2023

Enhancing Customer Purchasing Experience with Fake Review Detector and Personalized Recommendation Engines

DERON NICHOLSON

TEK SOLUTIONS | 34 DS Street, Kingston 10

Table of Contents

Abstract

In the dynamic realm of e-commerce, the intersection of personalized recommendations and the challenge of fake review detection has emerged as a pivotal concern. This paper presents a comprehensive study that leverages the CRISP-DM methodology to address these issues and enhance the online shopping experience.

To combat the menace of fake reviews, sentiment strength and TF-IDF analysis were employed, along with review length, to construct a robust fake review detection system. Classification models including Naive Bayes, SVM, and Decision Trees were explored, revealing SVM's superiority with an Area Under the Receiver Operating Characteristic (ROC) curve of 0.86 and an Area Under the Precision-Recall (PR) curve of 0.95.

Simultaneously, personalized recommendations were optimized using clustering models (KMeans and Hierarchical Clustering) integrating sentiment strength as a feature. KMeans edged ahead in performance with a narrow silhouette score advantage. Collaborative filtering was employed using ALS, achieving an RMSE of 1.15.

In the backdrop of this technical analysis, the introduction underscores the significance of personalized recommendations in the vast e-commerce landscape. Fake reviews, often deceiving customers towards subpar products, have marred consumer trust. Instances like Amazon's persistent legal battles with fake review brokers emphasize the gravity of the issue. The interconnected challenge of personalized suggestions and fake reviews has led to new ways of thinking. These approaches aim to improve suggestions and fight against fake reviews.

Introduction

In the ever-evolving landscape of e-commerce, where consumers are presented with an overwhelming array of choices, the role of personalized recommendations has emerged as a crucial factor in enhancing user experience and driving sales. Simultaneously, the rampant proliferation of fake reviews has cast a shadow over the trustworthiness of online platforms. Customers have opted for often low-quality products due to the bombardment of fake reviews which has led to their diminished confidence in online shopping (Team, 2020). As consumers increasingly rely on online reviews to make informed purchasing decisions, the need to address the issue of fake reviews has become imperative to maintain the integrity of e-commerce ecosystems. Amazon for instance, an ecommerce giant, has been in a constant legal battle with fake review brokers to make their ecosystem a safe place. According to Amazon (2023), in February of this year, 6 lawsuits were filed to protects its customer base and selling partners.

The intricate relationship between personalized recommendations and the threat posed by fake reviews has sparked a captivating and multifaceted dilemma for e-commerce platforms, scholars, and enterprises in equal measure. By harnessing cutting-edge technologies, data analysis, and understanding user behaviors, the sector is currently immersed in the creation of inventive approaches. These approaches aim to not only refine personalized recommendations but also to simultaneously counteract the encroachment of deceitful reviews that undermine the authenticity of the e-commerce landscape.

The phenomenon of personalized recommendations has revolutionized how consumers navigate the vast expanse of products available online. Traditional methods of presenting generic lists of products have given way to sophisticated algorithms that analyze historical purchase data, browsing patterns, and even contextual cues to tailor suggestions that are uniquely suited to everyone’s preferences and needs. This has not only significantly improved user satisfaction and engagement but has also proven to be a potent tool for boosting sales and revenue and the overall customer retention for e-commerce businesses (Basu, 2021).

However, as e-commerce platforms strive to provide increasingly relevant recommendations, they also grapple with ethical considerations surrounding user privacy and data security (Gorripati et al, 2021). Striking a balance between hyper-personalization and respecting users' boundaries is an ongoing challenge. Furthermore, the sheer complexity of developing algorithms that accurately predict user preferences and behavior adds another layer of intricacy to this multifaceted endeavor.

In parallel, the surge in the prominence of online reviews as a critical source of information for potential buyers has brought to the forefront the pressing issue of fake reviews. Fake reviews, whether planted by unscrupulous sellers seeking to inflate their products' ratings or by malicious actors attempting to tarnish competitors' reputations, can significantly distort the perception of product quality and influence purchasing decisions (Wu et al, 2019). The potential consequences are not only financial but also erode consumer trust in the authenticity of online reviews, impacting the overall credibility of e-commerce platforms.

Based on the domain and the issues at we have outlined two business objectives. The business objectives of this project are to use Amazon's customer reviews data to provide personalized product recommendations to increase customer satisfaction and to also provide a fake review detection system that will optimize the customer purchasing experience by reducing the influx of fake reviews. The data mining goals are to develop a model to classify fake or manipulated reviews within the dataset and to investigate the impact of incorporating review text, on the performance of the product recommendation system.

I will be utilizing Natural Language techniques such as sentiment analysis and Text representation and classification algorithms to achieve these objectives. I will outline the structured approach undertaken in this project, following the CRISP-DM framework. From data understanding to deployment, each phase will contribute to the overarching goal of providing Amazon with the tools necessary to address the challenge of declining customer satisfaction and trust.

Methodology

1. Data Understanding

The Kaggle Amazon US Customer Reviews and Amazon Product Review (Spam and Non-Spam) Datasets serve as a valuable repository of customer opinions and experiences regarding products available on Amazon.com. With over a hundred million reviews and over twenty-seven million reviews respectively, contributed by Amazon customers over two decades, these datasets have become an essential resource for academic researchers in fields such as Natural Language Processing (NLP), Information Retrieval (IR), and Machine Learning (ML). The datasets were curated to provide insights into customer evaluations, regional variations in product perception, potential promotional intent, or review bias and to classify reviews as spam or non-spam(ham) specifically on the latter dataset. Amazon US Customer Reviews dataset boasts 15 attributes as seen in **Table 1** and the Amazon Product Review (Spam and Non-Spam) Dataset has 12 attributes as seen in **Table 2**. Datasets have more than enough attributes to get insights that will help me fulfill my research goals and business goals at a high level. The datasets are in a tab delimited and json formats, respectively.

Table 1. List of attributes related to Amazon US Customer Reviews Dataset

Attribute Name	Variable Type	Description
Marketplace	Nominal	2 letter country code of the marketplace where the review was written
customer_id	Nominal	Random identifier that can be used to aggregate reviews written by a single author.
review_id	Nominal	The unique ID of the review.
product_id	Nominal	The unique Product ID the review pertains to. In the multilingual dataset the reviews for the same product in different countries can be grouped by the same product_id.
product_parent	Nominal	Random identifier that can be used to aggregate reviews for the same product.
product_title	Nominal	Title of the product.
product_category	Nominal	Broad product category that can be used to group reviews
star_rating	Ordinal	The 1-5 star rating of the review.

helpful_votes	Numeric	Number of helpful votes, meaning people who found the reviews helpful.
total_votes	Numeric	Number of total votes the review received.
Vine	Nominal	Review was written as part of the Vine program which enables a select group of Amazon customers to post opinions about new and pre-release items to help their fellow customers to make educated purchasing decisions
verified_purchase	Nominal	The review is on a verified purchase.
review_headline	Nominal	The title of the review.
review_body	Nominal	The review text.
review_date	Temporal	The date the review was written.

Table 2. List of attributes related to Amazon Product Review (Spam and Non-Spam)

Attribute Name	Variable Type	Description
_id	Nominal	The unique ID of the review.
reviewerID	Nominal	Random identifier that can be used to aggregate reviews written by a single author.
Asin	Nominal	The unique Product ID the review pertains to. In the multilingual dataset the reviews for the same product in different countries can be grouped by the same product_id.
reviewerName	Nominal	The name of the reviewer
Helpful	Nominal	Number of helpful votes, meaning people who found the reviews helpful.
reviewText	Nominal	The review text.
Overall	Nominal	The 1-5 star rating of the review.
Summary	Ordinal	The title of the review.

unixReviewTime	Numeric	Number of helpful votes, meaning people who found the reviews helpful.
reviewTime	Termporal	Number of total votes the review received.
Category	Nominal	Broad product category that can be used to group reviews
Class	Numeric	A categorical label indicating whether the review is classified as spam or non-spam

1.1. Data Licensing and Usage

Both datasets are derivatives of the Amazon Customer Reviews Library and are subject to Amazon's Condition of use. Users are granted a limited, non-exclusive, non-transferable, non-sublicensable, revocable license to access and use the dataset for academic research purposes. Users are prohibited from reselling, republishing, or making any commercial use of the dataset or its contents. The dataset should not be used for commercial research, consultancy contracts, internships, or other commercial purposes. Additionally, users should not attempt to link or associate content in the dataset with personal information, and they should not attempt to identify the authors of the content. Violating these conditions may result in the termination of the user's license to access and use the dataset. The license reinforces the intended use of the dataset for academic research while preventing commercial exploitation and safeguarding user privacy. The spam dataset is a product of Hussain et al (2020) research into Spam Review Detection using the Linguistic and Spammer Behavioral Methods.

1.2. Obtaining the data

Initially, I started off on a google colab notebook and used the Kaggle API to download the Amazon US Customer Reviews dataset, henceforth will be labelled as **dataset A**, and Amazon Product Review (Spam and Non-Spam) dataset, henceforth will be labelled as **dataset B**. In hindsight, the latter dataset was all that was needed but, in my defense, I was working with two research goals which I wasn't sure was sufficient before I added the latter dataset which had the spam or ham class needed in the classification of fake reviews, but I digress. After downloading the datasets to my google drive, I transferred them to google bucket that I had created for my compute cluster on the Google Cloud Platform. To initiate my exploration, I created a notebook and loaded the Kaggle dataset into the analysis environment, to review the structure and contents of the datasets. The preliminary examination aimed to gain a high-level understanding of the datasets' key attributes and dimensions. This allowed me to gauge the richness of the data and set the stage for deeper analysis.

1.3. Getting Preliminary insights

Each attribute within the dataset was subjected to meticulous analysis to extract its significance and potential utility. Attributes were dissected to comprehend their roles and relationships within the dataset. This step aimed to lay the groundwork for subsequent analysis and modeling.

I started out by getting the shape of the datasets as seen in Figure A and Figure B.

Figure 1: The number of rows and columns of the **dataset A**.

```
In [96]: shape(df)
         Shape: (1033354, 13)
```

Figure 2: The number of rows and columns of the **dataset B.**

```
In [11]: shape(spam_df)
         Shape: (1010113, 12)
```

I also retrieved the descriptive statistics from both datasets to identify potential issues in the data, such as missing values, unusual distributions, or extreme values thereby getting a feel of the data quality as seen in figures 3 and 4. I noticed that for dataset A the **star_rating** on average it was 4.15 stars which suggests that that most of the ratings are high. The **helpful_votes** and **total_votes** had a low average value with a huge standard deviation which suggest that there are outliers, especially considering the max values. For dataset B there was a mean of 0.78 which suggests that most of the reviews are spam.

Figure 3: Summary Statistics for dataset A

	summary	star_rating	helpful_votes	total_votes
0	count	1033354	1033354	1033354
1	mean	4.15050505441504	2.16458541796906	2.8719161100648956
2	stddev	1.305044384231812	21.947333910851263	23.990776655808123
3	min	1	0	0
4	max	5	993	99

Figure 4: Summary Statistics for dataset B

	summary	class	overall
0	count	1010113	1010113
1	mean	0.7775536004387628	4.1291528769553505
2	stddev	0.415889612897432	1.2905134894886963
3	min	0.0	1.0
4	max	1.0	5.0

I also used the printschema function available in pyspark to get a snapshot of the data types for each column for dataset A and B as seen in Figures 5 and 6. Based on the visualization of the contents of the dataset in Figures 5 and 6 compared to the actual data types, there may be some need for some data conversions.

Figure 5. The schema of dataset A

```
root
 |-- marketplace: string (nullable = true)
 |-- customer_id: string (nullable = true)
 |-- review_id: string (nullable = true)
 |-- product_id: string (nullable = true)
 |-- product_parent: string (nullable = true)
 |-- product_title: string (nullable = true)
 |-- product_category: string (nullable = true)
 |-- star_rating: string (nullable = true)
 |-- helpful_votes: string (nullable = true)
 |-- total_votes: string (nullable = true)
 |-- vine: string (nullable = true)
 |-- verified_purchase: string (nullable = true)
 |-- review_headline: string (nullable = true)
 |-- review_body: string (nullable = true)
 |-- review_date: string (nullable = true)
```

Figure 6. The schema of dataset B

```
root
 |-- _id: struct (nullable = true)
 |    |-- $oid: string (nullable = true)
 |-- asin: string (nullable = true)
 |-- category: string (nullable = true)
 |-- class: double (nullable = true)
 |-- helpful: array (nullable = true)
 |    |-- element: long (containsNull = true)
 |-- overall: double (nullable = true)
 |-- reviewText: string (nullable = true)
 |-- reviewTime: string (nullable = true)
 |-- reviewerID: string (nullable = true)
 |-- reviewerName: string (nullable = true)
 |-- summary: string (nullable = true)
 |-- unixReviewTime: long (nullable = true)
```

I also created a function to get a snapshot of the number of missing values in each column for dataset A and B as seen in Figures 7 and 8, and based on the visualization, there is probably less than 10% of the datasets with null values. I tried try to maintain as much data as possible since it would be used in the recommendation engine for dataset A. As for dataset B, I opted to drop the **reviewerName.**

Figure 7. The number of null values in each column for dataset A

```
Missing values:
marketplace: 0
customer_id: 0
review_id: 0
product_id: 0
product_parent: 0
product_title: 0
product_category: 9
star_rating: 9
helpful_votes: 9
total_votes: 9
vine: 9
verified_purchase: 9
review_headline: 69
review_body: 117
review_date: 73
```

Figure 8. The number of null values in each column for dataset B.

```
Missing values:
_id: 0
asin: 0
category: 0
class: 0
helpful: 0
overall: 0
reviewText: 0
reviewTime: 0
reviewerID: 0
reviewerName: 8066
summary: 0
unixReviewTime: 0
```

2. Data Preparation

2.1. Data Cleaning and Handling Missing Values

After getting the statistics and structure of the data. I started the data cleaning for dataset A and B. To handle rows with null data, I choose to drop rows where the last 9 columns were all null, as these records wouldn't provide meaningful insights. This enhances the data's quality by eliminating irrelevant data. I filtered out rows where the **product_category** was 2011-09-09 as this row was beyond saving and served as one of the victims of data misalignment. I casted **star_rating** as an integer aligns with its ordinal nature, as it represented a discrete and ordered variable.

I was very methodical in the handling of date-related columns. I tried to maintain the integrity of the data by transforming cases where **review_body** appeared as a date to **review_date**. Converting **review_date** to the appropriate date format consolidates temporal data and filling null dates with the most frequent date maintains data consistency. I dropped the **product_parent** and **marketplace** columns as both don't contribute to my analysis and marketplace specifically held limited variability as the entire dataset was based in the US region.

To supplement empty or null **review_body**, I passed the contents of the **review_headline** to it as while the **review_headline** was a title of the review, it was sufficient to serve as a **review_body** and I saw it as a summary.

2.2. Feature Creation

I created **season**, **month**, and **year** columns from **review_date** as I wanted to get some temporal insights. I also generated a **sentiment_score** column through sentiment analysis using the TextBlob library and I further constructed a **sentiment** column which served as the category being either Negative, Positive or Neutral. I also generated **abs_sentiment_score** column to mitigate negative value issues while creating the feature engineering pipelines and modelling. I also created **review_text_length** as another potential feature for the spam classification models.

2.3. Text preprocessing

To prepare the **review_text** and **review_body** fields for use during the modelling phase , a slew of text preprocessing steps was performed in the form of a pipeline. These steps included removing unicode characters, lowercase normalization, tokenization, and lemmatization through a custom transformer.

2.4. Exploratory Data Analysis dataset A

I performed further Exploratory data analysis on the cleaned datasets before proceeded into the modelling phase.

The number of unique product categories was found to be 41 with the top reviews being mostly distributed within digital related products as seen in Figure 9 and 10. This validates the initial deduction as we can clearly see these digital products are barely punctuated by other categories. This makes sense

as customers can access these products through their Amazon accounts or other online platforms, and the focus is on digital consumption rather than physical delivery.

Figure 9. The number of reviews per product category for dataset A.

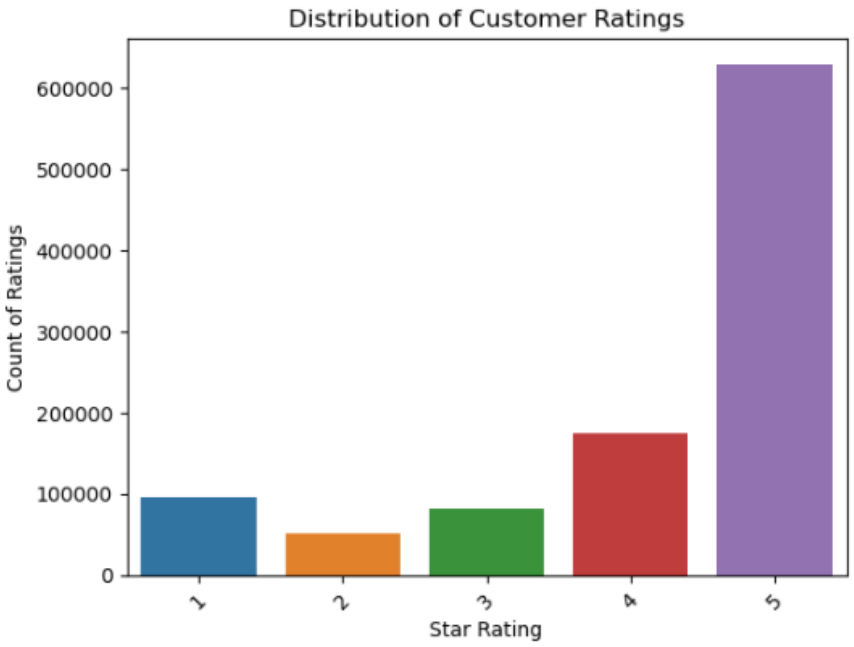
	product_category	number_of_Reviews
0	Mobile_Apps	36199
1	Digital_Ebook_Purchase	35532
2	Video DVD	34761
3	Digital_Video_Download	34560
4	Books	33567
5	Music	33557
6	Toys	30524
7	Digital_Music_Purchase	30514
8	Sports	30492
9	Beauty	30268
10	PC	30232
11	Wireless	30205
12	Camera	30194
13	Shoes	30186
14	Pet Products	30172
15	Office Products	30159
16	Tools	30124
17	Video	30113
18	Baby	30113
19	Health & Personal Care	30066

Figure 10. The number of reviews per product per product category for dataset A.

	product_id	product_category	product_title	number_of_Reviews
0	B004LLIKVU	Gift Card	Amazon.com eGift Cards	4387
1	B00H9A60O4	Digital_Software	Avast Free Antivirus 2015 [Download]	1431
2	B00NG7JVSQ	Digital_Software	TurboTax Deluxe Fed + Efile + State	967
3	B00A48G0D4	Gift Card	Amazon eGift Card - Happy Birthday (Candles)	907
4	B004RMK4BC	Digital_Video_Games	Playstation Network Card	781
5	B002VBWIP6	Digital_Video_Games	Xbox Live Subscription	727
6	BT00DDVMVQ	Gift Card	Amazon eGift Card - Smile	712
7	B00IX1I3G6	Gift Card	Amazon.com Gift Card Balance Reload	632
8	B00H9L7VIW	Personal_Care_Appliances	boostULTIMATE - 60 Capsules - Increase Workout...	588
9	B00FAPF5U0	Mobile_Apps	Candy Crush Saga	548
10	B004RMK5QG	Digital_Video_Games	Playstation Plus Subscription	537
11	B00GAC1D2G	Digital_Video_Games	Playstation Network Card	535
12	B004RMK4P8	Digital_Video_Games	Playstation Network Card	503
13	BT00CTOUNS	Gift Card	Amazon.com Gift Card in a Greeting Card (Vario...	497
14	BT00DDC7BK	Gift Card	Amazon.com Gift Cards - Print at Home	479
15	B007VTVRFA	Digital_Video_Games	SimCity - Limited Edition	450
16	BT00DDC7CE	Gift Card	Amazon.com Gift Cards - Print at Home	391
17	B00E8KLWB4	Mobile_Apps	The Secret Society® - Hidden Mystery	388
18	B004LLIKY2	Gift Card	Amazon eGift Card - Amazon Kindle	376
19	B004VSTQ2A	Digital_Video_Games	Xbox Live Subscription	355

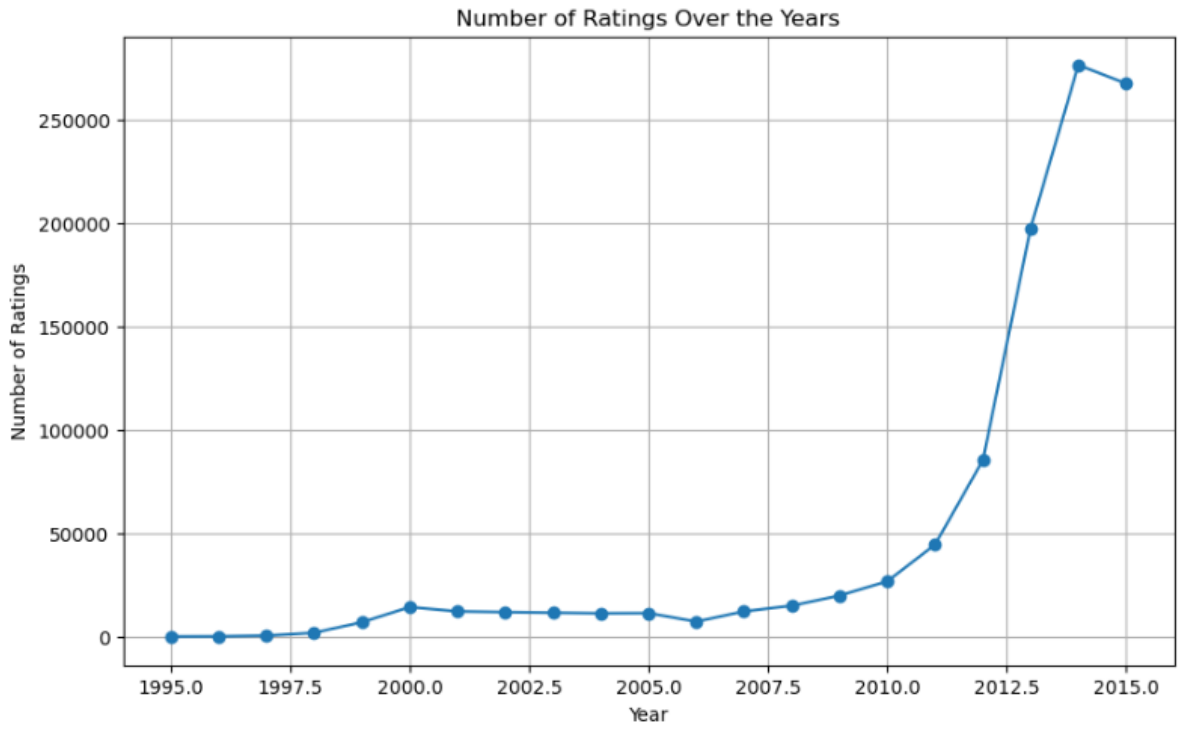
I also reviewed the distribution of star rating and found that there was high frequency of 5-star ratings with the 4-star ratings coming in at a close second and 1–3-star ratings being comparatively lower as seen in figure 11. This indicates a pattern of high customer satisfaction and positive feedback for the products or services on initial analysis.

Figure 11. The distribution of customer ratings for dataset A



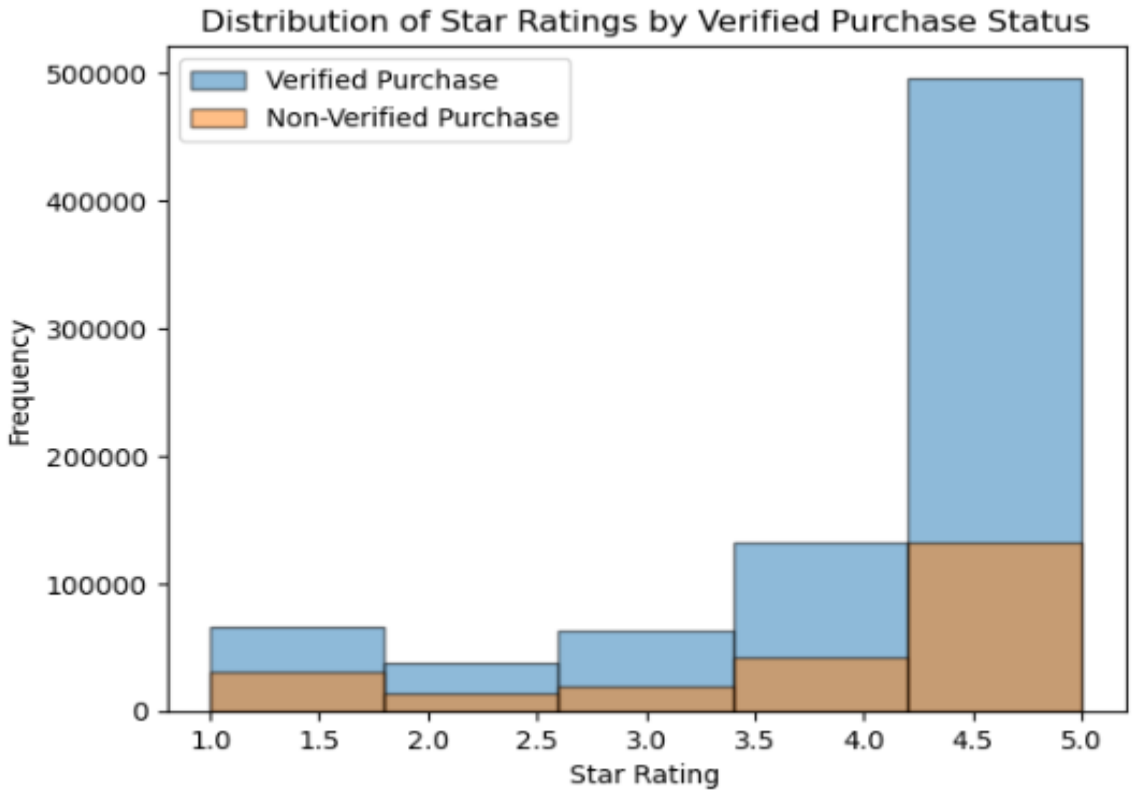
The distribution of the cumulative ratings was tracked over the 20 years and there was a steady growth of the number of ratings being given to products as seen in figure 12. We see a slight increase between 1995 and the year 2000 with a year plateau between 2000 and 2005. There is also an exponential increase in ratings which can be attributed to the garnering of more customers, due to the expansion of the amazon marketplace and general business model.

Figure 12. The number of star ratings over the years for dataset A



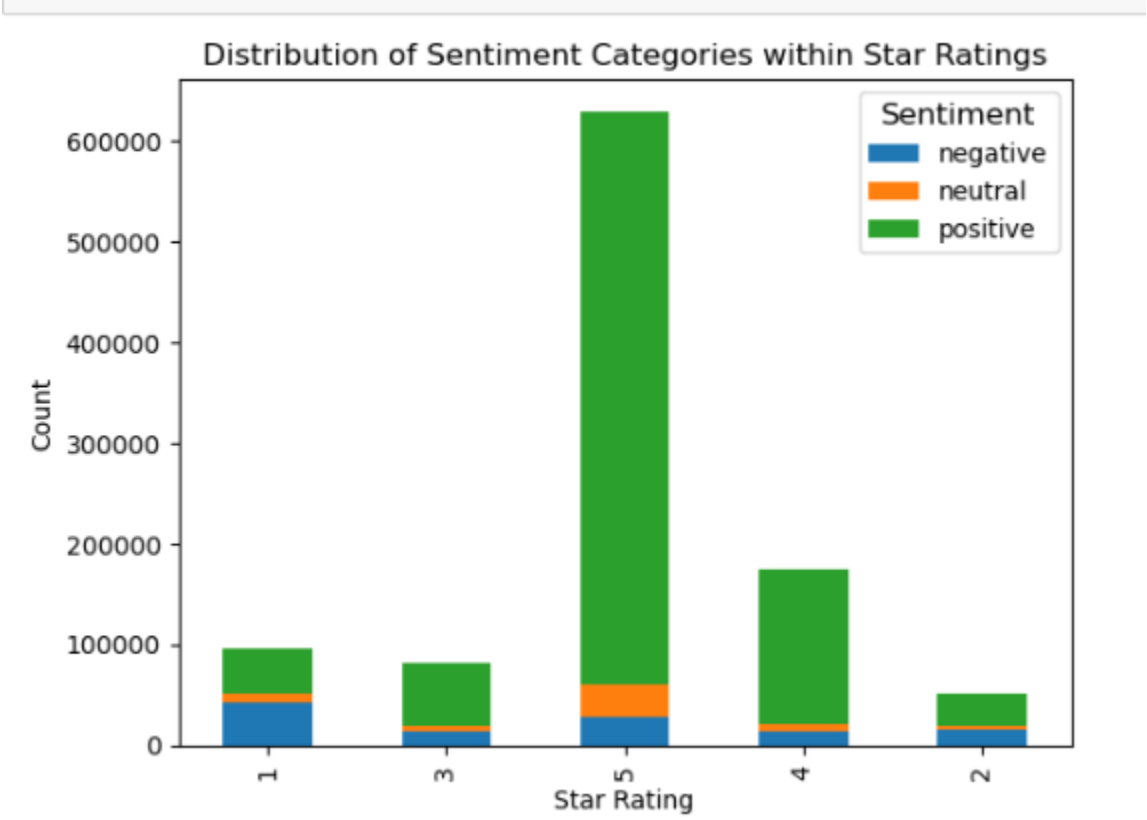
I also wanted to analyze the distribution of ratings as it pertains to whether the product was rated by a customer with verified purchase. The result was that more that 60% of the 5 star and 4-star ratings were done by customers who were denoted as having had a verified purchase as seen in figure 13.

Figure 13. The distribution of star ratings by verified purchase status for dataset A



It also appears that most of the ratings are how a lot of positive sentiment, especially as it pertains to the 5 star reviews as seen in figure 14. This gives credence to the fact that the distribution of ratings were primarily within 5 stars. This means there is no deviance between a high star rating and the sentiments being expressed in the reviews.

Figure 14. The distribution of sentiment categories within star ratings for dataset A

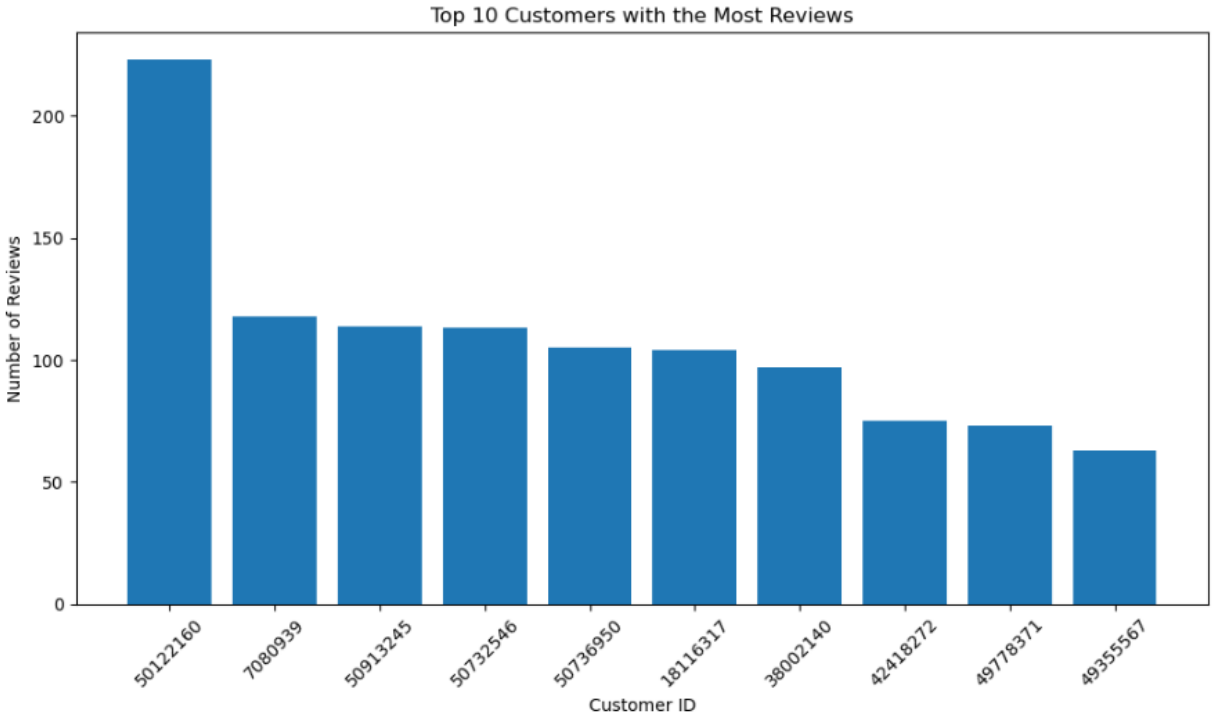


In figure 15, I analyzed how the reviews were distributed along the purchase verification attribute and there was a three to one ratio of reviews with verified purchases to review with unverified purchases in favor of reviews with verified purchases.

Figure 15. The number of reviews by purchase verification for dataset A

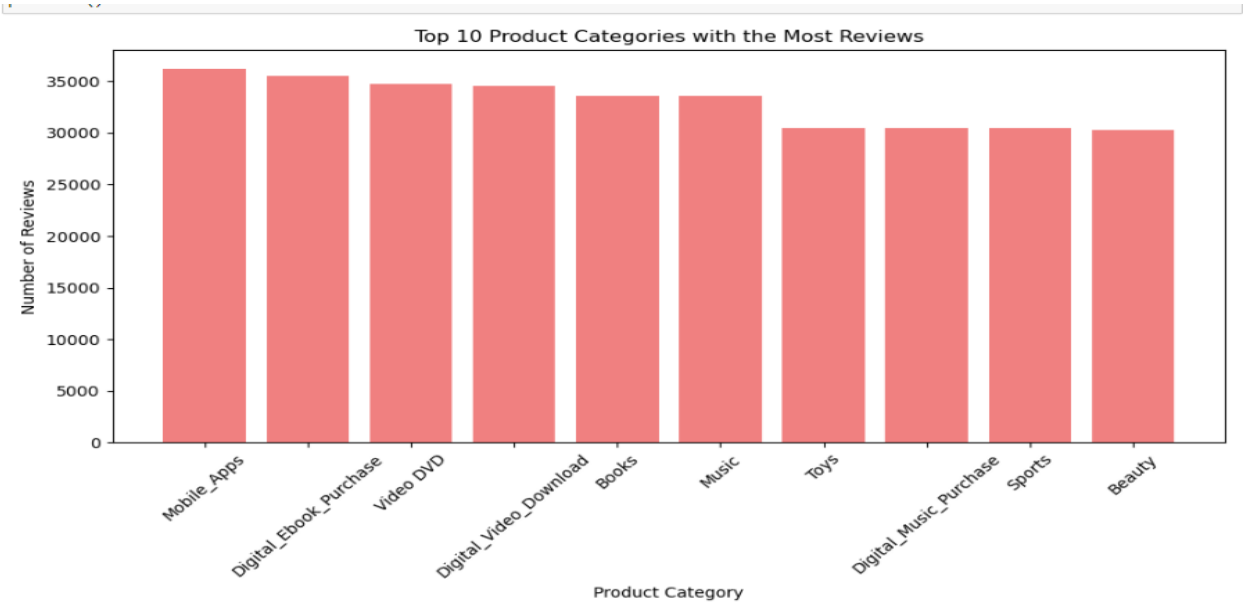
I also wanted to assess the top reviewers to check how many reviews were completed by each and I noticed that there were over 200 reviews being done by customer with id 50122160 as seen in figure 16 which may see this customer having a lot of product interactions.

Figure 16. The top 10 customers with the most reviews for dataset A



I also reviewed the top 10 product categories with the most reviews and found that majority of the apps were distributed along digital products as seen in figure 17.

Figure 17. The top 10 product categories with the most reviews for dataset A



In figure 18, I noticed that there were many reviews being categorized as helpful. I also noticed that in figure 19, the number of reviews climbed exponentially after the year 2011 which may be due to amazon's ongoing innovation within the space of ecommerce. On analyzing the word cloud in figure 20, I am interpreting the main words as being mostly clothing related.

Figure 18. The number of helpful vs unhelpful reviews for dataset A

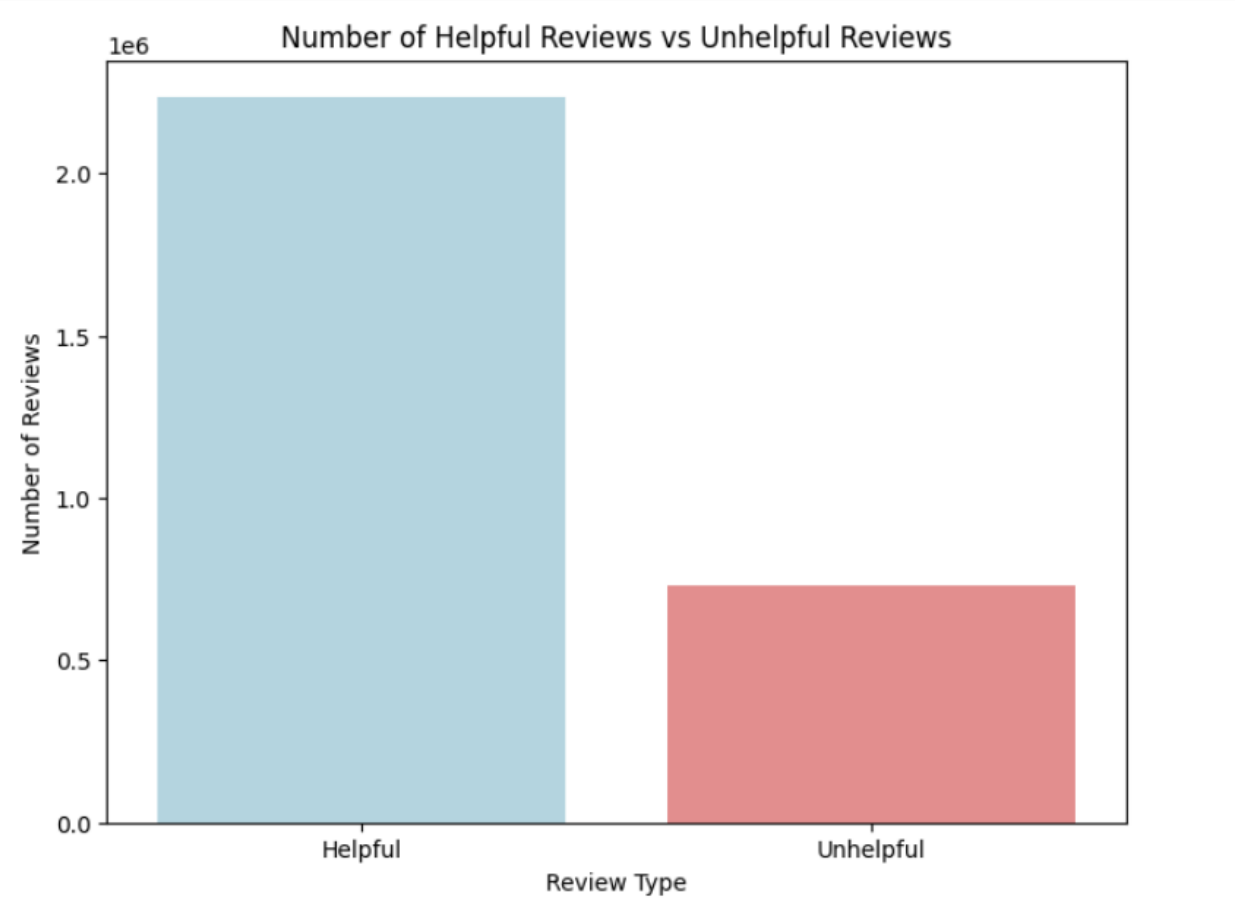


Figure 19. The number of null values in each column for dataset A

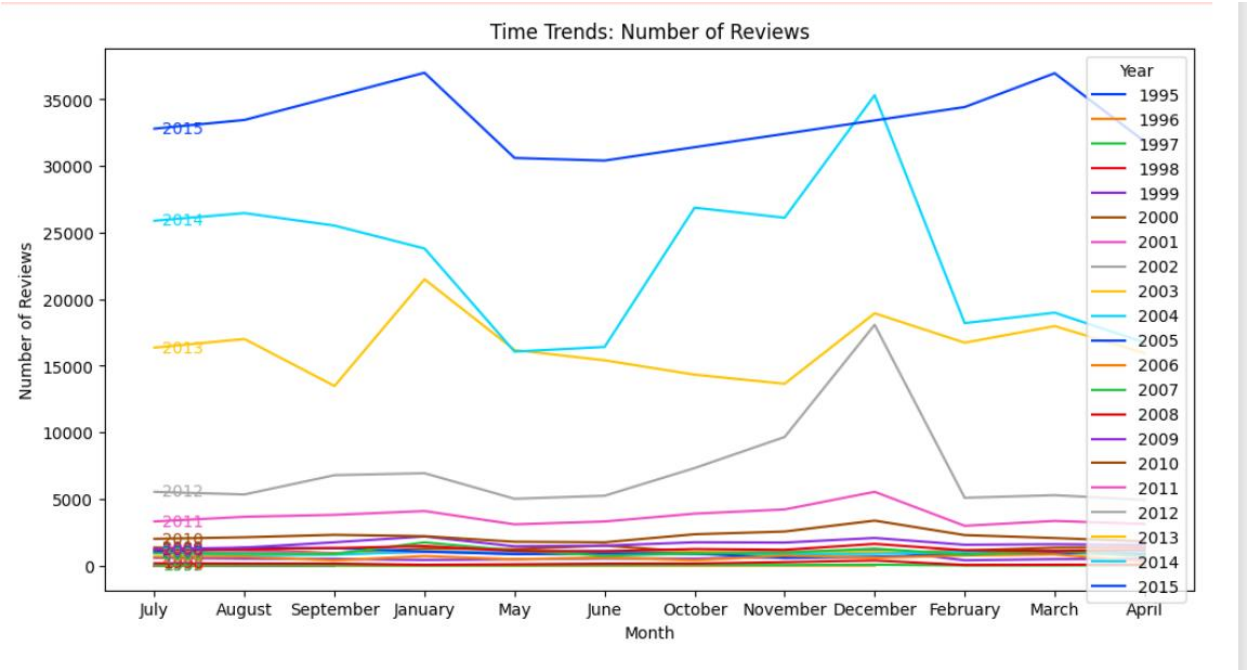
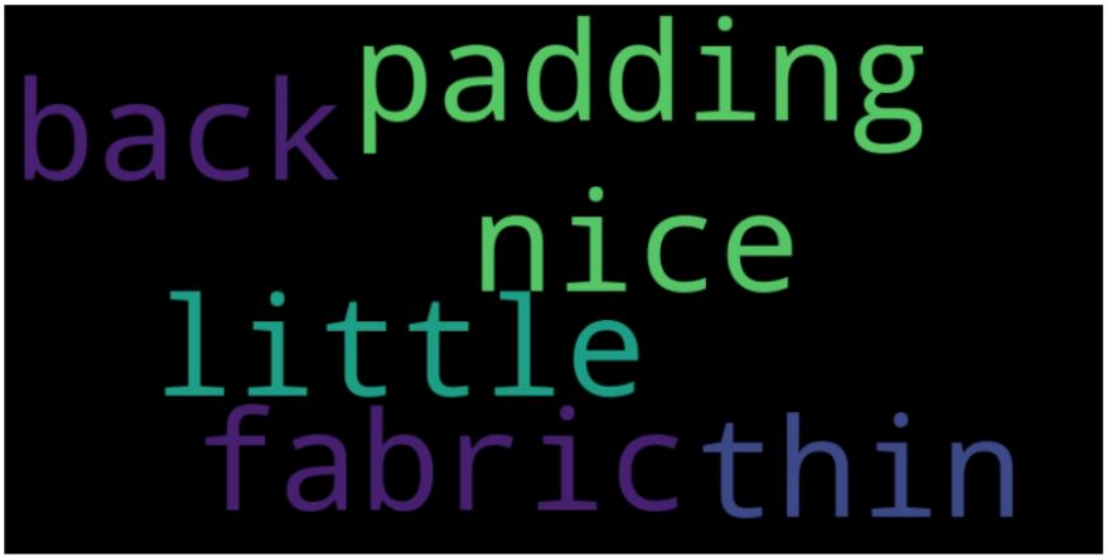
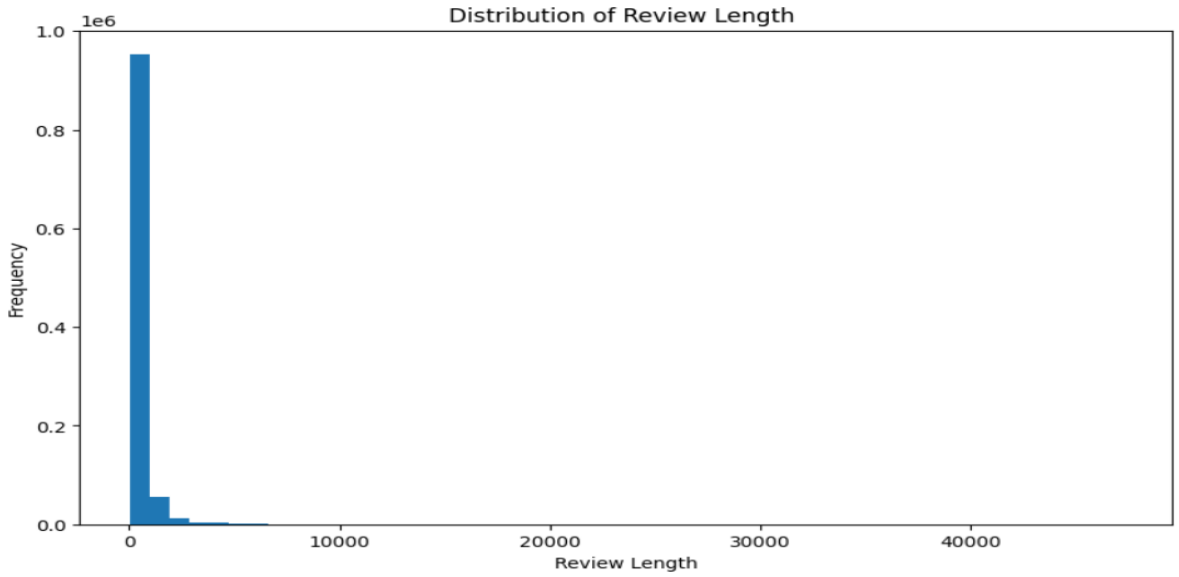


Figure 20. The number of null values in each column for dataset A



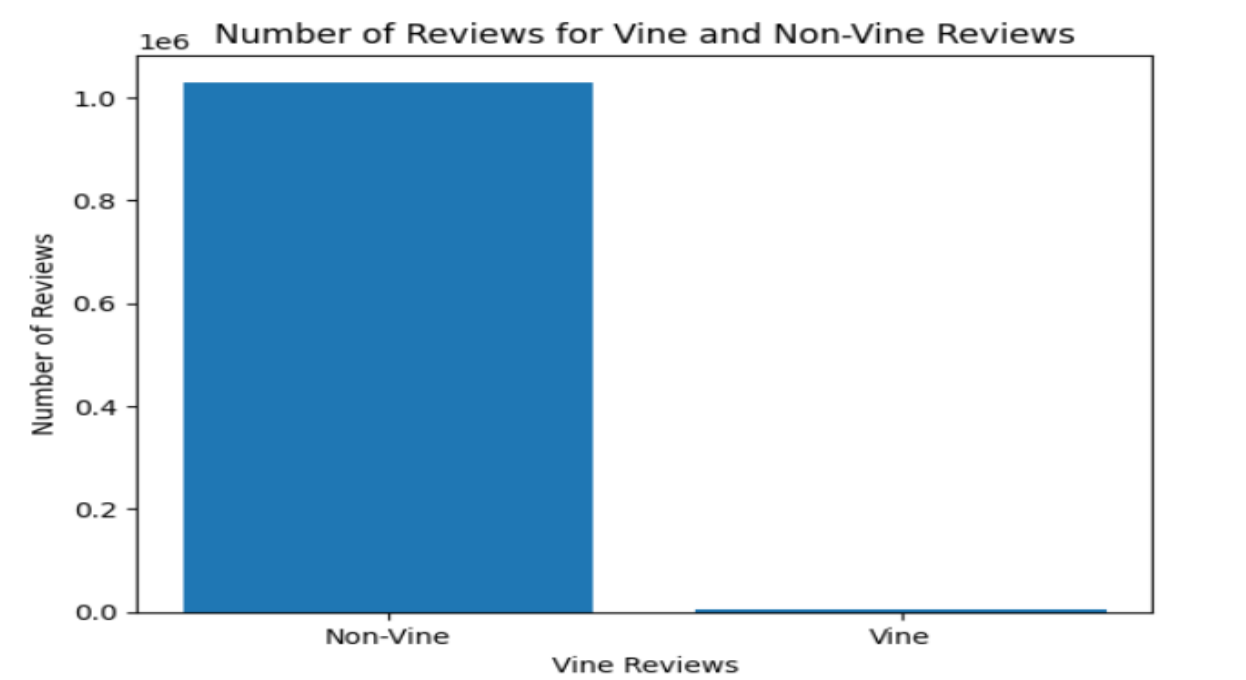
On reviewing the distribution of review length, I noticed that it is disproportionately between 0 and 625 characters as seen in figure 21.

Figure 21. The distribution of review length for dataset A



With the adoption of the Amazon vine program in 2019, it makes sense that the number of reviews for non-vine members is significantly higher as seen in figure 22.

Figure 22. The number of null values in each column for dataset A



2.5 Exploratory Data Analysis for dataset B

For Figures 23 and 24, it seems like most of the review text lengths hover around the 0-625 count with most of the review lengths disproportionately aligned with the spam reviews.

Figure 23. The distribution of Review Text Length for dataset B.

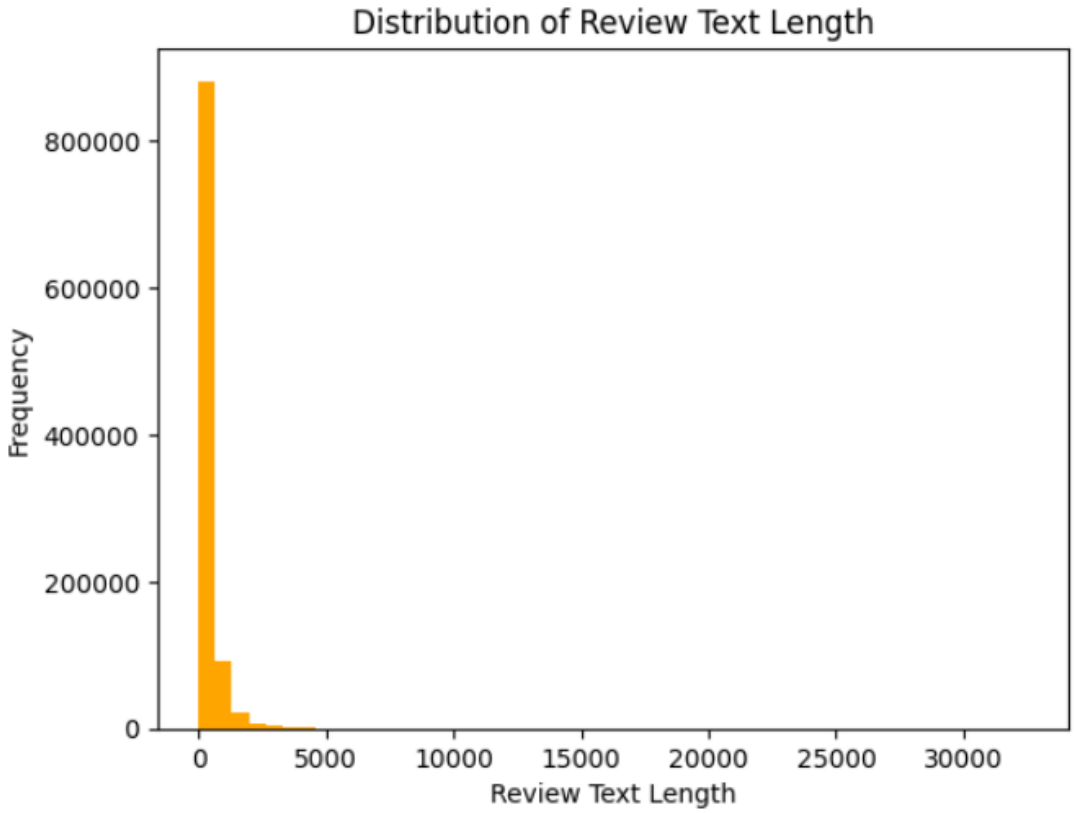
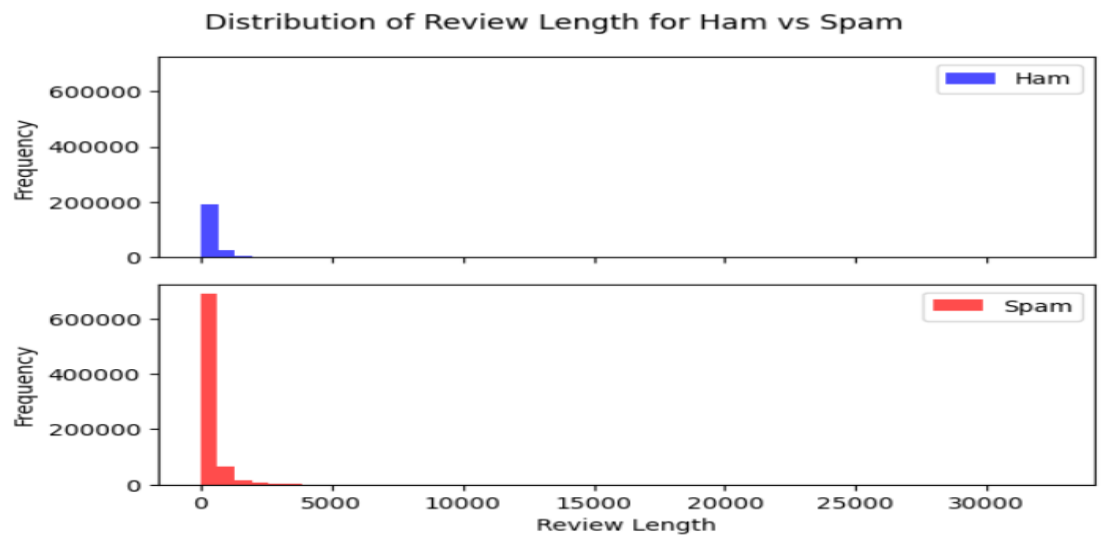


Figure 24. The distribution of Review Text Length for Ham vs Spam dataset B.



I noticed in figure 25, the grow of spam reviews has towered over ham reviews over the years which really gives credence to the fact that it is a significant problem for ecommerce especially in this case of Amazon. On assessment of the average length of spam vs ham reviews, I have noticed that the length of ham reviews in figure 26 are higher on average. There might be a case where since the spam reviews are bought, to maximize efficiency, they are shorter in nature.

Figure 25. The growth of the number of spam vs ham reviews for dataset B.

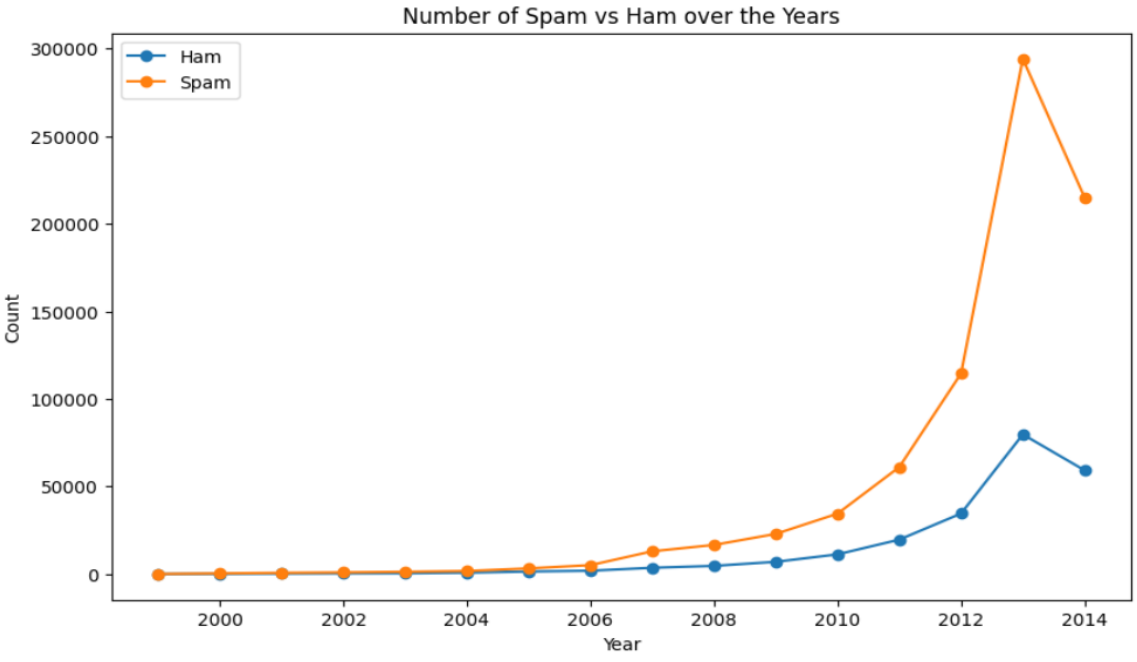
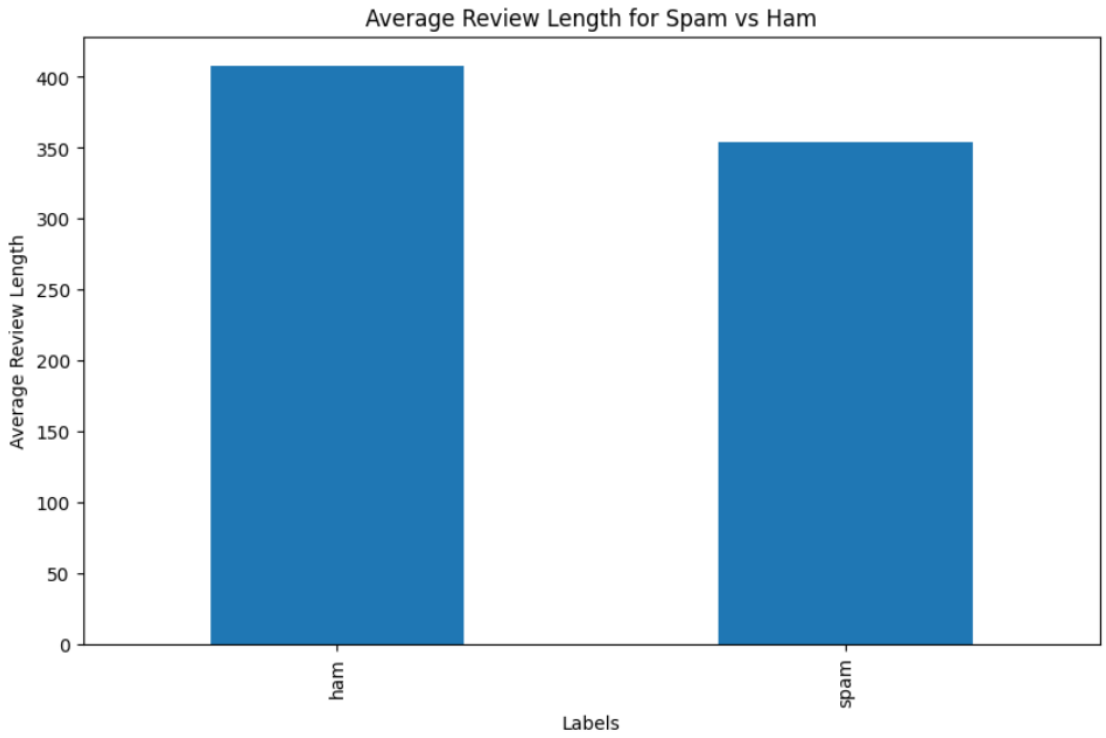


Figure 26. The average length of ham vs spam reviews



In Figure 27, an initial review of the word clouds for spam and ham reveals that the main words for spam are words of affirmation while the ham words are mostly a mixed bag of unrelated words.

Figure 27. Word Clouds for Spam and Ham Reviews for dataset B

In analyzing the bigrams and trigrams in Figures 30, 31,32 and 33. I noticed that the bigrams and trigrams for spam reviews is mostly positive which shows me that the review brokers are in full effect. I notice that effectively all the top 20 bigrams and trigrams for spam review are positive compared to the ham reviews which are mixed with positive and negative.

Figure 28. Top 20 Bigrams for Ham Reviews for dataset B

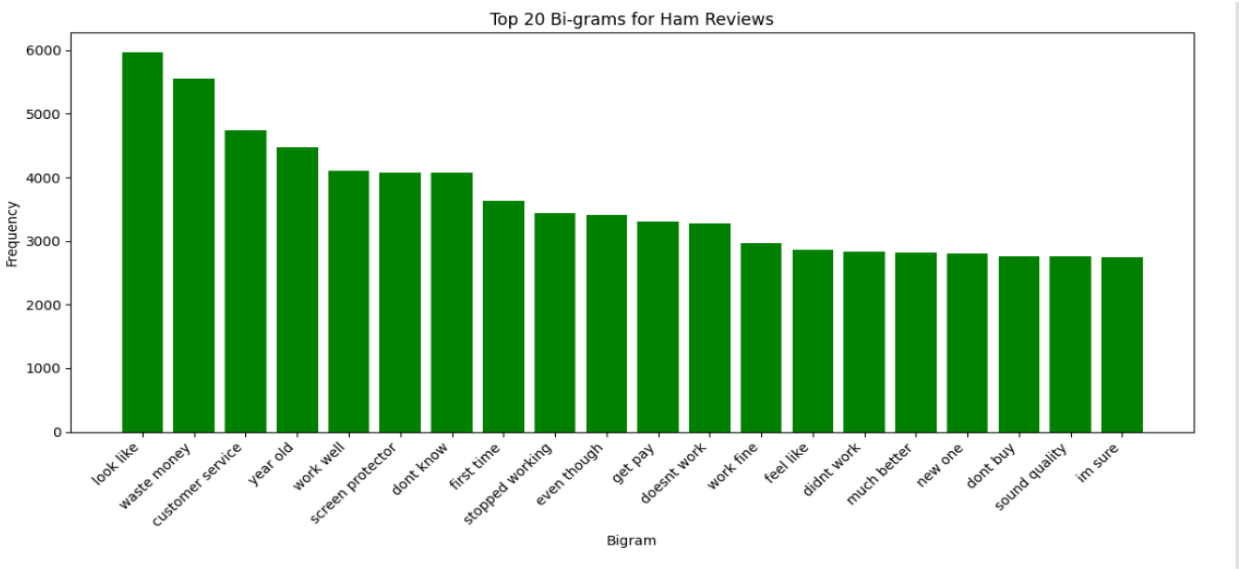


Figure 29. Top 20 Bigrams for Spam Reviews for dataset B

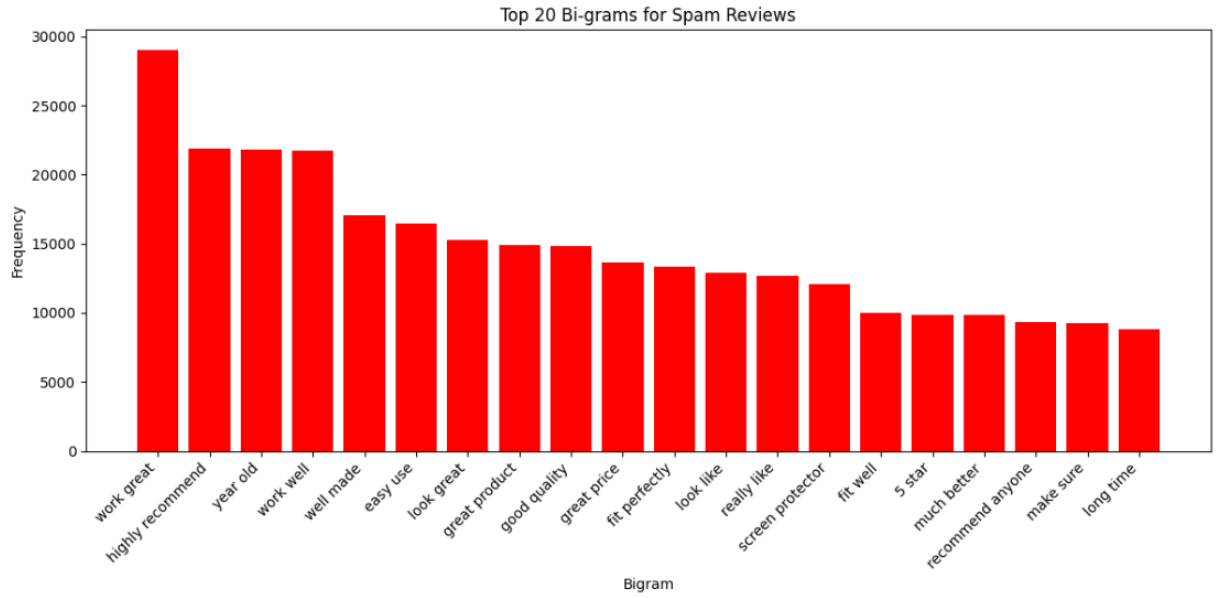


Figure 30. Top 20 Trigrams for Ham Reviews for dataset B

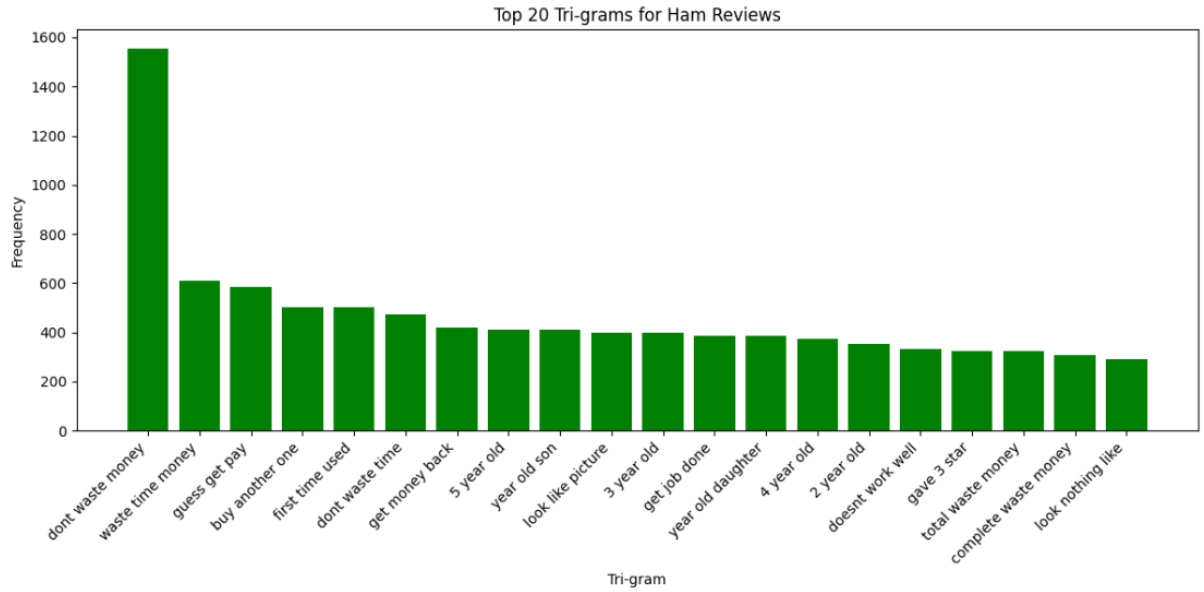
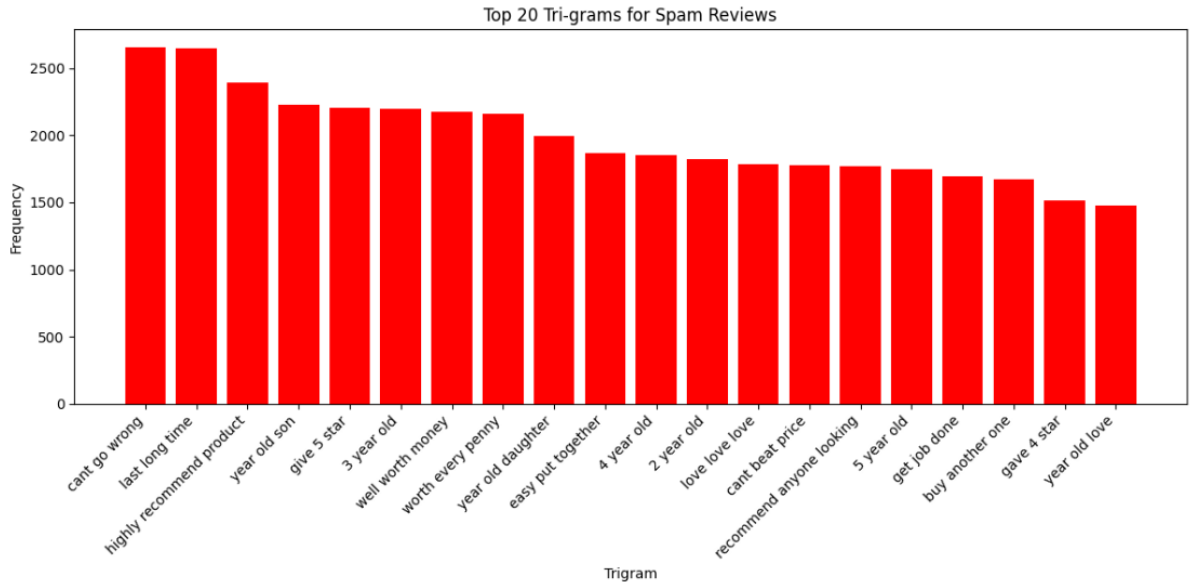


Figure 31. Top 20 Trigrams for Spam Reviews for dataset B



2.6 Feature Selection

For the clustering algorithm-based product recommender I choose to use **lemmas** and **abs_sentiment_score** (sentiment intensity) fields as I the thought if could cluster the products by those features, in terms of dataset A, I could provide meaningful recommendations through Content-Based Filtering. I chose **abs_sentiment_score** as a feature because sentiment analysis systems have the potential to enhance various types of recommender systems, including simple, aspect-based, and end-to-end deep models (Barriere & Kembellec, 2018). I chose to get the term frequency inverse document frequency of the **review_body** because of the use of the feature in a similar Netflix movie recommendation system proposed by Chiny et al (2014) that used TF-IDF and cosine similarity. I then grouped the dataset by the **product_id** and took the average **abs_sentiment_score,** producing a new column called **avg_ abs_sentiment_score. In the aggregation, I also** collected all the **lemmas** related to that grouping, producing a new column called **combined_lemmas**. This was to create a more stable and representative recommender and avoid cases where a product was would have been a part of its own recommendation.

For the alternating least-square algorithm I chose to group the data in dataset A by **the customer_id** and **product_id** and average the **star_rating** producing a new column **avg_star_rating**. I did this because as it existed the dataset had duplicate **customer_id** and **product_id** columns already based on the nature of the dataset. In doing so, we produce accurate representation of the customer-product interations.

In case of dataset B with the Fake Reviews Classifier, I used features like **lemmas**, **review_text_length** and the **abs_sentiment_score** column. My reasoning for choosing **review_length** as a feature is because the average length of reviews can serve as a significant indicator of potential questionable intentions among reviewers. Notably, approximately 80% of spammers exhibit a lack of reviews exceeding 135 words in length. In contrast, over 92% of trustworthy reviewers demonstrate an average review length exceeding 200 words. This disparity underscores the potential utility of review length as a distinguishing factor between spammers and reliable reviewers (Crawford et al,2015). I selected the **abs_sentiment_score** (sentiment strength) as a feature because sentiment strength proves to be a more effective indicator compared to rating scores in the context of identifying spam reviews (Peng and Zhong, 2014). It is noted by Sjarif et al (2019) that incorporating the term frequency inverse document frequency of the **review_text**, spam detection systems can effectively differentiate between relevant and generic words, optimizing the accuracy of the detection process. Moreover, TF-IDF extends its impact beyond individual documents to evaluate the broader importance of words across the entire corpus, thus enhancing the precision of spam detection.

2.7. Feature Transformation

For the Fake Reviews Detection features, using dataset B I created a Term Frequency Inverse document frequency of the tokenized and cleaned review text (**lemmas** field) using CountVecorizer class and IDF class. Then using the StandardScaler class, I normalized the tf-idf feature matrix, **review_text_length** and **abs_sentiment_score** features to prevent the models from favoring a specific feature. I then used Principal Component Analysis to reduce the dimensionality to avoid the curse of dimensionality.

For the clustering algorithm-based product recommender, I did something similar., I created a Term Frequency Inverse document frequency of the tokenized and cleaned review body using CountVecorizer class and IDF class. Then using the StandardScaler class, I normalized the tf-idf feature matrix and

abs_sentiment_score features to prevent the models from favoring a specific feature. I then used Principal Component Analysis to reduce the dimensionality to avoid the curse of dimensionality.

3. Modelling

According to a Kontsewaya et al (2020) machine learning techniques offer the highest level of accuracy when it comes to classifying spam especially as it pertains to the six most popular: Naive Bayes, SVM, Decision tree, K-Nearest Neighbors, Logistic regression, Random Forest. I chose to use the first three algorithms due to a time constraint.

The Naive Bayes algorithm, known for its probabilistic approach, effectively classifies spam. Its "naive" designation arises from its disregard for potential interdependencies or associations among inputs, simplifying a multivariate issue into a series of univariate problems (Sinha and Singh,2020).

According to Sajedi et al (2016), the Support Vector Machine (SVM) functions as a linear classifier by identifying the hyperplane that maximizes the separation between classes. Subsequently, new instances are projected into this space, and their categorization is determined based on their position relative to the gap between classes. The classifier aims to enhance the spacing between points to establish heightened "confidence" in class distinction. Remarkably, the model demonstrates resilience to outliers.

The decision tree algorithm can be simplified as being a hierarchical structure used for making decisions or predictions in various fields.

The python MLLIB, doesn't currently contain the K-nearest Neighbor algorithm implementation so to maximize the use of the large-scale distributed environment provided by pyspark, I opted out of using the algorithm. I could have converted the dataset to a pandas data frame to use K-Nearest Neighbor but it uses a lot of processing power during the conversion which isn't viable.

I wanted to implement both collaborative filtering and content-based filtering to create a personalized product recommendation engine, so I used the Alternating Least Squares (ALS) matrix factorization algorithm provided by Pyspark MLLIB in the implementation of collaborative filtering and K-means and Hierarchical Clustering for the implementation of Content-Based Filtering. The primary focus of the collaborative filtering system revolves around identifying similarities between customers' preferences and items. Recommendations for new users are generated based on the preferences of similar individuals from their browsing history. Collaborative filtering involves combining items, identifying similarities through user ratings, and creating new recommendations by comparing across multiple users (Gosh et al, 2021).

Content-based filtering proposes recommendations to users that closely resemble the items they have previously selected or shown interest in (Nallamala et al 2020). Iliopoulou et al (2020) explored Content-based filtering by employing K-Means clustering methods to center on uncovering similarities within movie plots. Their initial strategy involved grouping movies using the Tf/Idf weighting scheme, assigning significance to terms within movie plots. I utilized a similar concept with the products except I went a bit further by finding the cosine similarity of the products within the same cluster and choosing the products with the highest cosine similarity as recommendations.

4. Evaluation

In this phase, I subjected the trained models to rigorous testing, leveraging key evaluation metrics based on the specific research objectives mentioned in the business objectives to ascertain the best performing one. For the fake review classification, which is a binary classification problem, I used the area under the Precision Recall Curve (areaunderPR) and area under the Receiver Operating Characteristic Curve(areaUnderRoC) which are the only metrics provided by BinaryClassificationEvaluator from Pyspark's MLlib. To optimize the performance of each algorithm, I embarked on the crucial task of hyperparameter tuning using the TrainValidationSplit technique which is also provided by Pyspark's MLlib. It involves splitting the data into training and validation sets, training models on different hyperparameter settings, and evaluating their performance on the validation set. I employed the ParamGridBuilder to create a list of values for each parameter for each classification model.

For fake review detection, I found that based on the list of classification models, which performed well overall, and the resulting metrics, the best performing model was the Supoort Vector Machines model as seen in Table 3 which had an areaUnderPR value of **0.95** and an areaUnderROC value of **0.86**.

Table 3. The performance of the classification models

Models	areaUnderROC	areaUnderPR
Naive Bayes	0.56	0.81
Support Vector Machines	0.86	0.95
Decision Trees	0.70	0.88

With the use of SVM model to predict the ham vs spam reviews we get the results as seen in figures 32 and 33.

Figure 32. Showing identified spam reviews

	reviewText	prediction
0	😍😍😍😍😍 love love love haven't gotten a chance to wear them yet but can't wait!!!! A little small but I manage to make them fit by sliding my feet to the front !! They are so adorable on my feet . I ordered a 8.5 I wear an 8 in closed toe and 8.5 in open toe but theses ones were a little small !! Love them though !!❤️❤️❤️	1.0
1	صانع الروغة , تجربة شراء جيدة , القطعة جميله نوعا ما وتصنغ لك رغوة رائعة من الصابون السائل , تجربة لابائس بها	1.0
2	(...)Here are the details:Access your music without taking your player out of the bag! The convenient pocket clip secures your MP3 player to your belt, purse or backpack. The FM Wired Remote combines FM tuner capabilities, a built-in microphone, a cool blue backlit LCD, as well as the playback controls. Listen to or even record your favorite FM stations!Features Include:Blue Luminescent Backlit LCDBuilt-in Microphone for Voice RecordingRadio Recording in Stereo32 FM Preset ChannelsRadio Frequency AutoscanText Display for Track informationBattery and EAX IndicatorElapsed Time CounterClock! hope this helps.	1.0
3	(...)Since I have installed various Wi-Fi Routers and Access Points for a number clients the Router was fairly easy to configure and set up WPA security. This Router is cerainly not the easiest product to set up.The router drops wireless connections and the only way to reconnect is to reboot the router. This is true for Netgear PC Card with Super G as well as other Wi-Fi vendors.	1.0
4	(Note: review copied from other horse in collection) My 4 year old daughter loves horses, so if it's shaped like a horse I can't go wrong! (Basically she is going to like it regardless if it's worth the money I paid!) I liked the additional fact that the horse's head and leg moves and the realistic sounds. Doll sits okay on the horse...it's made so that the legs are moveable in most directions and jointed (unlike the Barbie that comes with a horse and legs aren't too easy to bend apart to it get to stay on)I also have the other two colors for her in the collection...using this review for both purchased on Amazon. Pretty sturdy, made it through several drops so far and of course she loves them. Lily gray and her rider are smaller than Aspen Gold or Sugar. Saddle and blanket are sewn together but removeable from horse. A little pricy because they are hard to find.	1.0

Figure 33. Showing identified spam reviews

	reviewText	prediction
0	+ POINTS:== Good product overall.== Very very easy to assemble.== It was delivered in like a week.== Comfortable to sit.- POINTS:== Front legs are just too weak.== It makes noise.== Uncomfortable to sleep.03/2013== UPDATE ==Both legs broke within 6 months, I am just 76Kg.. which should be around 140lb.I had to put some wood blocks below it.. Beside the weak legs it's a good product for the price.	0.0
1	... it happens.Mine would link up with the three different items I paired it to, but I couldn't get any audio from it. even tried another type of dock. I also set my iPod into the main dock on the Onkyo HT's RI Dock to be sure my settings were still set right and audio came blasting through that way. But not from this unit. Guess I got a bad one.Decided to go a different route with a better name brand. Hope that one isn't broke. I have that kind of luck sometimes.	0.0
2	..as I'd buy this in every color.Strange thing, though, regarding the style number. I ordered 3 Glamorise bras in one order just recently. 2 of them fit and 1 did not. In my order history and on my invoice, the description for this bra is "Magic Lift Front Close Posture Support Bra". The style number on the tag is 1275.I also ordered Style 1265, which is NOT this one. Yet the description matches this one-"Soft Shoulders Comfort Back Support Bra". That one, which says style 1265, is too big in the band and the cup size.	0.0
3	.Battery life is rediculous wicked bad. Laughable really. Othef than that its a good pphone. Cant belive how bad it is	0.0
4	1. I love the mats. They are incredible and feel great when working in the kitchen. I highly recommend them. I'd give the product five stars.2. I ordered Cinnamon mats. The boxes said they were Cinnamon, but they were dark brown, not the orangish brown ones I ordered. It's pretty clear that to have two mislabeled boxes means someone substituted the dark brown mats on purpose.I wasn't amused. Although I didn't return them or complain to the Amazon, I wouldn't buy this from Amazon again.Either the vendor supplying them to Amazon substituted the wrong color in the Cinnamon boxes or someone at the Amazon warehouse took it upon themselves to make the substitution prior to shipping. I don't know which, but either way, I was not satisfied with my order because of it.I'm only giving three stars because of the color change.	0.0

For the content-based recommender using the clustering-based approach with Kmeans and Hierachial Clustering, the Kmeans clustering model performed the best with a silhouette score of 0.993 as seen in Table 4.

Table 4. The performance of the clustering models

Models	Silhouette Score
Kmeans Clustering	0.993
Hierarchical Clustering	0.992

With the implementation of the kmeans clustering model with the cosing similarity for products within the same cluster as the target product we get a list of decent products as seen in figure 34.

Figure 34. Results Kmeans recommender with cosine similarity

```
+----------+--------------------------------------------------------------------------------------------------------+
|product_id|product_title                                                                                           |
+----------+--------------------------------------------------------------------------------------------------------+
|B001JTX8FK|Chakra Healing: Guided Meditation and Creative Visualization                                            |
|B004HQLV9C|Crate Appeal Dog Crate Black Repl Tray                                                                  |
|B000V2UBS8|Polk Audio RTI A1 Bookshelf Speakers                                                                    |
|B000BNWZXQ|BlueBonnet Pycnogenol Vegetarian Capsules, 100 mg, 60 Count                                             |
|B000H86BQ2|Camp Chef Carry Bag for BB90L #BB90BAG                                                                  |
|B00GBL7J8I|6 Assorted Large Christmas Gift Bag Selection, Gift Tissue Paper and our Exclusive Christmas Reusable Bag|
|B00H2JEEYU|Flash Furniture Foldable Tabletop Lectern in Mahogany                                                   |
|B007VAONDM|Peaceful Doves - Set of 2                                                                               |
|B00ES3UBJU|Angry Birds Star Wars Yoda 12" Plush Pillow by Rovio                                                    |
|B00K6PNEVQ|Back to Back World War Champs Flag Tank Top                                                             |
+----------+--------------------------------------------------------------------------------------------------------+
```

The collaborative filtering model using the alternating least squares algorithm had an excellent performance with a root mean square error(rmse) of **1.15**. On reviewing the results of figure 35, user 50122160 doesn't struggle with the cold start problem as he has the most products interactions as seen in Figure 16.

Figure 35. Showing the recommended products for user 50122160

Out[37]:

	customer_id	product_id	product_title
0	52582258	0671667513	WHO'S AFRAID OF CLASSICAL MUSIC? : A highly arbitrary and thoroughly opinionated guide to listening to and enjoying symphony, opera and chamber music
1	49535233	B004P8JL7G	Nokia C5-03 Unlocked GSM Phone with Symbian OS, 5MP Camera, Ovi Maps Navigation, Wi-Fi and microSDHC Slot
2	52797327	0312192479	Christmas With Rosamunde Pilcher
3	15880879	B00N16TDY4	Gates Bar-B-Q Sauce Original Classic - 40 Oz. Bottle (2 Pack)
4	23949432	B000ELVUCC	Country Life Arctic Kelp 225 mcg, 300-Count

5. Deployment

I implemented web-based recommendation systems (Kmeans Content-Based Filtering and ALS Collaborative filtering) and Spam classifier using Flask, PostgreSQL, and PySpark. The objective was to

provide users with personalized recommendations based on their preferences and allow users to predict if a review is classified as a spam or ham. Here's a coherent description of the deployment process, along with the reasons for each step taken:

5.1 Platform and Hardware Selection

To host the recommendation system, I chose to deploy it on a Google Cloud Platform (GCP) virtual machine (VM) instance. The instance was configured with an e2 CPU platform, offering good performance and scalability. I opted for a standard machine type with 4 vCPUs (2 cores) to ensure sufficient computational resources for running PySpark and handling web traffic.

5.2 Access Scope and Firewall Configuration

I granted the VM instance full access to all Cloud APIs. This allowed the application to interact with various GCP services and resources seamlessly. Additionally, I configured the firewall to permit incoming HTTP traffic, enabling users to access the web application. I also set up a specific firewall rule to allow incoming traffic on port 8080, which I designated for running the Flask application.

5.3 Environment Setup

To prepare the VM environment, several tools and dependencies needed to be installed:

- **Java Development Kit (JDK) and Hadoop**: Installed these components to support the execution of PySpark. PySpark leverages Hadoop's distributed processing capabilities for efficient data analysis.
- **Git**: Used to clone the repository containing the web application code onto the VM.
- **Python and Pip**: Installed Python along with Pip to manage and install Python packages required by the application.
- **Environment Variables**: I configured environment variables for Java and Hadoop. This step was essential to ensure that the VM could locate and utilize these components during the execution of PySpark tasks.
- **Database Setup**: PostgreSQL was selected as the database management system to store reviews and ratings data. To set up the database and tables, I executed the dbscript.py script. This script, located within the repository, created the necessary schema and tables for storing and retrieving data.
- **Flask Application Launch**: The heart of the recommendation system was the Flask-based web application. To initiate the application, I executed the app.py file. Flask provided a reliable and efficient framework for building the web interface that users would interact with.
- **Accessing the Web Application**: With the Flask app running, users could access the recommendation system by navigating to the external IP address of the VM at port 8080. This URL led users to the web interface where they could input their preferences and receive personalized recommendations.

5.4 How Does the Application Work?

Both the K-means Recommendation engine and the ALS recommendation engine utilize Select2.js which gives you a customizable select box with support for searching, tagging, remote data sets, infinite scrolling, and many other highly used options. This search box allows you to type complete product titles in the case of the kmeans recommender and customer ids in the case of the als recommender or substrings which makes a call an api and returns a list of products matching that title or substring. Once you select an item from the dropdown and click search you are provided with 5 recommendations. In the case of the ALS recommendation engine, there are cases where there are no recommendations (related products), this is because of the cold start problem where there is insufficient user-item interaction for a customer to give a recommendation.

Future Work

Considering the limitations of the collaborative filtering model and the relative success of the K-Means filtering approach (Smith et al., 20XX), there's a promising way to enhance personalized recommendations by combining these two strategies. Typically, the Collaborative Filtering (CF) recommendation systems rely solely on ratings and as a result, experience what is known as the cold-

start problem. With the cold-start problem, the system lacks knowledge about the preferences of new users, leading to an inability to provide relevant recommendations. Similarly, with new items, the absence of ratings results in the system's uncertainty about which users to suggest these items to. Hybrid recommendation systems address this by integrating CF or other techniques with features from items, often utilizing association rule mining (Cano & Morisio, 2017).

Furthermore, there's potential for refining the K-Means recommender. This can be accomplished by implementing filtering mechanisms to exclude reviews with low ratings (1-2 stars) and negative sentiments thereby creating a sort of popularity content-based filtering model. By adopting such a methodology, the K-Means model could be fine-tuned to highlight items of higher popularity, subsequently yielding more relevant and valuable recommendations.

Conclusion

In conclusion, the application of the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology has provided valuable insights and practical strategies for addressing two critical aspects of modern e-commerce: fake reviews detection and personalized recommendations. Throughout the project, I followed a structured and iterative approach, ensuring the effectiveness of my solutions and their alignment with business objectives.

For the Fake Reviews Detection component, with leveraging advanced Natural Language Processing (NLP) techniques and classification algorithms, I developed a robust model to identify fraudulent reviews by review length, term frequency-inverse document frequency of the review as well as its sentiment. With this I was able to distinguish between genuine and fabricated content. This not only safeguards the credibility of product reviews but also bolsters consumer trust in the authenticity of e-commerce platforms.

The Personalized Recommendation component of my project was equally comprehensive. Following a similar integration of nlp techniques I was able to provide somewhat reasonable recommendations. The model does need some refinement specifically along the lines of the product category but overall, it Is performing as it ought to. This dynamic approach not only contributes to increased customer satisfaction and engagement but also drives sales and revenue growth.

The successful implementation of the CRISP-DM process was instrumental in the achievement of our goals. By adhering to the six phases—Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment—we ensured a comprehensive and data-driven approach to both fake reviews detection and personalized recommendations.

As a result, my CRISP-DM-based approach has not only bolstered e-commerce's integrity and trustworthiness by mitigating the impact of fake reviews but also enriched the user experience through precise and engaging personalized recommendations. Moving forward, the insights gained from this project will serve as a foundation for ongoing improvements and innovations in the e-commerce ecosystem, ensuring its competitiveness and relevance in a rapidly evolving digital landscape.

References

Amir Sjarif, N. N., Mohd Azmi, N. F., Chuprat, S., Sarkan, H. M., Yahya, Y., & Sam, S. M. (2019). SMS SPAM message detection using term frequency-inverse document frequency and random forest algorithm. *Procedia Computer Science*, *161*, 509–515. https://doi.org/10.1016/j.procs.2019.11.150

Barrière, V., & Kembellec, G. (2018). Short review of sentiment-based Recommender Systems. *Proceedings of the 1st International Conference on Digital Tools & Uses Congress - DTUC '18*. https://doi.org/10.1145/3240117.3240120

Basu, S. (2021). Personalized product recommendations and firm performance. *Electronic Commerce Research and Applications*, *48*, 101074. https://doi.org/10.1016/j.elerap.2021.101074

Çano, E., & Morisio, M. (2017). Hybrid Recommender Systems: A systematic literature review. *Intelligent Data Analysis*, *21*(6), 1487–1524. https://doi.org/10.3233/ida-163209

Chiny, M., Chihab, M., Bencharef, O., & Chihab, Y. (2021). Netflix recommendation system based on TF-IDF and cosine similarity algorithms. *Proceedings of the 2nd International Conference on Big Data, Modelling and Machine Learning*. https://doi.org/10.5220/0010727500003101

Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., Richter, A. N., & Al Najada, H. (2015). Survey of review spam detection using machine learning techniques. *Journal of Big Data*, *2*(1). https://doi.org/10.1186/s40537-015-0029-9

Gorripati, S. K., Angadi, A., & Saraswathi, P. (2021). Recommender Systems. *Security in IoT Social Networks*, 149–178. https://doi.org/10.1016/b978-0-12-821599-9.00007-8

Gosh, S., Nahar, N., Wahab, M. A., Biswas, M., Hossain, M. S., & Andersson, K. (2021). Recommendation system for e-commerce using alternating least squares (ALS) on Apache Spark. *Advances in Intelligent Systems and Computing*, 880–893. https://doi.org/10.1007/978-3-030-68154-8_75

Hussain, N., Turab Mirza, H., Hussain, I., Iqbal, F., & Memon, I. (2020). SPAM review detection using the linguistic and spammer behavioral methods. *IEEE Access*, *8*, 53801–53816. https://doi.org/10.1109/access.2020.2979226

Iliopoulou, K., Kanavos, A., Ilias, A., Makris, C., & Vonitsanos, G. (2020). Improving movie recommendation systems filtering by exploiting user-based reviews and movie synopses. *Artificial Intelligence Applications and Innovations. AIAI 2020 IFIP WG 12.5 International Workshops*, 187–199. https://doi.org/10.1007/978-3-030-49190-1_17

Kontsewaya, Y., Antonov, E., & Artamonov, A. (2021). Evaluating the effectiveness of machine learning methods for spam detection. *Procedia Computer Science*, *190*, 479–486. https://doi.org/10.1016/j.procs.2021.06.056

Nallamala, S. H., Bajjuri, U. R., Anandarao, S., Prasad, Dr. D., & Mishra, Dr. P. (2020). A brief analysis of collaborative and content based filtering algorithms used in recommender systems. *IOP Conference Series: Materials Science and Engineering*, *981*(2), 022008. https://doi.org/10.1088/1757-899x/981/2/022008

Peng, Q., & Zhong, M. (2014). Detecting spam review through sentiment analysis. *Journal of Software*, *9*(8). https://doi.org/10.4304/jsw.9.8.2065-2072

Staff, A. (2023, June 28). *Amazon's latest actions against fake review brokers: Amazon sues fraudsters attempting to deceive customers*. US About Amazon. https://www.aboutamazon.com/news/policy-news-views/amazons-latest-actions-against-fake-review-brokers#:~:text=In%20February%202023%2C%20Amazon%20filed,to%20operate%20Amazon%20selling%20accounts.

Team, A. (2020, May 28). *The real impact of fake reviews - which? policy and Insight*. Which? https://www.which.co.uk/policy-and-insight/article/the-real-impact-of-fake-reviews-aRbxj0Q0aVzf

Wu, S., Wingate, N., Wang, Z., & Liu, Q. (2019). The influence of fake reviews on consumer perceptions of risks and purchase intentions. *Journal of Marketing Development and Competitiveness*, *13*(3). https://doi.org/10.33423/jmdc.v13i3.2244