Assignment2-PartA

Poojal Katiyar – 220770

8th March 2025

Introduction

The goal of this assignment is to detect fake reviews by analyzing ratings and review text to identify anomalies, ensuring that customers can make informed purchase decisions based on reliable information. For this analysis, I used the Amazon Review 2023 Dataset, which is extremely large. However, I only worked with a smaller subset — the Appliance data and its associated metadata. Since the dataset does not contain labels indicating whether a review is fake or genuine, I applied unsupervised machine learning techniques to uncover patterns and relationships within the data.

With no labeled output available, only the input data was used to train the models. The primary techniques applied for modeling were K-Means clustering and Latent Dirichlet Allocation (LDA). I have also used sentimental Analysis to detect any anomaly in the dataset. In the later stages of the assignment, I also explored whether there is any relationship between the reviews identified as fake and the price of the products.

1 Data Cleaning

The input data is provided in a JSON Lines file (`Appliances.jsonl`), where each line is a separate JSON object representing a review. The file is processed line by line to check for valid JSON formatting. Only properly formatted lines are retained and saved into a cleaned file (`clean_Appliances.jsonl`) for further analysis. Similarly, for meta data also, the cleaned file is saved in (`clean_meta_Appliances.jsonl`) after processing. I have attached these 2 cleaned files at the end of the report.

- **Handling Missing Values:** Missing values were analyzed, and empty text fields were filled, while rows with missing prices were removed to ensure data completeness.
- **Dropping Unnecessary Columns:** Irrelevant columns, such as images, were eliminated to streamline the dataset.
- **Handling Duplicates:** Duplicate entries were detected and removed to maintain data integrity and avoid bias in analysis.
- After reviewing the dataset, the following columns were identified as having unsuitable data types and were adjusted accordingly:

- **Verified Purchase:** Originally a boolean variable, it was converted into a binary numeric format (0/1) for compatibility in analysis.
- **Timestamp:** Initially stored as a string, it was transformed into a `datetime` The original review timestamp, recorded in Unix time (milliseconds since epoch), was converted to human-readable datetime format for easier analysis

2 Data Preprocessing and Feature Extraction

There were already some major features present in the dataset like ratings, text, verified user or not, helpful votes count, parent_asid, etc. But there was feature extraction done to identify some more features. Several features were created to assist in identifying suspicious or fake reviews.

- **is_short_review**: To analyze the distribution of review lengths, we computed the number of words in each review and plotted its distribution. The histogram with a KDE curve illustrates the frequency of different review lengths.

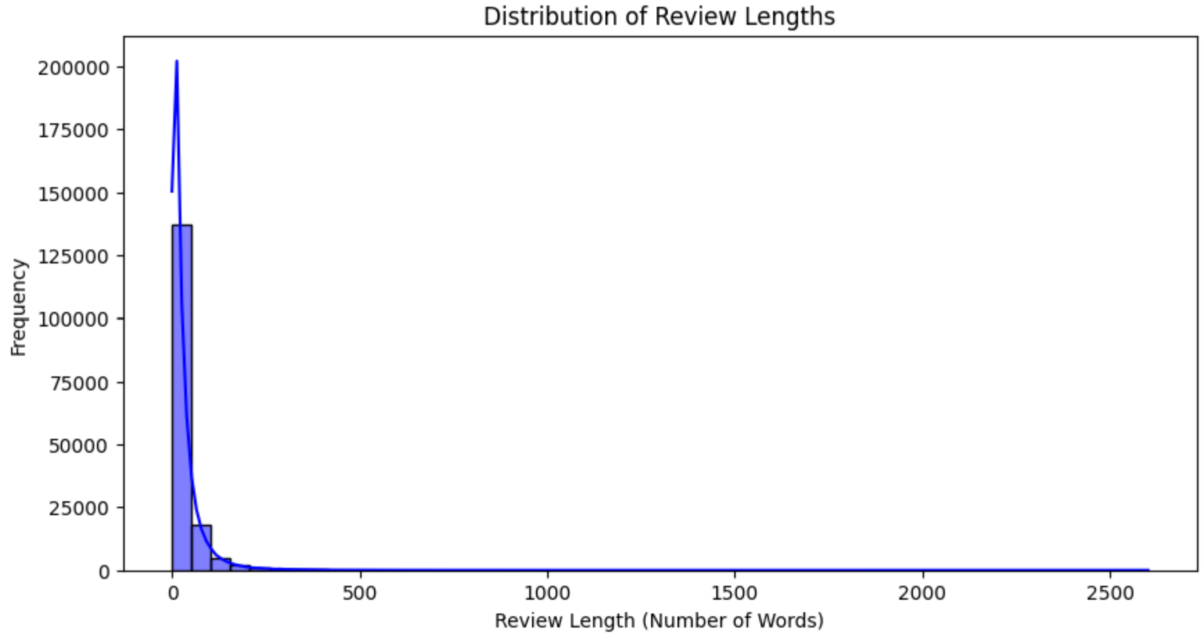


Figure 1: Distribution of Review Lengths

From the plot, it was observed that some of reviews contained fewer than 10 words. Since extremely short reviews often lack detailed explanations or justification, they could potentially indicate fake or spam reviews. Businesses or bots generating fake reviews may prefer shorter text to minimize effort while still influencing ratings. Thus, we introduced the feature **is_short_review**, which flags reviews with fewer than 10 words as potentially suspicious. This feature can be useful in identifying anomalies and improving the reliability of review-based insights.

- **Suspicious user occurrence:**: To detect potential fraudulent behavior, we analyzed how often a user reviewed the same product multiple times. Generally, a genuine customer is expected to leave only one review per product, whereas suspicious accounts may post multiple reviews to manipulate ratings. We plotted the curve and saw that such users are very few so I plan not to consider this as an important feature.

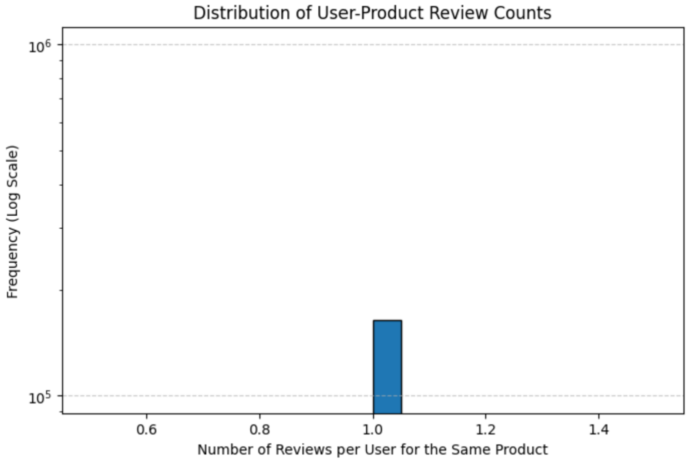


Figure 2: Distribution of User Product Review Counts

- **is_review_burst**: This flag identifies review bursts, where the daily review count for a product exceeds the 95th percentile. By grouping reviews by *parent_asin* and *review_date*, the daily review frequency is computed. If this count surpasses the defined threshold, it is marked as a review burst. Sudden spikes in reviews may indicate review manipulation campaigns, coordinated fake reviews, or promotional activities. I will show you the Review Activity for the most popular product and red dots which shows the spikes.

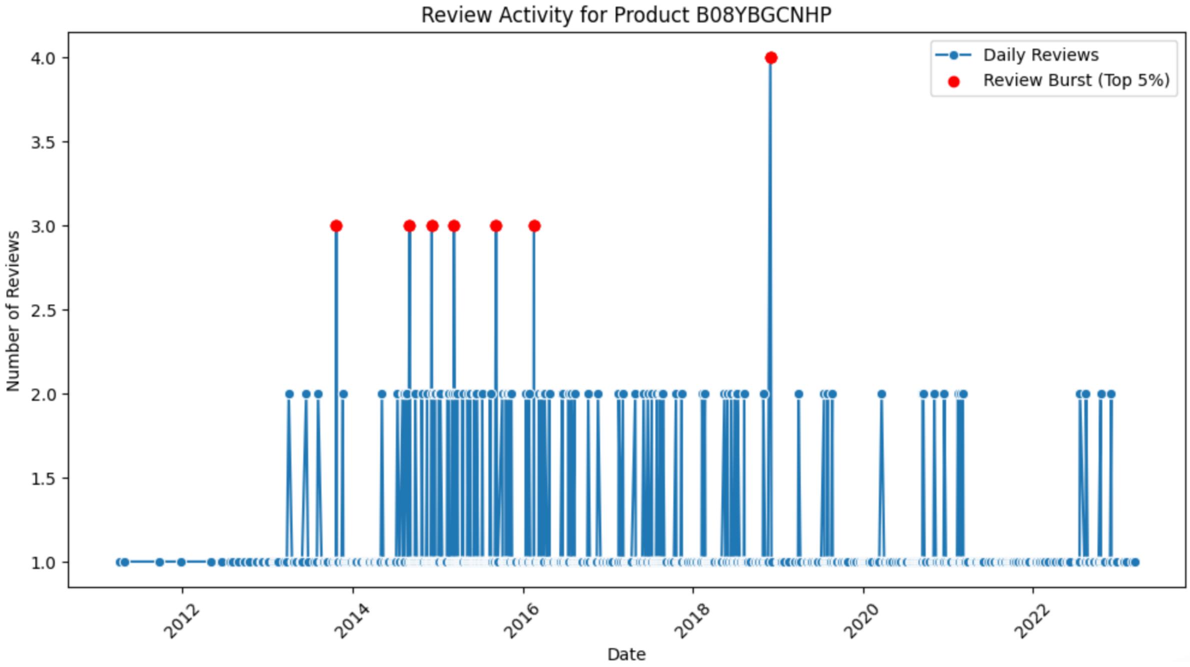


Figure 3: Review Activity for Popular Product

- **negative_word_count:** The number of negative words present in each review.

- **positive_word_count:** The number of positive words present in each review.

- **fake_review_flag:** Reviews falling in the top 5% for either negative or positive word count were flagged as potentially fake. Extremely polar reviews can be a sign of manipulation, where fake reviews are either overly critical or excessively praising. After that positive and negative word count field was removed.

- **User Count Analysis:** To understand user behavior, the number of reviews submitted by each user was analyzed. The distribution of review counts per user was calculated using percentiles.Up to the 90th percentile, each user submitted only a single review. This indicates that the majority of users contribute very few reviews, which limits the usefulness of review count per user as a distinguishing feature for detecting fake reviews. As a result, this feature was considered unsuitable for inclusion in the final model.

- To better understand reviewer behavior, the distribution of ratings given by the top reviewers (those with the most reviews) was analyzed. By plotting rating patterns for these reviewers, several distinct groups emerged:

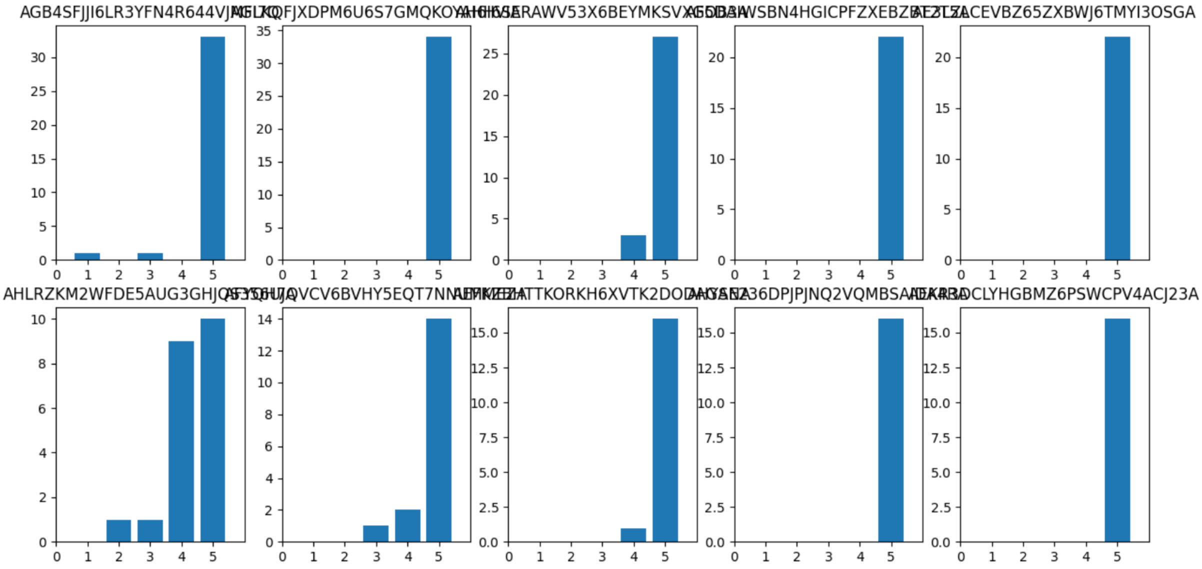


Figure 4: Voting patterns of top reviewers, showing different rating behaviors across users.

  - **A bit of everything:** Reviewers who gave a variety of ratings across the full spectrum.
  - **Mostly 5 stars:** Reviewers who primarily gave 5-star ratings.
  - **Evenly distributed:** Reviewers whose ratings were evenly spread across all values.
  - **3, 4, and 5 stars — mostly 5:** Reviewers who mostly gave high ratings, with a slight spread to 3 and 4 stars.
  - **Mainly 5 stars with a few 4 stars:** Reviewers heavily skewed towards 5-star reviews, with occasional 4-star reviews.

  - **Only 5 stars:** Reviewers who exclusively gave 5-star ratings.

In the analysis of reviewer voting patterns, one particularly suspicious group stands out — reviewers who exclusively gave 5-star ratings across all their reviews. This behavior raises concerns as these reviewers may be incentivized, either directly by businesses or through third-party services, to leave artificially positive feedback aimed at boosting product ratings. On the other hand, we do not observe a similar group of reviewers who exclusively gave 1-star ratings across all reviews. If a business were attempting to discredit a competitor's products by flooding them with negative reviews, it would be impractical and too easily detected if those reviews all came from a small set of accounts consistently leaving only 1-star ratings. Instead, such negative campaigns would likely be carried out using a larger number of accounts, each leaving only a single 1-star review on a given product. But I plotted the graph of rating distribution which clearly states that number of 1 star reviewers are very low as compared to 5 star reviewers.

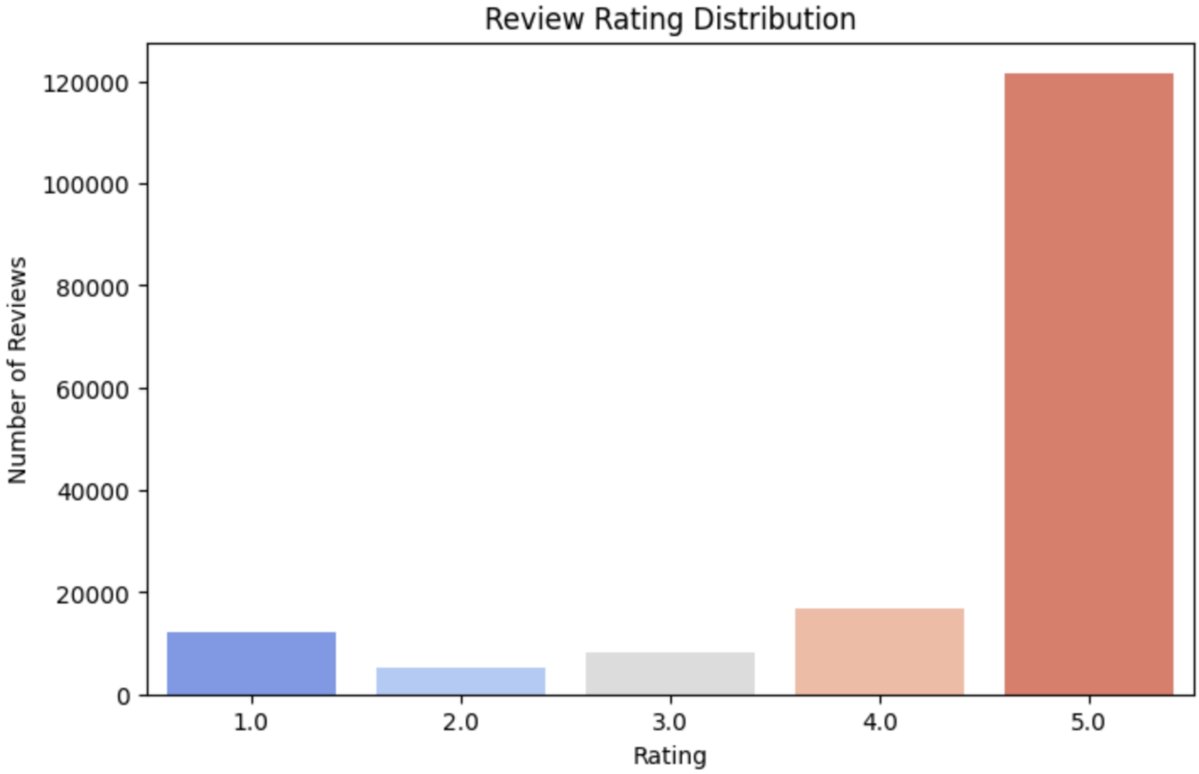


Figure 5: Review Rating Distribution.

- **multipleReviews_reviewer**: I created a binary feature for reviews per reviewer. If the reviewer had posted more than 1 review, then this new feature will be equal to 1 (ie. True), but it they only posted one review, then this feature will be equal to 0 (ie. false).

- There is a graph plotted below. From that I have drawn interpretation. As we can see, the proportion of 1-star reviews is higher from reviewers who only posted a single review compared to those who posted more than one. This supports the assumption that if a business aims to undercut its competitor(s), it may attempt to leave multiple 1-star ratings. However, to do so, they would likely need to use multiple accounts. Therefore, we would not expect to see many reviewers leaving multiple 1-star reviews; instead, there would be more reviewers with just a single 1-star review.

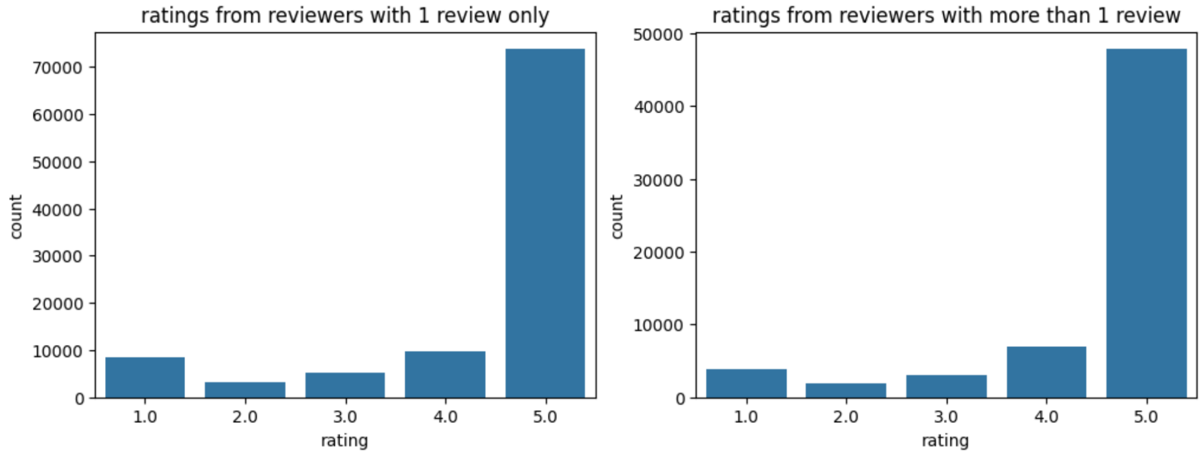


Figure 6: Ratings from reviewers with 1 vs many review

- **reviewer_five_star_only**: This feature is a binary flag (1/0) indicating whether a reviewer has exclusively given 5-star ratings.
- **reviewer_one_star_only**: This feature is a binary flag (1/0) indicating whether a reviewer has exclusively given 1-star ratings.
- Asin is just the unique product ID for the product being reviewed. The one feature we can extract is we can check how many reviews a certain product has. This can give us an indication of how popular a product is. The more reviews, the more popular. From this, we could see if more popular or less popular products tend to have more fake reviews.

  numReviews_product: It is used to get number of reviews associated with each product (asin) - to be used as a popularity index

3 Data Exploration

- I plotted the pairplot distributions for features, and looking at the graph they were highly skewed. Out of all features, helping votes, verified purchase, ratings and text are most important features. I plotted the distributions for these curves.

  From the below graphs, we can see these features are heavily skewed. Reviews are skewed to:

  - 5 star ratings
  - verified purchases
  - low number of helpful votes

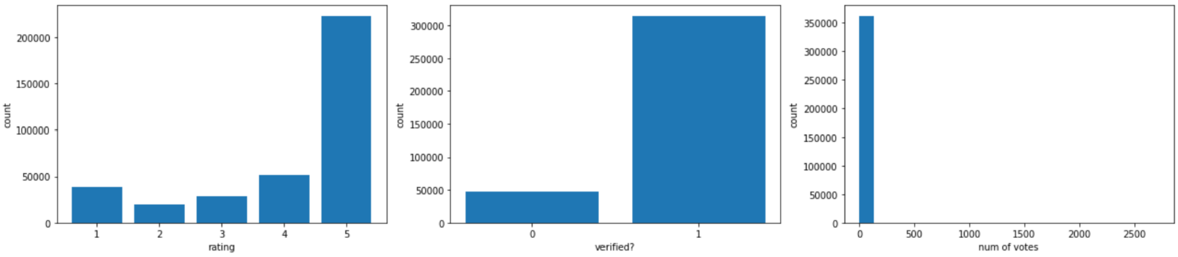


Figure 7: Distribution of ratings, verified purchases, and helpful votes.

- I plotted the distribution of number of reviews per product. Quite evidently, the distribution is heavily skewed to low values. So what we can figure out from this is most products have very few reviews.

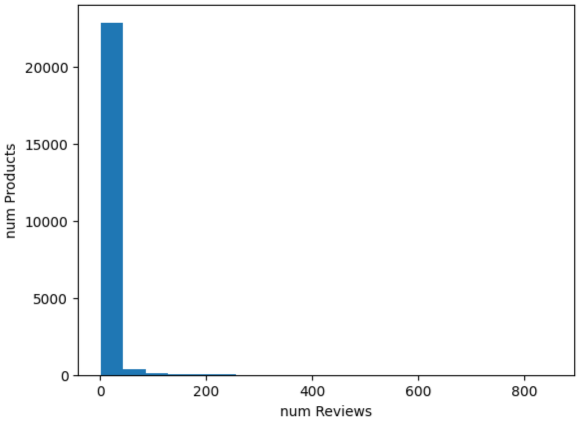


Figure 8: Number of reviews per product

Up until the 85th percentile, most products have received no more than 10 reviews, which again tells us that products have few reviews.

- I have also tried to explore trends between year and number of verified purchases. The plot is as follows:

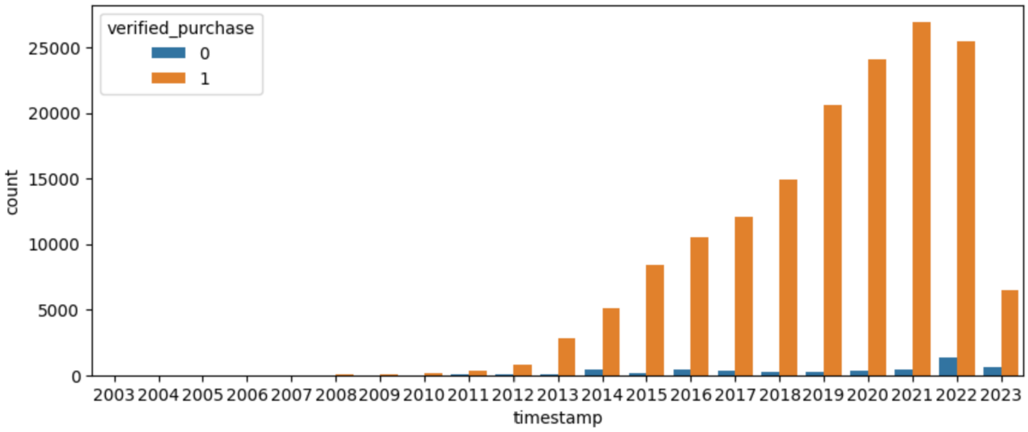


Figure 9: count of verified/non-verified purchases with time

The plot shows a clear increase in Amazon purchases during the COVID-19 period, including both verified and non-verified purchases. After the pandemic ended in 2022, purchases declined, possibly due to reduced trust in online platforms. The drop in non-verified purchases suggests Amazon introduced stricter policies to curb fake reviews. While this made it harder for competitors to post fake negative reviews, businesses might still incentivize customers to leave positive reviews, in exchange of discounted/free products. This does make it more difficult for competitors to post bad reviews as this would entail purchasing their competitor's product!

4 Tokenization, Scaling, Lemmatization, and Dimensionality Reduction

- The textual data undergoes several preprocessing steps to prepare it for analysis.Converts text to lowercase, removes punctuation, tokenizes, removes stopwords, lemmatizes, and rejoins tokens. I stored them in new column **cleaned_text** and removed the earlier **text** column.

- **Scaling Non-token Featues:** I tried multiple scalars like MinMax Scalar, RobustScalar, etc. But the most relevant was MinMax Scalar as the min max scaler bounds all features between 0 and 1. It is good as it doesn't distort the data and keeps the lower and upper bounds consistent which helps when performing clustering as the scales of all the features are the same.(The data is heavily skewed therefore others are not that relevant)

- I then performed **TF-IDF (Term Frequency-Inverse Document Frequency)** vectorization on the cleaned text data. The numeric columns were combined with the tokenized features to prepare the dataset for cluster analysis.

- The number of features after tokenisation are 508. As we have seen, the final dataframe contains 508 features so this dataset is a good candidate for dimensionality reduction. Will do so using PCA.

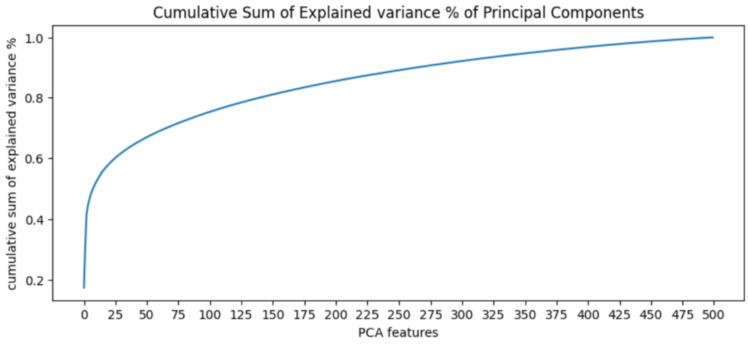


Figure 10: Cumulative Sum of Explained variance % of Principal Components

Looking at curve, it is very much evident that curve starts flattening after 200-250 features. I am interested in taking that many features which provides me with 90% explained variance. So I found out 300 features were sufficient for explained variance to be greater than 90%.

- I also plotted the PCA scatter plot which shows the first two principal components (PCA 0 and PCA 1) of the dataset. The points appear in distinct, well-separated groups, indicating strong clustering patterns in the data.The structured grouping indicates that applying clustering algorithms like K-Means Clustering could yield meaningful segmentations.

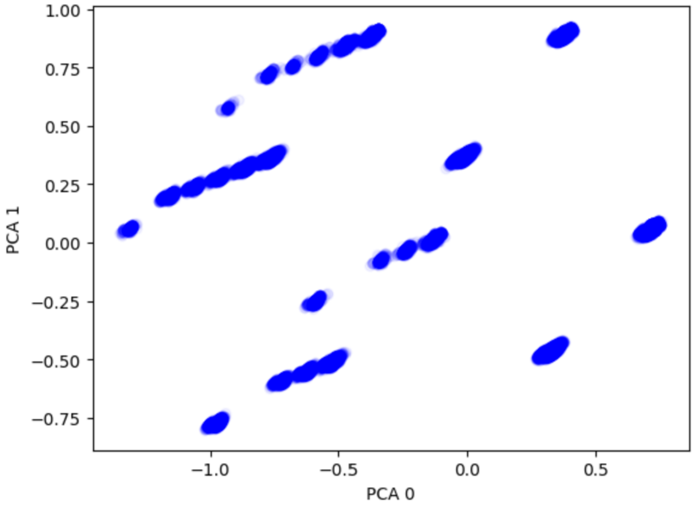


Figure 11: PCA scatter plot for first two components

5 Sentimental Analysis on Cleaned Text:

I performed sentimental analysis on the cleaned_ text obtained after text_ preprocessing. Anomalies are identified based on discrepancies between sentiment and rating: a review with a sentiment score above 0.3 but a rating below 3, or a sentiment score below -0.3 with a rating above 3, is flagged as an anomaly. The results are then visualized using Seaborn's countplot, where the frequency of legitimate and anomalous reviews is displayed under the title "Fake Review Detection" as shown:

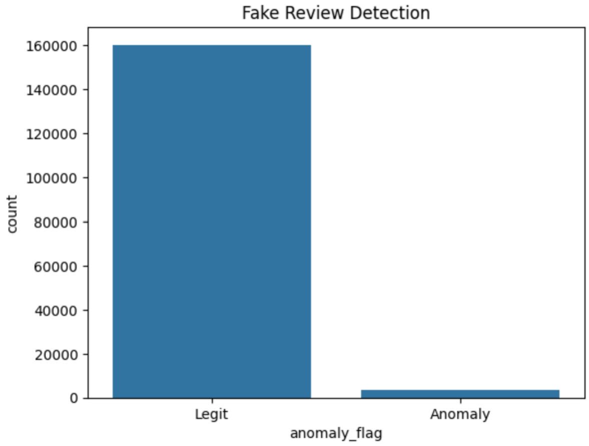


Figure 12: Anamoly vs Legit Reviews

This is one way in which by using sentimental analysis on cleaned text, I am predicting the Anamoly which could be a sign of fake reviews. If a review expresses highly positive sentiment but is rated poorly (or vice versa), it suggests a mismatch that could signal manipulation, bias, or fraudulent activity. This method helps identify suspicious patterns and improve the reliability of review-based systems.

6 Modelling using K-Means Clustering

We begin by trying to determine the optimal number of clusters by looking at inertia scores We cycle through different K values and append inertia score to plot the curve.

- We cycle through different values of k (number of clusters), computing the inertia score for each value.
- We plot these inertia scores against their corresponding k values to visualize the trend.
- The Elbow Method is used to identify the optimal number of clusters. The idea is to find the point where the inertia curve starts flattening, meaning additional clusters provide diminishing improvements.

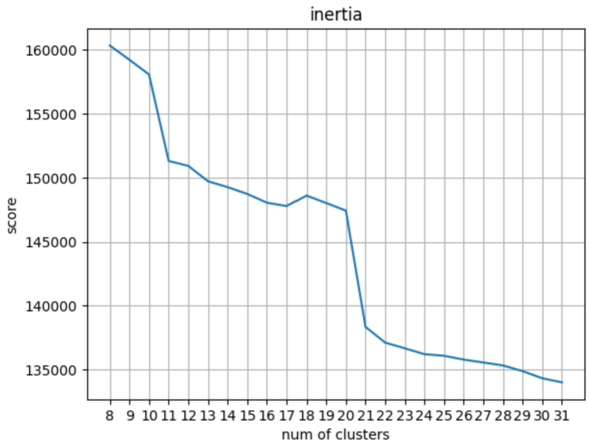


Figure 13: Inertia vs Number of clusters

From the graph, we observe a significant drop in inertia up to **k = 22**, after which the rate of decrease slows down. This suggests that k = 22 is a good choice for the number of clusters We instantiate the K-Means algorithm with k = 22 clusters and fit the K-Means model on the first 300 principal components (PCs) obtained from PCA. When we perform Principal Component Analysis (PCA), we reduce the dimensionality of the dataset from its original number of features (e.g., thousands) to 300 principal components (PCs). These 300 PCs capture most of the variance in the data while removing noise and redundancy.

Then, clustering (K-Means in this case) is performed on these 300 PCA components rather than the original high-dimensional data. This improves clustering efficiency and accuracy because PCA removes correlations and noise, making the clusters more meaningful. However, the PCA-transformed data is not directly interpretable because it is in the form of principal components rather than the original feature space. That's why we reconstruct the data back to its original feature space using matrix multiplication:

$$\text{Reconstructed Data} = \text{PCA Components} \times \text{Principal Component Loadings}$$

We add a "cluster" column to the reconstructed dataset, which assigns each row its K-Means cluster. Now I will do cluster analysis to mainly focus on clusters which has mainly extreme ratings, unverified purchases, lower amount of helpful votes and shorter reviews.

6.1 Cluster Analysis on Non-Token Columns:

- **Ratings:** I have calculated the average rating of all **22** clusters. From here we can see that more than half of the clusters have a mean of **0.148** rating. As we know, the data was heavily skewed to high ratings (ie. 4 and 5) so this on its own is not informative. What is interesting though is clusters **12, 5, and 7** have the lowest ratings, with cluster 7 having the lowest ratings(-0.85).

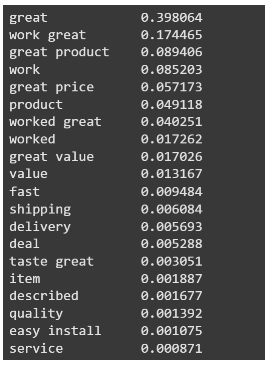
- **Verified Purchases:** Clusters **9** and **13** both have the most negative values which indicate they have the largest number of unverified purchases. These are good candidates to look into further as most unverified purchases are from fake reviewers.

- **Helping votes:** Cluster **7** and **11** have the highest helping votes and these are also those which have quite a large number of unverified Purchases. These can be potential for fake reviews.

- **MultipleReviews_reviewer:** I looked at multipleReviews_reviewer by cluster which informs us if reviewers post more than 1 review. Clusters 11, 14, 6, 13, 19, 1, 16, 5 10, and 8 were those which had the most number of reviews by a reviewer. I will examine how many of these clusters have reviewers giving 5-star reviews only. These were 13, 1, 16, 8 and 10. Some points about these clusters:

  - **All clusters have high ratings.**
  - **Clusters 16, 8, and 1** have a **high number of verified purchases**. On the other hand, **Clusters 13 and 10** have a **low number of verified purchases**.
  - **Clusters 10 and 13** also have a **low number of votes**.
  - **Cluster 10 has the lowest word count** (**Cluster 13 has a high word count**, so it is not included here).
  - **As observed earlier, all these clusters have a high number of reviewers who post multiple 5-star reviews**.

  Based on the above cluster 10 can be potentially representing fake reviews. Cluster 10 is the most suspicious in terms of potentially containing fake reviews. The combination of low verified purchases, low votes, low word count, and a high concentration of reviewers who leave multiple 5-star ratings makes this cluster of low quality. This gives credenece that if fake reviews existed, they would most likely be contained in cluster 10. Through an examination of the most frequently occurring words in Cluster 10, we observe a predominant use of generic positive terms. The top words in this cluster include:



```
great           0.398064
work great      0.174465
great product   0.089406
work            0.085203
great price     0.057173
product         0.049118
worked great    0.040251
worked          0.017262
great value     0.017026
value           0.013167
fast            0.009484
shipping        0.006084
delivery        0.005693
deal            0.005288
taste great     0.003051
item            0.001887
described       0.001677
quality         0.001392
easy install    0.001075
service         0.000871
```

Figure 14: top 20 review tokens for cluster 10

These terms are notably vague and lack specific details about the products being reviewed. This observation aligns with established characteristics of fake or incentivized reviews, which often exhibit the traits like Excessive use of generic praise, Absence of product-specific details,etc. It could also be seen through these prominence value calculated for various terms (like greatproduct, great price, great value) across different clusters in the graph below:

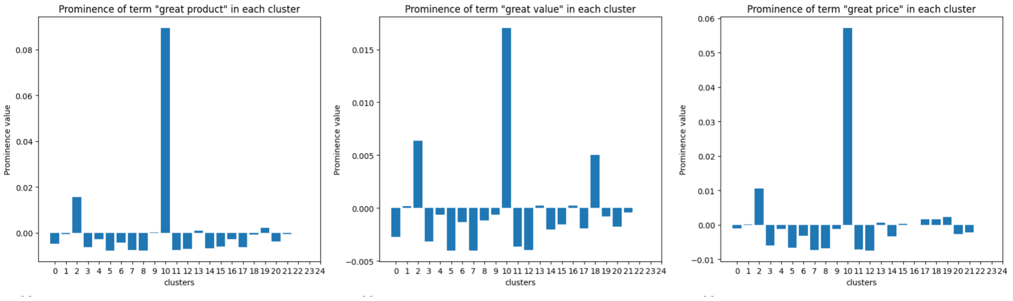


Figure 15: Prominence graph

I analyzed Cluster 10 and plotted the number of 5-star reviews over the years. I noticed a spike in reviews in 2019. To investigate further, I performed a month-wise analysis for 2019 and found that December had an unusual surge in 5-star reviews.

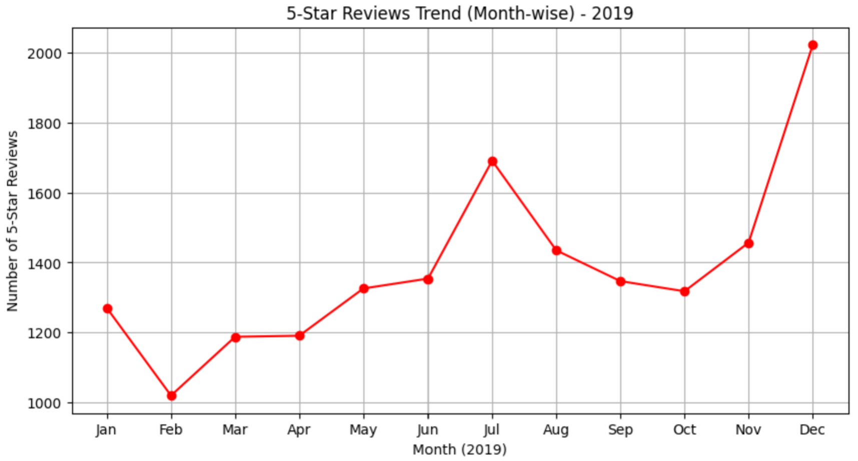


Figure 16: Month wise(5 star reviews) for 2019 of cluster 10

This could be due to fake reviews been added in December.

- **Word Count: Clusters 13, 9, 7, and 11** have a relatively **high word count**. However, **Clusters 7 and 11** have **bad ratings**, making them likely **genuine**, as they contain **long reviews, high helpful votes, and low ratings**.

  On the other hand, **Clusters 9 and 13** contain the **highest number of unverified purchases** along with **high ratings**. This suggests a potential scenario where **products are being provided for free externally from Amazon in exchange for reviews**.

  This could explain why the **word count is longer**, as reviewers might be making an effort to provide **detailed feedback**. However, their **helpful votes count is not high**, and their number of **5-star reviews is among the highest**. Additionally, they have **high review bursts** (spikes in the reviews), which could be due to **promotions**. Additionally, **Cluster 13 has a high multipleReviews_reviewer count**, meaning that many reviewers in this cluster have posted multiple reviews. The **low number of verified purchases** in these clusters further supports the possibility that many reviewers **received the products for free**, leading to an increased likelihood of **biased reviews**. While these reviews may not necessarily be **fake**, they could be **unreliable** since **almost all reviews in Clusters 9 and 13 are 5-star ratings**, suggesting that reviewers may have been **incentivized** to leave positive feedback.

  Additionally, I have performed **Latent Dirichlet Allocation (LDA)**, which I will discuss in the next section. During the analysis, I noticed that one of the topics contained prominent terms such as *item, recommend, would, highly, highly recommend, and would recommend.* These terms strongly suggest that some reviews may have been **given in exchange for a product**, as they emphasize **recommendations** rather than **genuine user experiences**.

  This pattern is also evident through the **prominence values calculated for various clusters** in the graph below, where we can see that **Clusters 9 and 13** rank among the highest for these words.

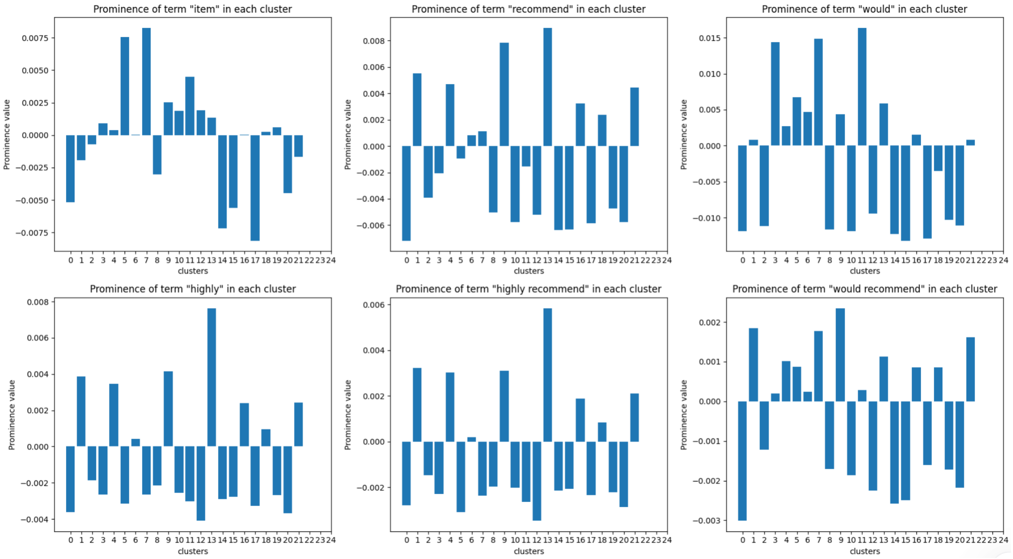


Figure 17: Prominence graph

- I believe Cluster 4 is the most genuine cluster that got Positive Reviews. This is because:
    - It is among the cluster with the **highest ratings**.
    - It has a **high number of votes** as well as a **high number of verified purchases**.
    - The reviews are **neither too short nor too long**, maintaining a balanced length.
    - Reviews are posted by **multiple distinct users**, rather than a single reviewer repeatedly giving **5-star ratings**.
    - It does not contain products with **sudden spikes in reviews** (low `is_review_burst`), indicating a more **organic review pattern**.
    - The **top 20 review tokens** in **Cluster 4** primarily focus on the **product itself**, mentioning terms like *coffee, ice, dryer*, etc., making it a **genuine cluster**.
    - Analysis of reviews over the years shows that **2021 and 2022 had the most reviews**, with **no unusual monthly spikes**, further supporting the **authenticity** of this cluster.
- Cluster 7 according to me is also a genuine cluster which have negative reviews because:
    - It is the cluster with the **lowest rating reviews**.
    - It contains reviews that are mostly posted by **reviewers who post only a single review**.
    - It has a **low review burst**, indicating no sudden spikes in review activity.
    - It has the **highest number of helpful votes**, suggesting that users find these reviews useful.
    - The **top 20 review tokens** of Cluster 7 include negative words like *waste of money, disappointed, return*, etc., indicating dissatisfaction.
    - The reviews are **longer in length**, which is generally associated with **more genuine and detailed feedback**.

6.2 Conclusion

- **Cluster 10** exhibits the most telltale features of **fake reviews**, characterized by **5-star ratings**, **low word count**, and the use of **general positive terms**.
- **Cluster 13** contains reviews from people who may have been **incentivized** to give positive feedback; while these reviews are not necessarily fake, they may be **unreliable**.
- Based on my observations, the **most genuine cluster** is **Cluster 4**, which provides **high ratings** to the products.
- Another **genuine cluster** is **Cluster 7**, which tends to provide **low ratings**.

7 Latent Dirichlet Allocation(LDA) Analysis:

In parallel with K-Means Clustering, I also would like to try clustering the review text using LDA Topic Modeling. The key difference between the 2 clustering methods is LDA topic modeling clusters reviews into different topics by solely looking at text data which in this case will be the review text. In contrast, K-Means Clustering can cluster the reviews based on all features, tokenized text, and other numeric features.

For LDA topic Modeling, we need to pre-select the number of topics we think exist in our text. To be consistent with K-Means clustering, I will choose 22 topics as we had selected 22 clusters for K-Means. Note: this is not necessarily the optimal way to determine the number of topics. In LDA topic modeling, 22 topics were generated from the review text. For the most part, the topics were quite clear. There were topics related to:

- Logistics (e.g. shipping)
- Specific types of products like Household & Kitchen Appliances, Water Filtration & Purification, Coffee & Beverage Accessories,etc.
- General positive sentiment (e.g. great product, recommend)
- General negative sentiment (e.g. waste of money, didn't work)

As mentioned previously, fake reviews would generally use general terms and make grand claims. Through LDA topic modeling, there were several topics that fit this description. This gives a good sense that there are possibly fake reviews within the corpus. These topics are generally those which are not product related and generally just some Positive words. I will give the Word Cloud for those topics:

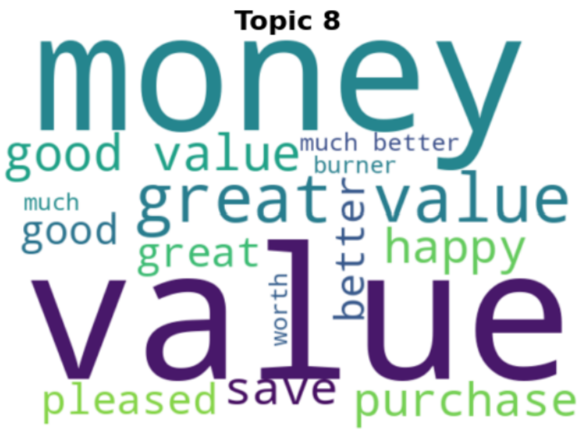


Figure 18: Topic 8

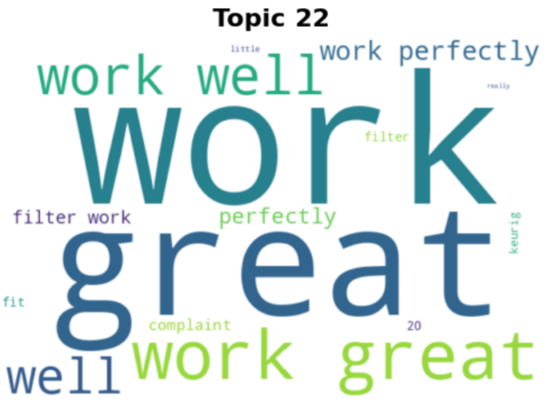


Figure 19: Topic 22

8 Relation between Price of items and fake Reviews:

I am considering these incentivized reviews as also fake. The conclusion was drawn from K-Means that cluster 10 and 13 are probably Fake Reviews. I have not taken cluster 9 as I had more doubts about cluster 13 as it had multiple reviews by a reviewer. I will label them as fake_review_suspect. This I am doing is to see the relation that price has on a number of Fake Reviews.

Key Findings:

- **Coefficient for fake_review_suspect** (-0.0003):
  - The negative coefficient suggests that products with suspected fake reviews tend to have slightly lower prices.
  - The effect size is very small (-0.0003), meaning the impact on price is minimal.
- **P-value** (0.004):
  - Since the p-value is less than 0.05, the relationship between `fake_review_suspect` and `price_1` is statistically significant.
  - However, statistical significance does not imply practical significance—the effect is very small.
- **R-squared** (0.000):
  - This indicates that the model explains almost no variance in `price_1`.
  - In other words, `fake_review_suspect` is not a strong predictor of price.

I plotted boxplot to visualize the price for cluster 13 and cluster 10.

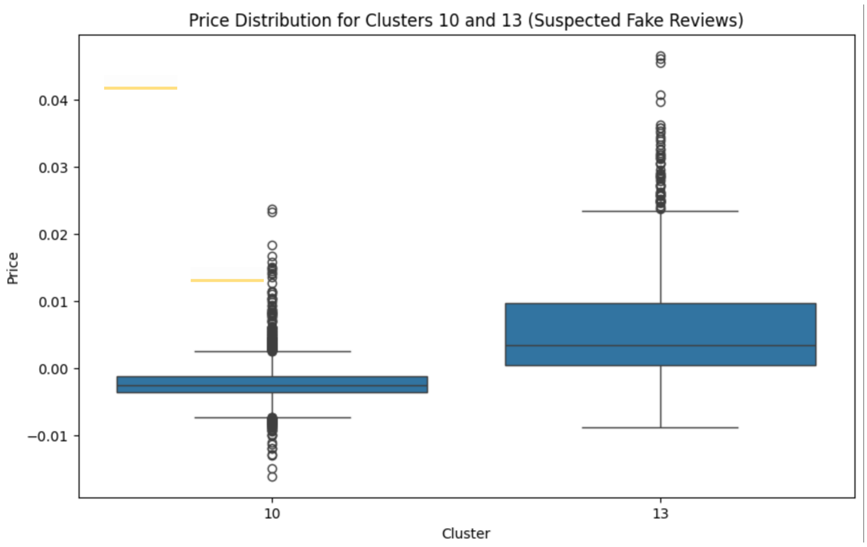


Figure 20: Boxplot for price of cluster 10 and Cluster 13

This boxplot shows that the mean for Cluster 10 is less than Cluster 13. The mean price of Cluster 10 is lower than that of Cluster 13. This is because fake reviews are generally associated with lower-quality, discounted products, while incentivized reviews are linked to promotional or higher-quality items that are priced higher. In essence, the difference in pricing reflects the nature of the reviews: fake reviews tend to be linked with cheaper products, while incentivized reviews are often associated with more expensive, higher-quality promotional items. This is just the interpretation that I am making.

I have also plotted a histogram to visualize the price distribution for two different clusters (Cluster 10 and Cluster 13).

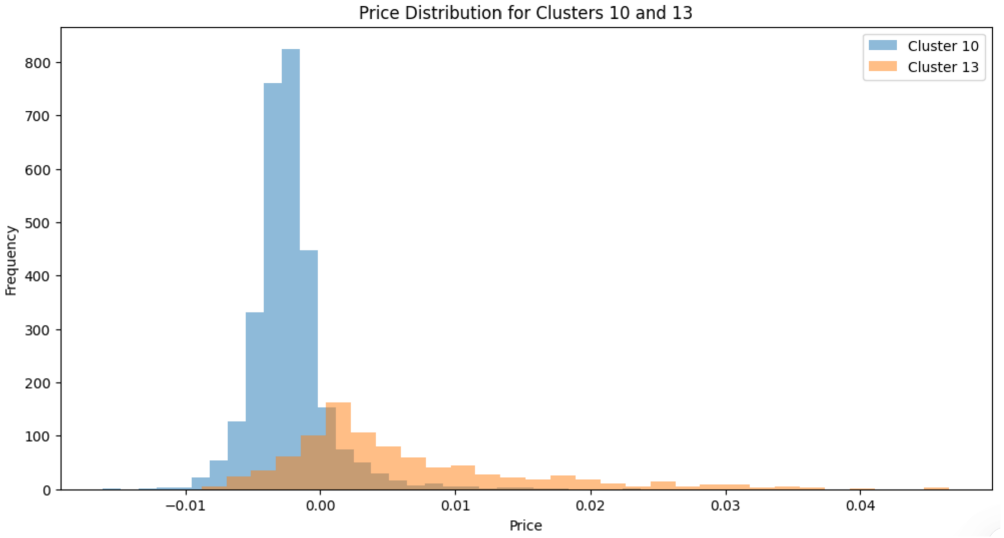


Figure 21: Price Distribution for Cluster 10 and 13

This plot also shows that the frequency of products in Cluster 10 is higher and limited to a lower price range. In contrast, Cluster 13 contains products with prices that span a broader range, covering higher price points as well.

Improvements that can be made:

- Deal with class imbalances that exist in the data
- Use more data from the total dataset
- Experiment with other clustering method

The potential utility of having such a model is to create a web application that generates grade reports for online product reviews. For example, a customer could copy and paste a url of a product of interest into the web app. The web app would then grab all the reviews for that product and grade them based on what cluster they fall into. If reviews fall into one of the clusters of potentially fake reviews, then the grade for the product would decrease. The customer would then be able to discern, based on the final grade, if the reviews for their product of interest are reliable or not.

References:

- https://sanjayasubedi.com.np/nlp/nlp-with-python-document-clustering/
- https://medium.com/@dmitriy.kavyazin/principal-component-analysis-and-k-means-clustering-to-visualize-a-high-dimensional-dataset-577b2a7a5fe2

- https://pandas.pydata.org/pandas-docs/stable/
- https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0

Colab Links and Dataset:

I have collected data from the Amazon Reviews 2023 site. However, due to the large size of the data and some invalid JSON formatting, I had to clean it. I have provided a Google Drive link to the cleaned dataset for Appliances in JSONL format. You can upload these files and run the code on Colab. I have commented out the cleaning done on original data and metadata of Appliances.

- **Colab Link for the Code:** Assignment_2_PartA
- **DataSet used (for Appliances):** clean_Appliances .jsonl file
- **MetaData used (for Appliances):** clean_meta_Appliances .jsonl file