Amazon reviews had an uneven distribution of sentiment classes, with fewer Negative and Neutral reviews compared to Positive ones.

  Mitigation: SMOTE (Synthetic Minority Oversampling Technique) was applied to balance the training data and improve the model's ability to accurately classify underrepresented sentiment classes.

- **Limited Data Access**

  Amazon restricts data scraping to a maximum of 10 pages of reviews per product, which limited the overall dataset size.

Mitigation: Reviews were collected from multiple products to expand the dataset and ensure sufficient diversity for effective model training.

- **Mixed Sentiments and Complex Language**

  Some reviews contained mixed sentiments with both pros and cons, while others used complex language like sarcasm. This makes them difficult to classify.

  Mitigation: Advanced text preprocessing techniques, including tokenization, lemmatization, and TF-IDF vectorization were used to structure the data. The LightGBM model was chosen for its ability to handle such complexities effectively.

- **High Training Time for Some Models**

  Certain models, such as LightGBM and LinearSVC required a significant amount of time to train due to their complexity.

  Mitigation: Hyperparameter tuning with **early stopping** was applied to reduce training time without compromising model performance. This technique stopped training once the model showed no further improvement after a certain number of iterations (Bengio, 2012).

8. Conclusion

This project successfully developed a sentiment analysis tool for Amazon device reviews by effectively combining Natural Language Processing (NLP), Machine Learning (ML) and Deep Learning (DL) techniques. Through the systematic process of data collection, preprocessing, model development, and deployment, the project fulfilled its objectives and provided answers to the research questions.

At the core of this project is a **user-friendly web application** that delivers real-time sentiment classification. This tool allows users to analyze individual or multiple reviews efficiently which offers a practical and reliable solution for sentiment analysis. To enhance user insights, a **comprehensive Power BI dashboard** was designed to visualize key aspects of the analysis, including sentiment distribution, product performance comparisons, review trends over time and keyword frequency analysis. The interactive dashboard enables users to explore customer feedback dynamically, uncover patterns and make data-driven decisions.

The project identified **LightGBM** as the most effective machine learning model which achieved the highest accuracy (0.80) and F1-score (0.80) among the six tested models. Its ability to handle imbalanced datasets and high-dimensional features made it the ideal choice for this task. The deep learning model **LSTM** showed potential in capturing sequential relationships in text, however its performance was limited due to the relatively small dataset.

The tool performed well in identifying clear positive and negative sentiments with accurate and actionable results. However, challenges remain in handling mixed or complex sentiments. This highlights opportunities for further improvement in both the model and the overall system.

Key sentiment drivers were identified: positive sentiments were linked to functionality, ease of use, and performance, while negative sentiments stemmed from hardware reliability, battery life, and customer service issues. These insights provide valuable guidance for improving product design and customer satisfaction.

Challenges encountered during the project including imbalanced data, limited data access, and high training times for some models were effectively mitigated through methods like SMOTE, multi-product data scraping and hyperparameter tuning with early stopping.

Overall, the project demonstrates the potential of sentiment analysis to extract meaningful insights from customer feedback and provides a reliable tool for real-world applications.

9. Future Work

Building on the findings and limitations of this project, several areas have been identified for future work to further enhance the sentiment analysis system:

- **Dataset Expansion**

  To improve model generalization and reduce data limitations, future efforts should focus on scraping additional reviews across a wider range of products. Additionally, incorporating external datasets can provide more diverse and representative data for training and evaluation.

- **Explore Advanced Embedding Methods**

  Future work could implement advanced word embedding techniques such as Word2Vec or GloVe. These pre-trained embeddings can capture semantic and syntactic relationships between words, providing richer feature representations and potentially improving model accuracy.

- **Higher-Order N-Grams**

  Experimenting with trigrams or higher-order n-grams may help capture more contextual information from reviews, particularly for complex sentiments. Dimensionality issues can be managed by applying techniques such as term frequency reduction or selecting the most relevant n-grams.

- **Improving Class Imbalance**

  To further refine model performance, future work can explore advanced methods to address class imbalance, including oversampling, class weighting, or synthetic data generation. These approaches can enhance predictions, especially for underrepresented classes and mixed sentiment reviews.

- **Focus on Neutral Class**

The neutral class presented a challenge in this project due to its overlap with both positive and negative sentiments. Future work should prioritize improving feature representation and ensuring a balanced dataset for the neutral class, ultimately reducing misclassifications and enhancing precision.

- **Analyze Misclassifications**

A detailed analysis of misclassified reviews across all sentiment classes can provide valuable insights into the model's limitations. This includes identifying patterns in ambiguous, mixed, or complex sentiments that the model struggles to classify correctly. By addressing these issues, feature extraction techniques and model training strategies can be refined to improve overall classification accuracy and reliability.

10. Ethical Considerations

Ethical considerations were a key focus throughout this project to ensure responsible data collection, privacy protection and fair model training. The following measures were implemented:

- **Data Collection**

Reviews were scraped from Amazon’s website while strictly adhering to its ethical guidelines and terms of service. This included limiting data collection to a maximum of 10 pages per product to comply with Amazon’s scraping restrictions. These measures ensured that the data collection process was both legal and ethical.

- **Data Privacy**

Only publicly available reviews were used in this project. No personally identifiable information (PII) was collected, stored or analyzed during the data scraping process. This safeguarded user privacy and aligned with ethical standards for handling sensitive data.

- **Imbalanced Dataset**

Addressing potential bias caused by an imbalanced dataset was a priority to ensure fair and accurate sentiment classification across all sentiment classes. SMOTE

(Synthetic Minority Oversampling Technique) were applied to balance the dataset and enhance the model’s ability to handle underrepresented classes.

10. References

Pang, B. and Lee, L., 2008. Opinion Mining and Sentiment Analysis. Information Retrieval, 2(1-2), pp.1-135.

Cambria, E., Schuller, B., Xia, Y. and Havasi, C., 2013. New avenues in opinion mining and sentiment analysis. IEEE Intelligent systems, 28(2), pp.15-21.

Medhat, W., Hassan, A. and Korashy, H., 2014. Sentiment analysis algorithms and applications: A survey. Ain Shams engineering journal, 5(4), pp.1093-1113.

Ng, A. and Jordan, M., 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. Advances in neural information processing systems, 14.

Sebastiani, F., 2002. Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1), pp.1-47.

Zhang, H., 2004. The optimality of naive Bayes. Aa, 1(2), p.3.

Breiman, L., 2001. Random forests. Machine learning, 45, pp.5-32.

Joachims, T., 1998, April. Text categorization with support vector machines: Learning with many relevant features. In European conference on machine learning (pp. 137-142). Berlin, Heidelberg: Springer Berlin Heidelberg.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.Y., 2017. Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30.

Dube, L. and Verster, T., 2023. Enhancing classification performance in imbalanced datasets: A comparative analysis of machine learning models. Data Science in Finance and Economics, 3(4), pp.354-379.

Alzamzami, F., Hoda, M. and El Saddik, A., 2020. Light gradient boosting machine for general sentiment classification on short texts: a comparative evaluation. IEEE access, 8, pp.101840-101858.

Naji, L.A.W.A., Eltahir, I.K. and Elsheikh, H.H.A., 2024. Integrating LSTM, Transformer, and LightGBM for Enhanced Predictive Modeling: A Mechanistic Approach. Journal of Intelligent Systems and Applied Data Science, 2(1).

Young, T., Hazarika, D., Poria, S. and Cambria, E., 2018. Recent trends in deep learning based natural language processing. ieee Computational intelligenCe magazine, 13(3), pp.55-75.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y. and Potts, C., 2013, October. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing (pp. 1631-1642).

Schmidhuber, J. and Hochreiter, S., 1997. Long short-term memory. Neural Comput, 9(8), pp.1735-1780.

Bengio, Y., Goodfellow, I. and Courville, A., 2017. Deep learning (Vol. 1). Cambridge, MA, USA: MIT press.

Zhang, C., Bengio, S., Hardt, M., Recht, B. and Vinyals, O., 2021. Understanding deep learning (still) requires rethinking generalization. Communications of the ACM, 64(3), pp.107-115.

Zhao, J., Liu, K. and Xu, L., 2016. Sentiment analysis: mining opinions, sentiments, and emotions.

Maynard, D.G. and Greenwood, M.A., 2014, March. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In Lrec 2014 proceedings. ELRA.

Poria, S., Majumder, N., Hazarika, D., Cambria, E., Gelbukh, A. and Hussain, A., 2018. Multimodal sentiment analysis: Addressing key issues and setting up the baselines. IEEE Intelligent Systems, 33(6), pp.17-25.

Japkowicz, N. and Stephen, S., 2002. The class imbalance problem: A systematic study. Intelligent data analysis, 6(5), pp.429-449.

Buda, M., Maki, A. and Mazurowski, M.A., 2018. A systematic study of the class imbalance problem in convolutional neural networks. Neural networks, 106, pp.249-259.

Eisenstein, J., 2013, June. What to do about bad language on the internet. In Proceedings of the 2013 conference of the North American Chapter of the association for computational linguistics: Human language technologies (pp. 359-369).

Pradha, S., Halgamuge, M.N. and Vinh, N.T.Q., 2019, October. Effective text data preprocessing technique for sentiment analysis in social media data. In 2019 11th international conference on knowledge and systems engineering (KSE) (pp. 1-8). IEEE.

Richardson, L., 2007. Beautiful soup documentation.

Avasarala, S., 2014. Selenium WebDriver practical guide. PACKT publishing.

Ramos, J., 2003, December. Using tf-idf to determine word relevance in document queries. In Proceedings of the first instructional conference on machine learning (Vol. 242, No. 1, pp. 29-48).

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, pp.321-357.

Ghazvini, A., Sharef, N.M. and Sidi, F.B., 2024. Prediction of Course Grades in Computer Science Higher Education Program Via a Combination of Loss Functions in LSTM Model. IEEE Access.

Chollet, F. and Chollet, F., 2021. Deep learning with Python. Simon and Schuster.

Grinberg, M., 2018. Flask web development. " O'Reilly Media, Inc.".

Bengio, Y., 2012. Practical recommendations for gradient-based training of deep architectures. In Neural networks: Tricks of the trade: Second edition (pp. 437-478). Berlin, Heidelberg: Springer Berlin Heidelberg.

Jurafsky, D., 2000. Speech and language processing.