Advances in Data Science

Final Project: Extension of a previous project

Topic: Cell Phone Recommendation System with Sentiment Analysis

Tanmay Shekhar (002747412)
Riya Virani (002747048)

Introduction

- Background:

  In this project, our objective is to enhance the user experience within an online retail platform that specializes in cell phones by providing personalized product recommendations. Leveraging an amazon dataset of customer reviews about various cell phone models, we have undertaken an extensive analysis and implemented a recommendation system. Initially, we conducted sentiment analysis on the reviews using Natural Language Processing (NLP) techniques, including traditional machine learning algorithms such as Naive Bayes and deep learning algorithms such as LSTM, to classify sentiments. Building upon this foundation, we have now delved deeper into the data by employing Word2Vec embeddings. By harnessing the power of Word2Vec, we aim to capture the semantic nuances of customer reviews, allowing us to create embeddings for both users and cell phone models. These embeddings serve as the basis for a recommendation system that offers personalized product suggestions, ultimately improving user satisfaction and engagement on the platform.

- Motivation:

  The motivation behind this project stems from the ever-increasing importance of delivering tailored user experiences in the competitive landscape of online retail, especially in the realm of consumer electronics like cell phones. With a plethora of choices available to customers, the ability to offer precise and relevant product recommendations is a strategic advantage. Understanding the sentiment of customer reviews is the first step in comprehending their preferences, while the integration of Word2Vec embeddings enables us to delve deeper into the rich textual data and uncover hidden insights. By developing an advanced recommendation system, we aim to address the challenge of information overload, making it easier for users to discover products that align with their needs and preferences. Ultimately, our goal is to enhance user satisfaction, increase engagement, and drive business growth by leveraging the power of natural language processing and recommendation algorithms.

- Goal:

  The primary goal of this project is to design and implement an effective recommendation system for an online retail platform specializing in cell phones. Through the utilization of sentiment analysis and Word2Vec embeddings, we aim to provide personalized product recommendations to users based on their historical behavior and the content of their reviews. Our objectives include enhancing user satisfaction by delivering relevant product suggestions, increasing user engagement and retention, and ultimately driving higher conversion rates and revenue for the platform. Additionally, we intend to explore the impact of this recommendation system on mitigating data imbalances and improving the discoverability of less-reviewed cell phone models. By achieving these goals, we aim to establish a robust and data-driven approach to customer engagement and product promotion within the online retail space.

Methodology

This section outlines the systematic approach taken to develop a sentiment analysis model capable of classifying customer reviews. The methodology encompasses several stages: data acquisition, preprocessing, exploratory data analysis (EDA), data balancing, feature extraction, model training for sentiment analysis, and evaluation, then using features such as the product_name, price and Average rating to Build a Recommendation system for an increase in user engagement.

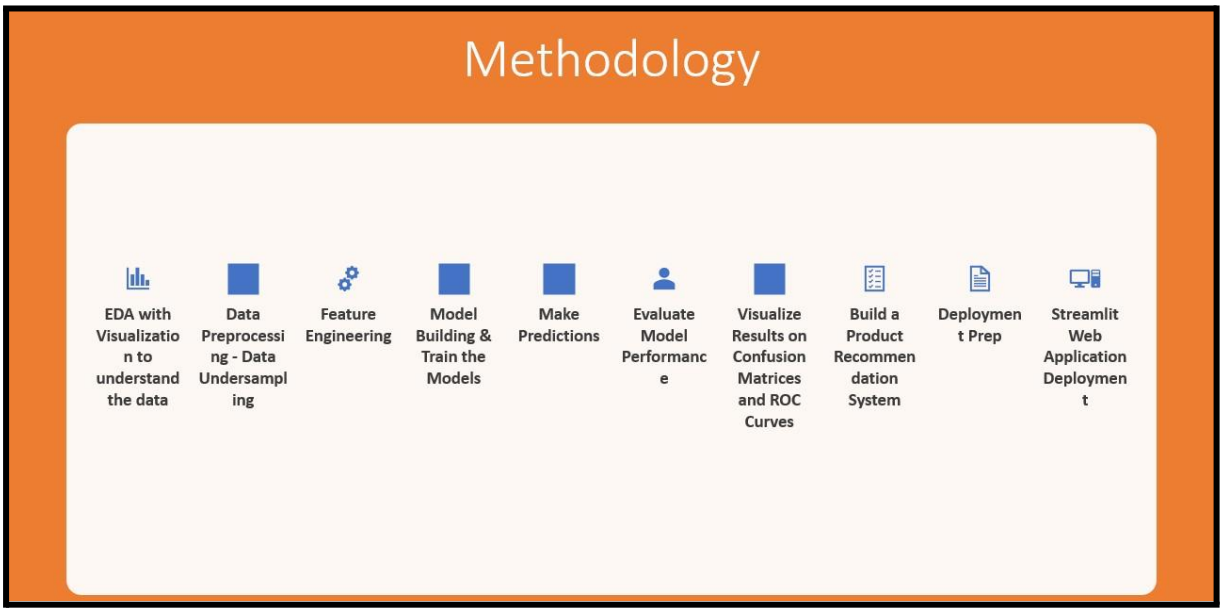


Figure 1: Methodology

Dataset Overview

The dataset has been obtained by combining two datasets. It is an Amazon dataset about cell phone reviews by several customers.

This dataset contains 67,986 reviews from Amazon about cell phones from 2004 up until 2020. Each review can be associated with an item and brand name and comes with a rating ranging from 1 to 5. This makes the dataset a perfect sample for sentiment analysis.

The rating is an integer value ranging from 1 to 5, with 1 being the lowest (negative sentiment) and 5 being the highest (positive sentiment). The dataset is reflective of customer interactions and opinions about various products offered on the platform.

```
combined_data.head()
```

Out[6]:

	asin	brand	title_x	url	image	rating_x	reviewUrl	totalReviews	price
0	B0000SX2UC	NaN	Dual-Band / Tri-Mode Sprint PCS Phone w/ Voice...	https://www.amazon.com/Dual-Band-Tri-Mode-Acti...	https://m.media-amazon.com/images/I/2143EBQ210...	3.0	https://www.amazon.com/product-reviews/B0000SX2UC	14	0.0
1	B0000SX2UC	NaN	Dual-Band / Tri-Mode Sprint PCS Phone w/ Voice...	https://www.amazon.com/Dual-Band-Tri-Mode-Acti...	https://m.media-amazon.com/images/I/2143EBQ210...	3.0	https://www.amazon.com/product-reviews/B0000SX2UC	14	0.0
2	B0000SX2UC	NaN	Dual-Band / Tri-Mode Sprint PCS Phone w/ Voice...	https://www.amazon.com/Dual-Band-Tri-Mode-Acti...	https://m.media-amazon.com/images/I/2143EBQ210...	3.0	https://www.amazon.com/product-reviews/B0000SX2UC	14	0.0
3	B0000SX2UC	NaN	Dual-Band / Tri-Mode Sprint PCS	https://www.amazon.com/Dual-Band-Tri-Mode-Acti...	https://m.media-amazon.com/images/I/2143EBQ210...	3.0	https://www.amazon.com/product-reviews/B0000SX2UC	14	0.0

asin:	A unique identifier for the product.
brand:	The brand or manufacturer of the product.
title_x:	The product's title.
url:	The URL of the product on Amazon.
image:	URL of the product's image.
rating_x:	The product's AVERAGE rating on Amazon.
reviewUrl:	URL to access the product's reviews.
totalReviews:	The total number of reviews for the product.
price:	The price of the product.
originalPrice:	The original price of the product (not discounted price).

name:	The name of the reviewer.
rating_y:	The rating given by each reviewer.
date:	The date when the review was posted.
verified:	Indicates whether the review is from a verified purchase (True/False).
title_y:	The title of the reviewer's review.
body:	The content of the review.
helpfulVotes:	The number of helpful votes the review received.

Exploratory Data Analysis:

Several Python functions, namely head, describe, and info, were used to analyze the data's features and learn more about the structure and contents. Along with the following plots:

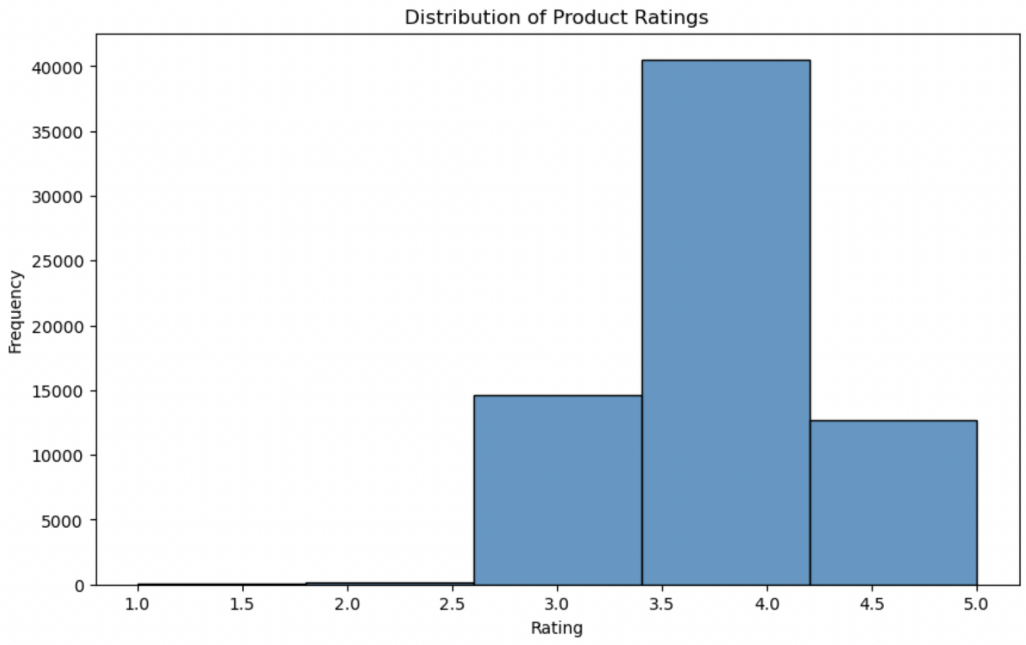


Figure 2: Distribution of the Ratings of products

This histogram showcases the frequency distribution of product ratings. Most ratings are clustered around the higher end of the scale, indicating a skew towards positive reviews.

```
In [34]: df.head()
```

Out[34]:

	asin	brand	avg_rating_per_item	totalReviews	price	rating	date	verified	review_title	review_body	helpfulVot
14	B0009N5L7K	Motorola	3.0	7	49.95	1	2016-03-05	1	Stupid phone	DON'T BUY OUT OF SERVICE	0
15	B0009N5L7K	Motorola	3.0	7	49.95	4	2006-02-09	0	Exellent Service	I have been with nextel for nearly a year now ...	0
16	B0009N5L7K	Motorola	3.0	7	49.95	5	2006-02-07	0	I love it	I just got it and have to say its easy to use,...	0
17	B0009N5L7K	Motorola	3.0	7	49.95	1	2016-12-20	1	Phones locked	1 star because the phones locked so I have to ...	0
18	B0009N5L7K	Motorola	3.0	7	49.95	5	2009-12-13	1	Excellent product	The product has been very good. I had used thi...	0

Figure 3: Initial Data

This line graph depicts the average rating of products over time. There is a visible trend showing fluctuations in the average rating with time, but there seems to be a general improvement or stabilization in product ratings after 2010.

Next, we formed a 'WordCloud' using the words in the body of the review within the Dataset.

Figure 5: WordCloud created using the reviews in the dataset

In the following step I plotted a Correlation matrix in order to check the correlation between the features or columns.

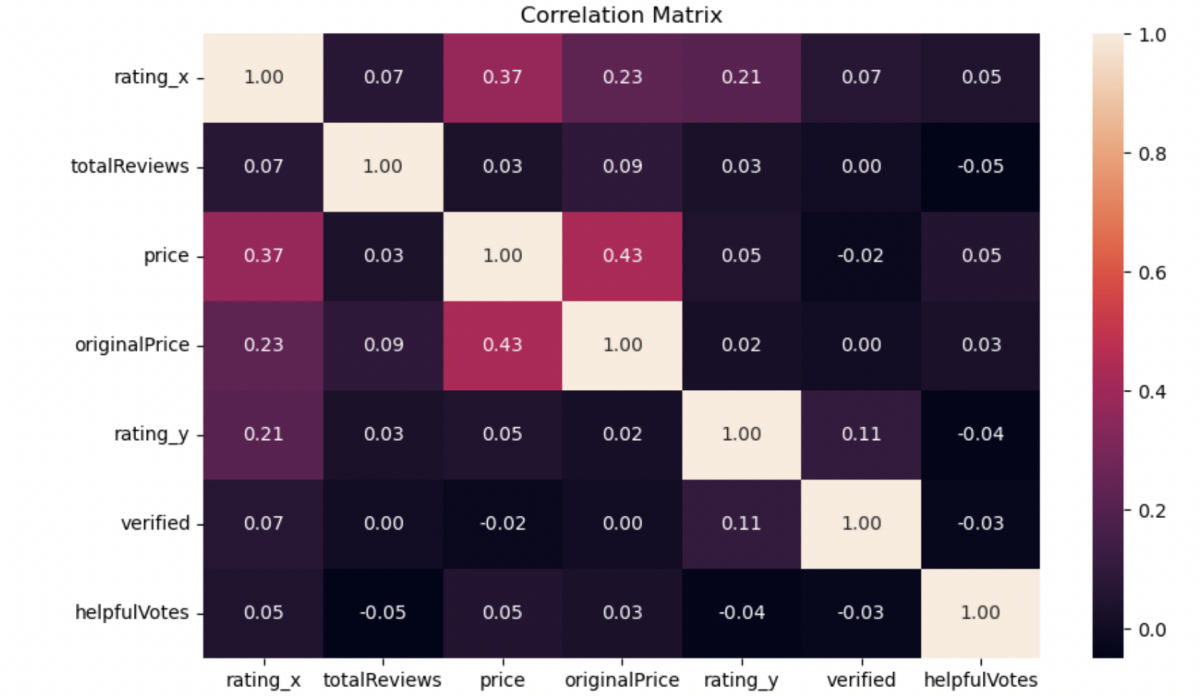


Figure 6: Correlation Matrix

Next, we created a graph for checking Anamolies / Outliers.

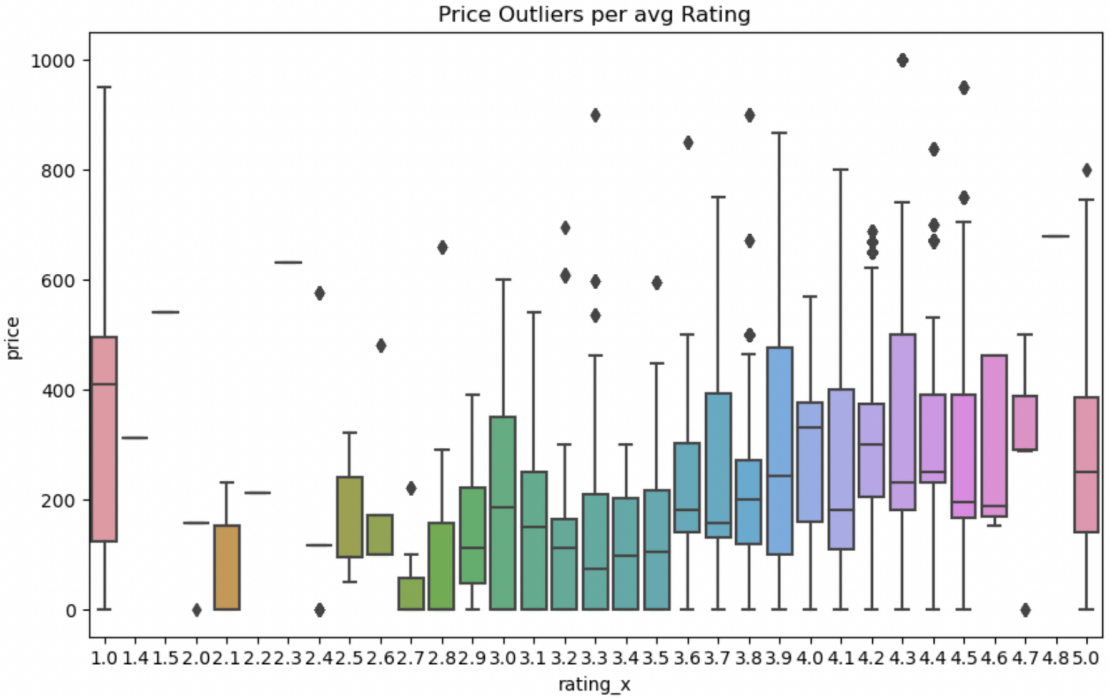


Figure 7: Price Outliers per avg Rating

From the EDA we performed we understood a significant amount about the kind of Data we are dealing with, such as:

1. The Biases within the dataset which is the skewness of ratings towards the higher values
2. The possible obstacles that could hinder with our model training such as stop words and Upper case letters within the words
3. Rating Distribution: We assessed the distribution of star ratings to identify imbalances.
4. Text Analysis: Frequent words and phrases were analyzed to understand common themes across different ratings.

Feature Engineering

Added a column called as review_length.

For the Naive Bayes model to process textual data, it was necessary to convert the text into a numerical format:

TF-IDF Vectorization: The Term Frequency-Inverse Document Frequency technique was utilized to transform the text into a matrix of TF-IDF features, reflecting the importance of words within the corpus.

Data Pre-processing

Overcoming the problem of an imbalanced dataset by using sampling techniques

The data we are using is rather uneven, and we have found that there is a large difference between the number of Product Ratings that are high as compared to the product ratings that are Low and this is a Bias within the Dataset that we Intend to solve in this stage of Data Preprocessing. To solve this problem, we chose to produce a more balanced dataset by Undersampling the Higher Rated product reviews.

To mitigate the bias introduced by imbalanced classes, we employed undersampling techniques:

Undersampling: High-frequency classes (ratings 4 and 5) were randomly undersampled to match the count of lower-rating classes.

I created a new dataset named "balanced_df" by combining the original dataset of lower Rated Reviews and the new undersampled version of the Higher Rated reviews. This strategy aids in improving the dataset's representation of the minority class..

Then, using a bar chart with the x-axis representing the class labels (ratings of the reviews) and the y-axis representing the amount of reviews for each rated product. After the

undersampling processes, this bar chart gives a clear visual representation of the achieved balance between the two classes.

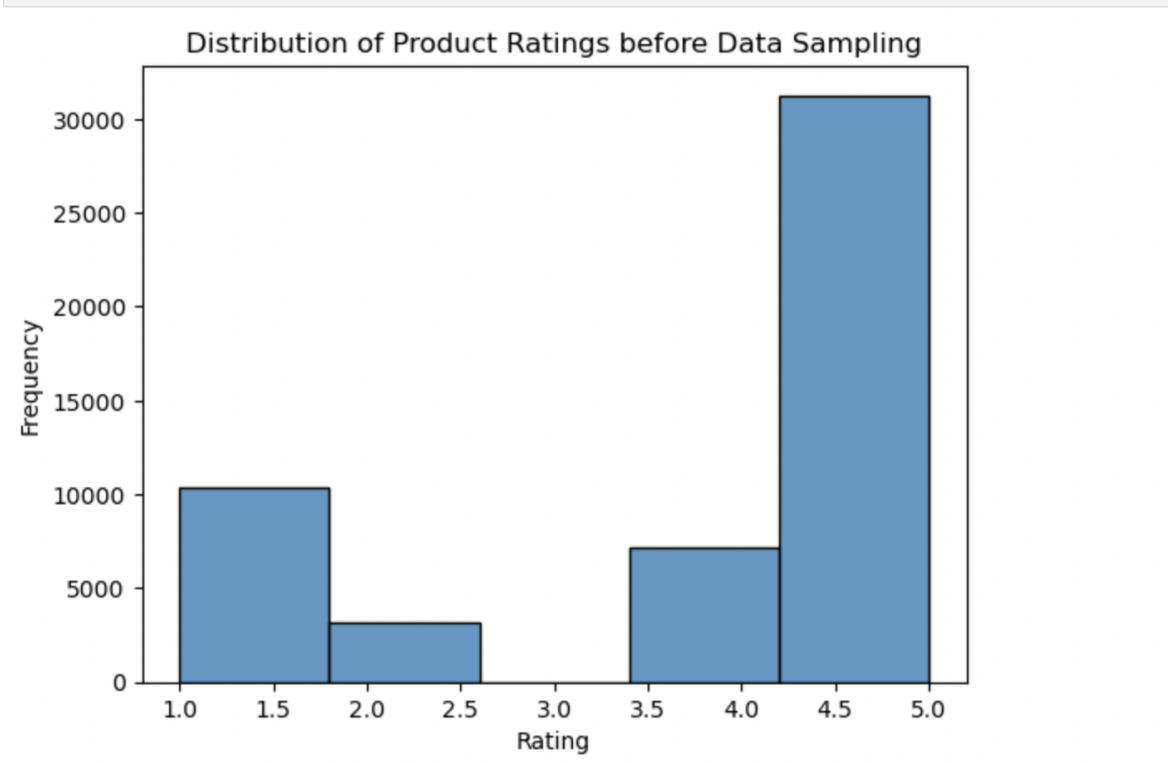


Figure 10: Bar plot before sampling

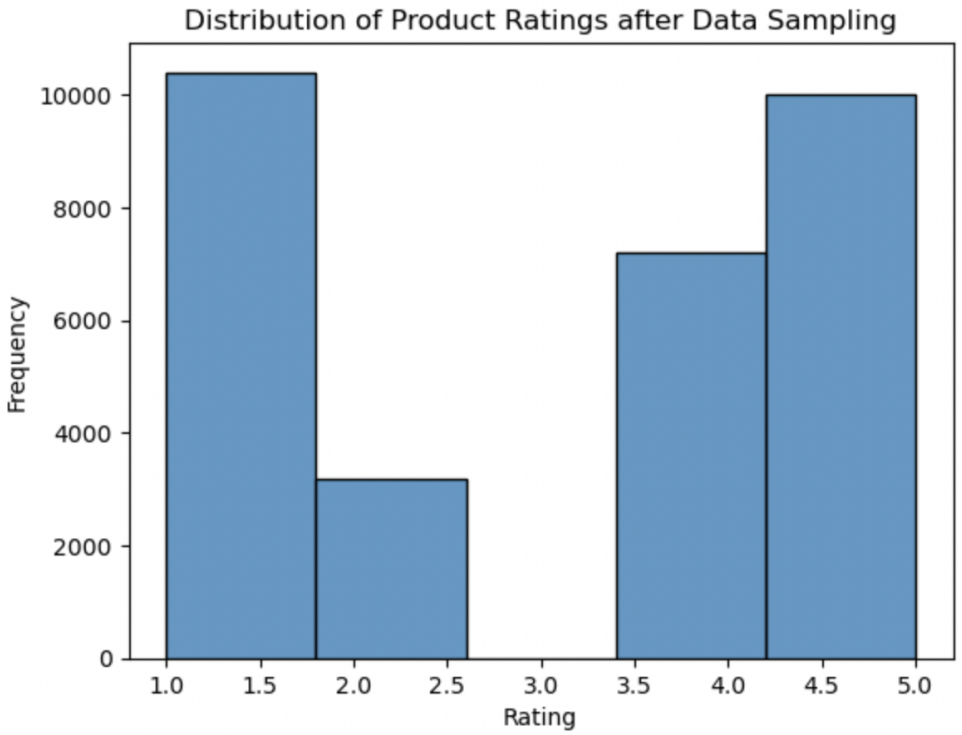


Figure 10: Bar plot after sampling

The next preprocessing steps were implemented as follows to clean and prepare the data:

- Handling Missing Values: Missing values in the 'brand' and 'review' columns were identified and handled appropriately. The number of missing data initially in the Dataset were as shown below.

```
Percentage of Missing Data:
                Column Name  Missing Percentage
helpfulVotes    helpfulVotes           59.969700
brand                brand            0.294178
body                  body            0.030889
title_y            title_y            0.020592
name                  name            0.002942
originalPrice  originalPrice           0.000000
verified          verified            0.000000
date                  date            0.000000
rating_y          rating_y            0.000000
asin                  asin            0.000000
totalReviews    totalReviews           0.000000
reviewUrl          reviewUrl           0.000000
rating_x          rating_x            0.000000
image                image            0.000000
url                    url            0.000000
title_x            title_x            0.000000
price                price            0.000000
```

- Dealing with '0' Values: 'Price' and 'originalPrice' columns were analyzed, and rows with a '0' price were removed. Given that the rows that did have the value as zero were not that much as compared to the entire dataset and dropping them would not hurt our model Training much and hence we dropped them.

- Dropping Irrelevant Features: Columns not contributing to sentiment analysis, such as 'asin', 'url', 'image', 'reviewUrl', 'name', were dropped. We believed that these

features were not going to have any impact in the model and hence decided to drop them solely based on domain knowledge.

- Text Preprocessing: The review texts were cleaned by removing HTML tags, punctuation, and numbers.
- Stopwords were also removed, and all text was converted to lowercase. By removing stopwords, the results of sentiment analysis become more interpretable. You can easily identify the significant words that contribute to the sentiment score, which can be valuable for understanding why a particular sentiment was assigned to a piece of text
- Performed lemmatization

Training and Testing Data

Now that the sampling is completed and the data is now balanced, we will continue with splitting the data into Training and Testing. 80% of the total data will be used to train the ML model, with the remaining 20% used to test the model's accuracy.

Training and Testing Machine Learning Models:

The project aimed at classifying reviews into positive or negative sentiments. This binary classification approach formed the basis of the sentiment analysis.

The dataset was split into training and test sets, ensuring a fair evaluation of the model's performance.

Two models were implemented and trained: Naive Bayes and LSTM. Each model was chosen for its strengths in handling different aspects of text data and sentiment analysis.

The models were evaluated using various metrics, including classification reports, confusion matrices, and ROC curves, to assess their performance accurately.

- **Naive Bayes:**
- **Long Short Term Memory(Deep Learning):**

Result and Analysis

Performance Evaluation

For each model, we have:

1. Compared the training set results v/s the testing set results
2. Plotted Confusion Matrix and ROC Curve to determine outcomes

A. Naive Bayes Modeling:

```
Classification Report for Training Data:
              precision    recall  f1-score   support

           0       0.92      0.88      0.90     10856
           1       0.91      0.94      0.93     13760

    accuracy                           0.92     24616
   macro avg       0.92      0.91      0.91     24616
weighted avg       0.92      0.92      0.92     24616

Classification Report for Testing Data:
              precision    recall  f1-score   support

           0       0.90      0.83      0.87      2716
           1       0.87      0.93      0.90      3438

    accuracy                           0.89      6154
   macro avg       0.89      0.88      0.88      6154
weighted avg       0.89      0.89      0.89      6154
```

Figure 11: Training Results v/s Testing Results

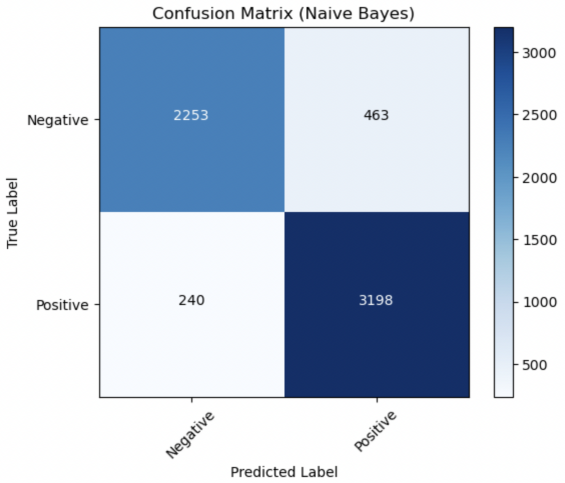


Figure 12: Test Confusion Matrix for Naive Bayes

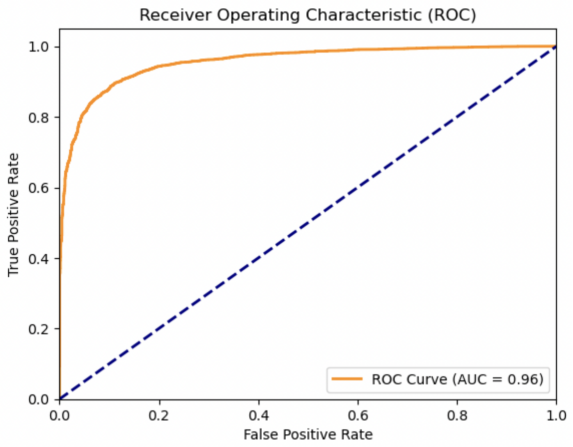


Figure 13: ROC Curve for Naive Bayes

B. Long Short Term Memory (Deep Learning):

```
193/193 [==============================] - 0s 490us/step
770/770 [==============================] - 0s 437us/step
Classification Report for Training Data:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99     10856
           1       0.99      0.99      0.99     13760

    accuracy                           0.99     24616
   macro avg       0.99      0.99      0.99     24616
weighted avg       0.99      0.99      0.99     24616


Classification Report for Testing Data:
              precision    recall  f1-score   support

           0       0.88      0.89      0.89      2716
           1       0.91      0.90      0.91      3438

    accuracy                           0.90      6154
   macro avg       0.90      0.90      0.90      6154
weighted avg       0.90      0.90      0.90      6154
```

Figure 16: Train/Test Accuracy

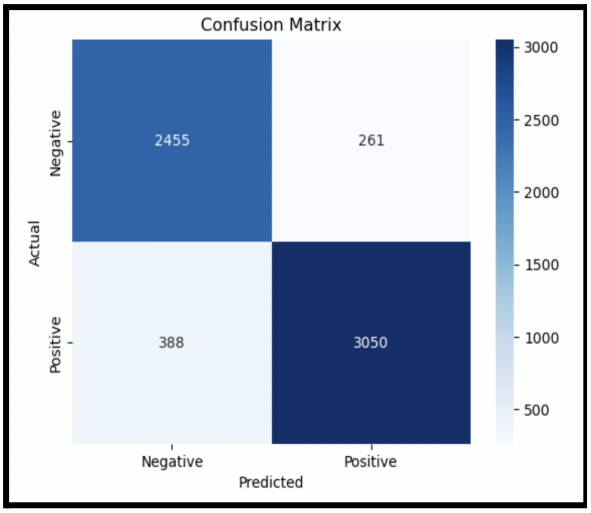


Figure 17: Test Confusion Matrix for LSTM

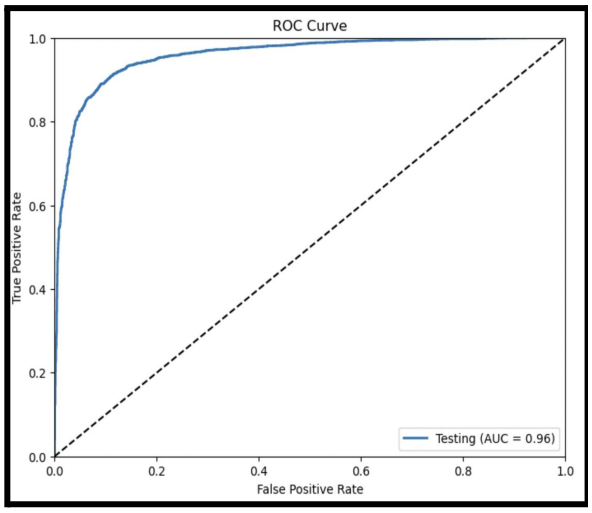


Figure 18: ROC Curve for LSTM

Recommendation System:

A significant portion of the project was dedicated to developing the recommendation system.

```
balanced_df.head()
```

	asin	brand	avg_rating_per_item	totalReviews	price	rating	date	verified	helpfulVotes	review	review_length	target	sentiment
14	B0009N5L7K	Motorola	3.0	7	49.95	1	2016-03-05	1	0.0	stupid phone dont buy service	29	0	0
17	B0009N5L7K	Motorola	3.0	7	49.95	1	2016-12-20	1	0.0	phone locked star phone locked pay additional ...	56	0	0
19	B0009N5L7K	Motorola	3.0	7	49.95	1	2005-07-21	0	0.0	warning problem nextel stop canceled service g...	1629	0	0
22	B000SKTZ0S	Motorola	2.7	22	99.99	1	2017-05-13	1	0.0	seems doesnt work existing att sim card purcha...	221	0	0
23	B000SKTZ0S	Motorola	2.7	22	99.99	1	2019-03-13	1	0.0	supply needed phone come charger didnt sims card	48	0	0

Data was grouped by 'asin' to identify unique products. Further feature engineering was conducted using TF-IDF and MinMaxScaler for 'product_name'.

```
grouped_df = grouped_df.reset_index()
grouped_df.head()
```

✓ 0.0s Python

	asin	brand	product_name	url	image	avg_rating_per_item	totalReviews	price
0	B0009N5L7K	Motorola	Motorola I265 phone	https://www.amazon.com/Motorola-i265-I265-phon...	https://m.media-amazon.com/images/I/419WBAVDAR...	3.0	5	49.95
1	B000SKTZ0S	Motorola	MOTOROLA C168i AT&T CINGULAR PREPAID GOPHONE C...	https://www.amazon.com/MOTOROLA-C168i-CINGULAR...	https://m.media-amazon.com/images/I/71b+q3ydkI...	2.7	13	99.99
2	B001DCJAJG	Motorola	Motorola V365 no contract cellular phone AT&T	https://www.amazon.com/Motorola-V365-contract-...	https://m.media-amazon.com/images/I/61LYNCVrrK...	3.1	6	149.99
3	B002WTC1NG	Motorola	Motorola Barrage V860 Phone (Verizon Wireless)	https://www.amazon.com/Motorola-Barrage-V860-V...	https://m.media-amazon.com/images/I/81k6Nq0KI1...	3.6	187	139.99
4	B0033SFV5A	Samsung	Verizon or PagePlus Samsung Smooth U350 Great ...	https://www.amazon.com/Verizon-PagePlus-Samsun...	https://m.media-amazon.com/images/I/61nD-TYqHm...	3.3	37	64.99

Cosine similarity measures were calculated to facilitate the recommendation process.

The team also incorporated Word2Vec embeddings to analyze product names, adding a layer of sophistication to the recommendation logic.

Custom functions were developed to generate product recommendations based on similarity scores, providing a tailored experience for the user.

Feature Engineering for Recommendation system model:

Implementation Details

- Applying TF-IDF to 'product_name':
  - The TfidfVectorizer from Scikit-learn is employed to transform 'product_name' into a TF-IDF matrix.
  - This step converts the textual data of product names into numerical form, capturing the importance of words in relation to the entire dataset.
  - By setting stop_words='english', common English words that add little value to the analysis are filtered out, allowing the vectorizer to focus on more meaningful terms in the product names.
- Normalizing 'avg_rating_per_item' and 'price':
  - The MinMaxScaler, also from Scikit-learn, is used for normalizing the 'avg_rating_per_item' and 'price' columns.
  - This normalization brings these features into a similar scale, preventing any feature from dominating others due to differences in scale.
  - Normalization is crucial for models or calculations that are sensitive to the scale of input features, like cosine similarity.

- Combining Features into One Matrix:
    - The TF-IDF matrix for product names and the normalized 'avg_rating_per_item' and 'price' features are horizontally stacked to create a single, unified feature matrix.
    - This matrix now encapsulates both textual and numerical attributes of the products in a format suitable for further analysis.
- Recomputing the Cosine Similarity Matrix:
    - With the comprehensive feature matrix ready, cosine similarity is calculated for each pair of products.
    - Cosine similarity measures the cosine of the angle between two non-zero vectors in a multi-dimensional space, in this case, our feature vectors. It is an effective measure for gauging similarity.
    - The resultant cosine similarity matrix is a pivotal element in the recommendation engine, as it quantifies the similarity between each pair of products based on the engineered features.

Recommendation system model:

We used two main methods or metrics on which we built our Recommendation models and those were Cosine similarity and Word2Vec.

- **Similarity Matrix:** We computed a cosine similarity matrix from the TF-IDF vectors. This similarity matrix quantifies the likeness between products based on their names.
- Function Design: A function was created that takes a product name as input and returns a list of top similar products based on cosine similarity scores from the TF-IDF vectors.

- **Word2Vec Based Model Implementation:** A Word2Vec model was trained on the tokenized product names to capture semantic relationships in the text.
- The recommendation function was further revised to work with the Word2Vec-based features, providing recommendations that consider both textual semantics and price.

After Performing the Sentiment Analysis to our Dataset we added a column labelled as Sentiment and our main goal with this column was to provide the user with a Sentiment Count graph of the product of his/her choice which would help the user decide if the product is worth buying or no.

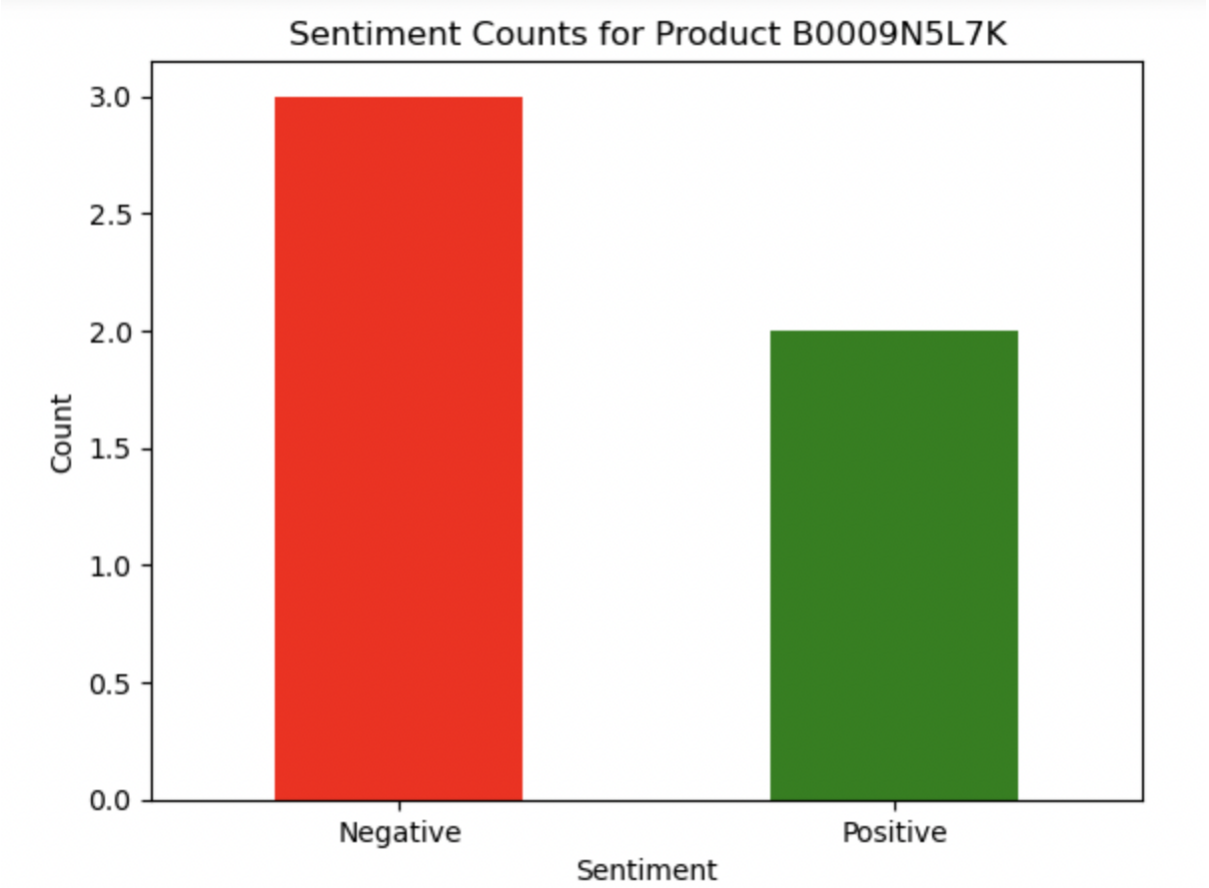


Figure 20: Test Confusion Matrix for GRU

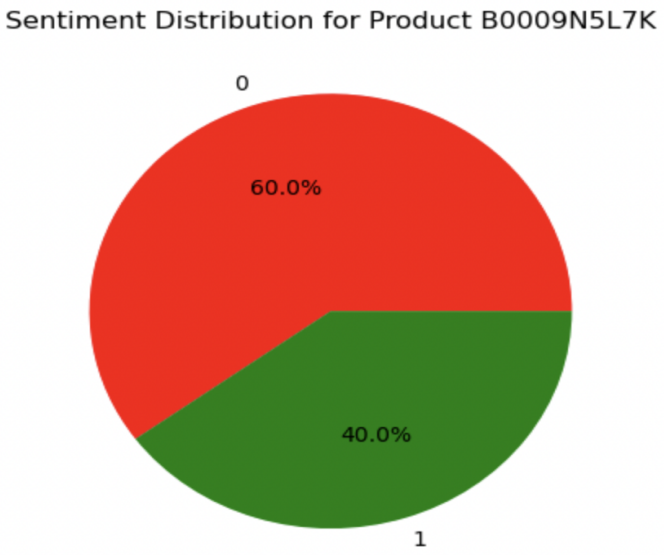
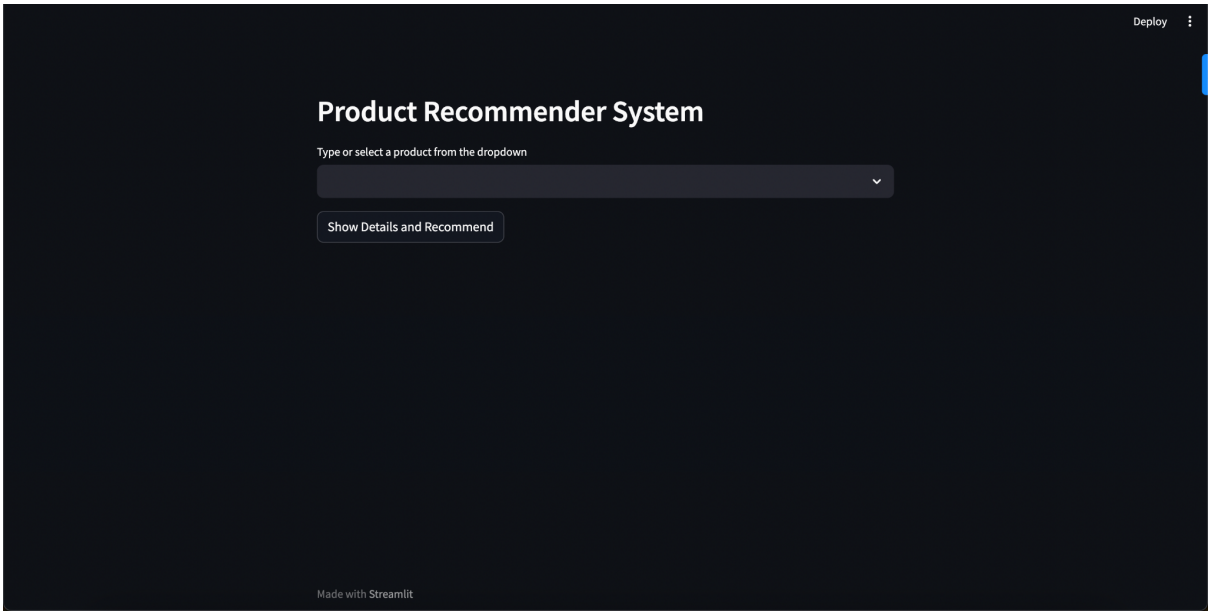


Figure 21: Test Confusion Matrix for GRU

Streamlit UI:


Deploy
Product Recommender System
Type or select a product from the dropdown
Show Details and Recommend
Made with Streamlit

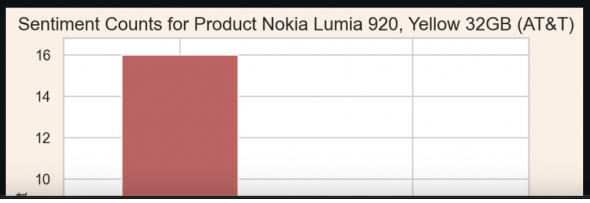

Deploy
Product Recommender System
Type or select a product from the dropdown
Samsung Galaxy S7 - Black - 32GB - Verizon (Renewed)
Apple iPhone 6S, 16GB, Gold - For AT&T (Renewed)
Apple iPhone 7, 32GB, Black - Fully Unlocked (Renewed)
Samsung Galaxy Note 4 N910P 32GB Frosted White - Sprint (Renewed)
Apple iPhone 6S, 128GB, Rose Gold - For AT&T / T-Mobile (Renewed)
Samsung TWSAS120VCP-3 Galaxy J1 Luna 4G LTE Total Wireless Prepaid Smartphone
Samsung Galaxy S7 Edge G935V 32GB Gold - Verizon (Renewed)
Made with Streamlit

Deploy

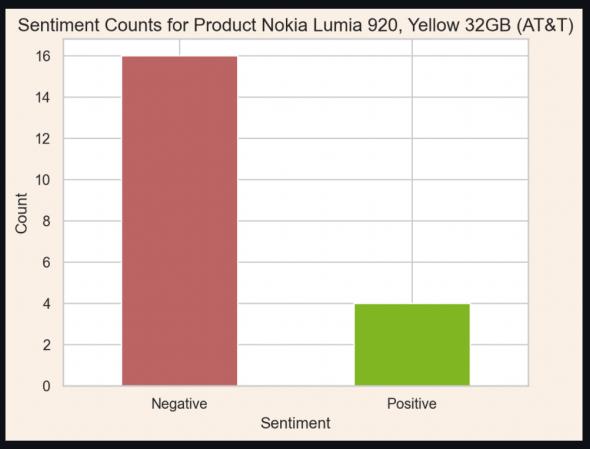
Price: 184.18

Average Rating: 3.0

Nokia Lumia Ico Xiaomi Mi 9 64G Apple iPhone 6S Google Pixel XL Samsung Galaxy



Deploy



Conclusion:

The sentiment analysis project aimed at dissecting cell phone reviews from Amazon has provided substantial insights into the application of machine learning and deep learning models in NLP tasks. The comprehensive journey from data acquisition through preprocessing to modeling and evaluation has underscored the intricate challenges and considerations inherent in sentiment analysis.

Achievements:

Sentiment Analysis:

Objective: The goal was to extract insights from customer reviews, categorizing them into positive, negative, or neutral sentiments. This understanding is crucial for businesses to gauge customer satisfaction and improve their offerings.

Process: Utilizing natural language processing (NLP) techniques, the textual data from reviews was preprocessed, tokenized, and vectorized. Machine learning models like Naive Bayes, SVM, or deep learning models like LSTM could have been employed to classify the sentiment of each review.

Outcome: The sentiment analysis would provide valuable insights into customer opinions and trends, which can inform product development, marketing strategies, and customer service improvements.

Recommendation System:

Objective: To develop a system that suggests products to users based on their preferences, using content-based filtering approaches.

TF-IDF Approach: We initially employed a TF-IDF vectorization strategy to transform 'product_name' data into a numerical format, followed by cosine similarity calculations to find similar products.

Word2Vec Approach: As an alternative, we implemented a Word2Vec model, which provided semantic representations of the product names. This model was combined with normalized price data for a more nuanced recommendation.

Streamlit UI: A user-friendly interface was created using Streamlit, allowing users to interact with the system and receive personalized recommendations.

Integration of Sentiment Analysis and Recommendation System:

The sentiment analysis and recommendation system, though developed independently, can be integrated. The insights from sentiment analysis (like identifying highly favored products) can refine the recommendation engine, prioritizing products with positive sentiments.

Key Takeaways:

Data Utilization: The project demonstrates effective utilization of various data types – textual data for sentiment analysis and a combination of textual and numerical data for the recommendation system.

Technological Diversity: It highlights the use of diverse technologies and methodologies, from NLP and machine learning to vector space modeling and user interface design.

User Experience and Business Insights: The end goal of enhancing user experience through personalized recommendations and providing businesses with actionable insights from customer feedback is central to this project.

Future Scope:

Model Refinement: There's scope for refining models using more complex algorithms or incorporating additional features (like user demographics, historical purchase data).

Scalability and Real-Time Analysis: Future developments could focus on scaling the system for larger datasets and implementing real-time analysis for dynamic recommendations.

Integrating User Feedback: Incorporating direct user feedback into the recommendation engine could further personalize and improve the accuracy of suggestions.

Overall, this project stands as a testament to the power of machine learning and data science in transforming customer data into meaningful, actionable insights and services.

Insights:

- Deep Learning Superiority: LSTM model showed a marked improvement over the traditional model in terms of both accuracy and the ability to discern sentiment from the sequence of words in the reviews.

- Bias Mitigation: The strategic approach to handling dataset imbalance through undersampling was successful in mitigating bias, thereby enhancing the models' generalizability to new data.

Sources of Bias:

- **Imbalanced Data:** Initially, the dataset was skewed towards positive reviews. Though undersampling is a step in the right direction, it may not be entirely sufficient. Future studies might explore synthetic data generation, like SMOTE, to augment minority classes without losing valuable data.

- **Cultural and Contextual Nuances:** The models' understanding of sentiment was limited to the text's linguistic features and may not fully grasp the cultural and contextual nuances. Incorporating metadata or domain-specific sentiment lexicons could enhance model sensitivity to such nuances.

In conclusion, this project not only showcased the potential of AI in understanding customer sentiment but also highlighted the critical role of preprocessing and the thoughtful application of modeling techniques. As sentiment analysis continues to evolve, it will undoubtedly become an even more powerful tool for businesses to harness customer feedback, driving improvements and fostering a better understanding of consumer behavior.

Future Scope:

- Expanding Data Sources: Integrating reviews from different platforms could help build a more robust and diverse dataset, minimizing the risk of platform-specific bias.

- Model Optimization: Hyperparameter tuning and advanced model architectures, such as Transformers or BERT, could further improve performance.

- Explainability: Incorporating model explainability measures, such as SHAP or LIME, could provide insights into model decision-making, fostering trust and interpretability.

Dataset Link:

https://www.kaggle.com/datasets/grikomsn/amazon-cell-phones-reviews?select=20191226-items.csv

Contributions: (50-50)

Tanmay Shekhar (002747412): Data Preprocessing, Naive Bayes Classifier Implementation, Long Short Term Memory Implementation, Word2Vec Implementation, Cosine Similarity Implementation, Integration of Sentiment Analysis Model and Recommendation System, Report Creation

Riya Virani (002747048): Datasets combination, EDA, Data Preprocessing, Support Vector Machine Implementation, Cosine Similarity Implementation, Word2Vec Implementation, Integration of Sentiment Model, Report Creation

References:

1. https://medium.com/towards-data-science/nlp-sentiment-analysis-for-beginners-e7897f976897
2. https://medium.com/mlearning-ai/10-python-functions-you-need-to-apply-before-you-build-your-nlp-sentiment-analysis-model-874a37e0217e
3. GPT 3.5
4. https://www.kaggle.com/datasets/grikomsn/amazon-cell-phones-reviews/code