Amazon US Reviews on electronics. CRISP-DM Analysis

Daniel Vakhrushev, Polina Zelenskaya, Ivan Chernakov, Elina Akimchenkova, and Roman Voronov

Innopolis University

Contents

1 Business Understanding

This initial phase focuses on understanding the given dataset, its' objectives, and requirements from a business perspective, then converting this knowledge into a data mining problem definition and a preliminary plan designed to achieve the objectives.

1.1 Glossary

- API (Application Programming Interface) – A suite of REST-based APIs that provides Amazon selling partners programmatic access to their Amazon Seller or Vendor Central account data
- SR (Sales Rank) – The ranking of an item in a product category.
- CTI (Category, Type, Item) – It is a term used in e-commerce and refers to the specific product category, type, and item number assigned to a particular product.
- ASIN (Amazon Standard Identification Number) – An ASIN that requires shoppers to select additional options, such as size or color, to purchase.
- Cloud Data Warehouse – An electronic system used to gather data from various sources within a company and then provides automated provisioning, administration, tuning, backup, and recovery to accelerate analytics and actionable insights while minimizing administration requirements.
- Big Tech – Big Tech, also known as the Tech Giants, refers to the largest information technology companies, primarily the Big Five tech companies in the United States: Alphabet (Google), Amazon, Apple, Meta (formerly Facebook), and Microsoft.
- Marketplace – A marketplace refers to a platform where individuals, businesses, or organizations can buy and sell goods or services.
- E-commerce Marketplace – An e-commerce marketplace, such as Amazon, is a digital platform that enables third-party sellers to list and sell their products to a wide customer base.

1.2 Business Objectives

Amazon, known as Amazon.com, is a global leader in e-commerce and cloud services. Initially established as an online book-selling company, Amazon has expanded its operations to encompass a wide range of internet-based services. These include e-commerce, cloud computing, digital streaming, and artificial intelligence (AI) services. The data that we are working with is a collection of reviews on United States' Amazon marketplace, specified in electronics. Amazon wants to enhance user experience: deliver the fastest, most comfortable, and profitable service to buyers and sellers. One of the main points to obtain such a goal is to make a precise recommendation system, so all parties could benefit from such predictions.

1.3 Assess Situation

As part of the Big Tech Five, Amazon has any possible resource for Data Mining purposes. Departments for Analysts, DB maintainers, ML Residency, and so on. Any computing resources (reasonable) can be given for the RecSys task.

The approximate period for the project is 6 months, including all CRISP-DM stages (3 months) and A/B Testing in production on some small settlement (also 3 months). The timing is tight due to seasonal customer sentiment, which is 3 months.

Costs and benefits

The company has to pay for the data mining experts, analysts, and equipment for data mining. The main idea is to increase retention on the site, which will be enhanced with a better Recommendation System. All the parties involved (Amazon, customers and sellers) will benefit from a more consumption.
As a result, Amazon will benefit from the results of the project as it will help with making a more clear and consistent connection between buyers and sellers (more sales, more customers, better retention).

1.4 Data Mining

In the context of Amazon's recommending system, the data mining objectives revolve around harnessing data to elevate customer satisfaction, enhance strategic decision-making, and achieve business objectives. Such leverage will affect predicting customer preferences. Based on their history of purchase, determine high-value customers and have an affect on inventory planning for whole company (but it is in a grand schema of planning).

1.5 Project plan

Project plan consists of the following stages:

- **Data understanding.**
  We will use given dataset to understand its features and purpose. The data is huge and very clean. Some preparation is needed, but on the first glance, it's nothing more than just feature choices. Timing: 1 week. Tools might be used: Python, SQL, Excel.
- **Data preparation.**
  The longest stage of the project. We will have to prepare the data for further analysis. Remove outliers, Normalize it and to make it ready for analysis. It might take a couple of weeks. Tools might be used: Python, SQL, Excel, Tableau, Power BI.
- **Modeling.**
  We will use different algorithms to solve RecSys task. Currently we are thinking about two ways on either clustering data or building ML model (Graph NN) for solving such task.
- **Deployment.**
  We will deploy the results of the project to the company. It might take a couple of weeks. Tools might be used: Python, Data Visualization tools, Power BI, Tableau, MS Office.

2 Data Understanding

In this phase, we will explore the Amazon dataset to gain a deeper understanding of the available features.

2.1 Initial Data

The data is sourced from Kaggle, a popular online platform for data science and machine learning competitions. Specifically, it is the "Amazon reviews US electronics," provided by a user identified as 'MOHITSHARMA527'. The dataset is publicly accessible and can be located at the URL.

2.2 Data Description

The dataset consists of 3,091,024 records and 15 columns, each containing valuable information about Amazon product reviews. The columns are as follows:

- **marketplace**: Identifier for the Amazon marketplace (there is only one marketplace, "US").
- **customer-id**: Unique identifier for the customer who has written the review.
- **review-id**: Unique identifier for the product review.
- **product-id**: Unique identifier for the reviewed product (ASIN).
- **product-parent**: Unique identifier for the parent product of the reviewed product.
- **product-title**: Title of the reviewed product.
- **product-category**: Category or type of the reviewed product (only one category, "Electronics").
- **star-rating**: Rating given by the customer, ranging from 1 (lowest) to 5 (highest).
- **helpful-votes**: Number of helpful votes received by the review from other customers.
- **total-votes**: Total number of votes (helpful and not helpful) received by the review from other customers.
- **vine**: Binary flag indicating if the review is part of the Amazon Vine program.
- **verified-purchase**: Binary flag indicating if the review was written by a customer who purchased the product on Amazon.
- **review-headline**: Title of the customer's review.
- **review-body**: Full text of the customer's review.
- **review-date**: Date when the review was posted ('YYYY-MM-DD' format).

2.3 Data Exploration

2.3.1 Univariate analysis

Univariate analysis examines one variable at a time to understand its characteristics and distribution.

1. **Star ratings**: The majority of the ratings in the dataset are 5 stars (1,779,371), followed by 4 stars (536,427), 1 star (357,817), 3 stars (238,387), and 2 stars (179,032). This indicates a generally positive trend in the reviews, as most customers seem to have had a satisfactory experience with the products.

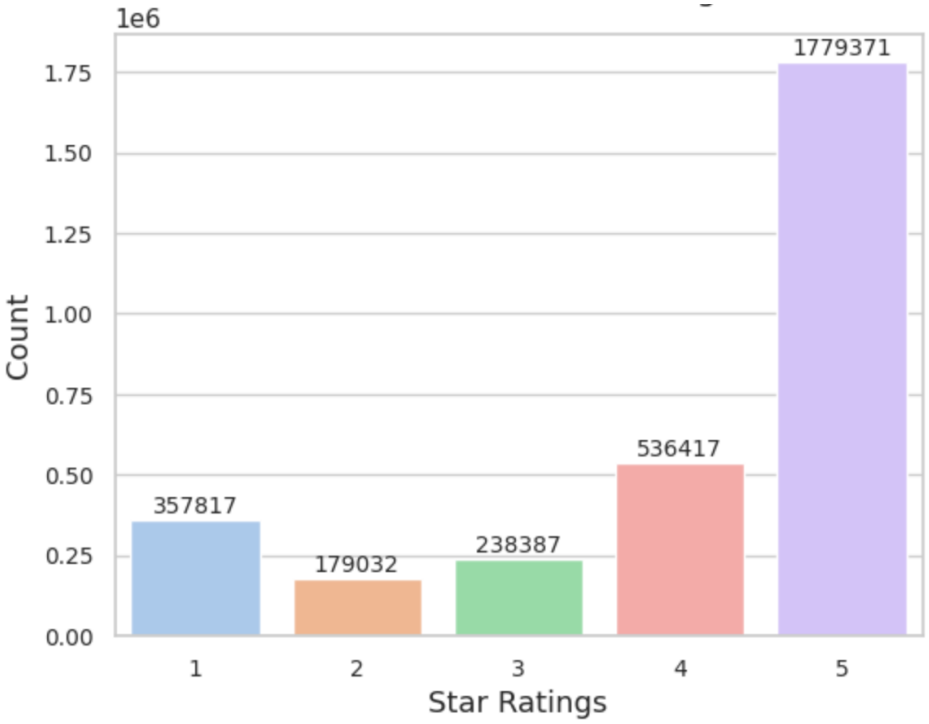

Figure 1: Distribution of Star Ratings

2. **Helpful votes**: The distribution of helpful votes is highly skewed to the right, with most reviews receiving few helpful votes. The density plot peaks at around 0 helpful votes per review. The density decreases as the number of helpful votes increases, with a nearly negligible density of reviews with 12 or more helpful votes.

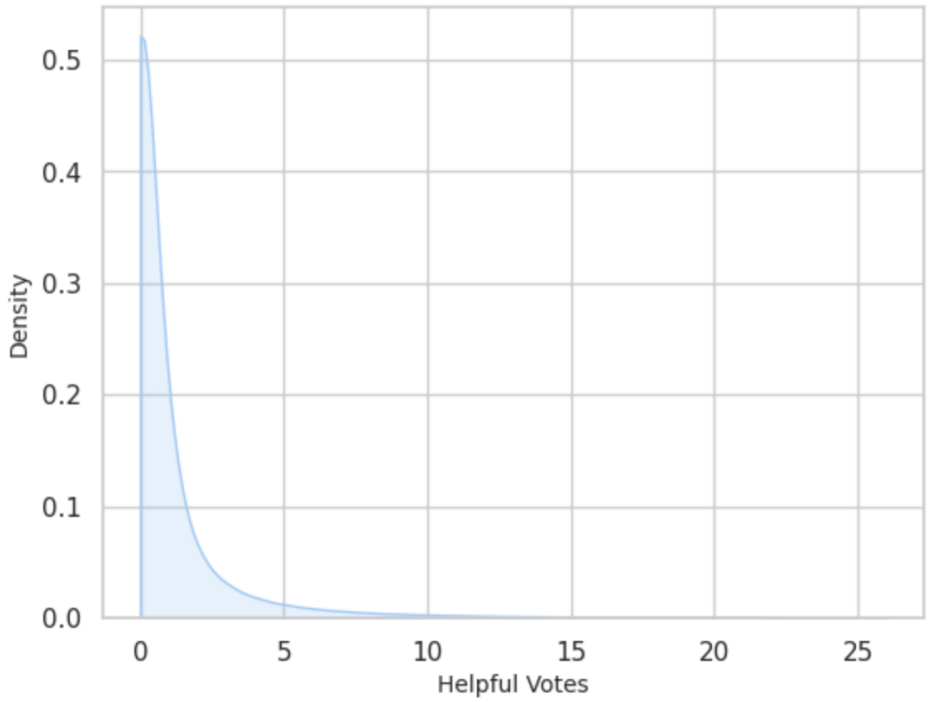

Figure 2: Distribution of Helpful Votes

3. **Total votes**: The distribution of total votes is also right-skewed, with most reviews receiving a small number of total votes. The density plot peaks at around 0 total votes per review, with the density decreasing as the number of total votes increases. Very few reviews have 15 or more total votes.

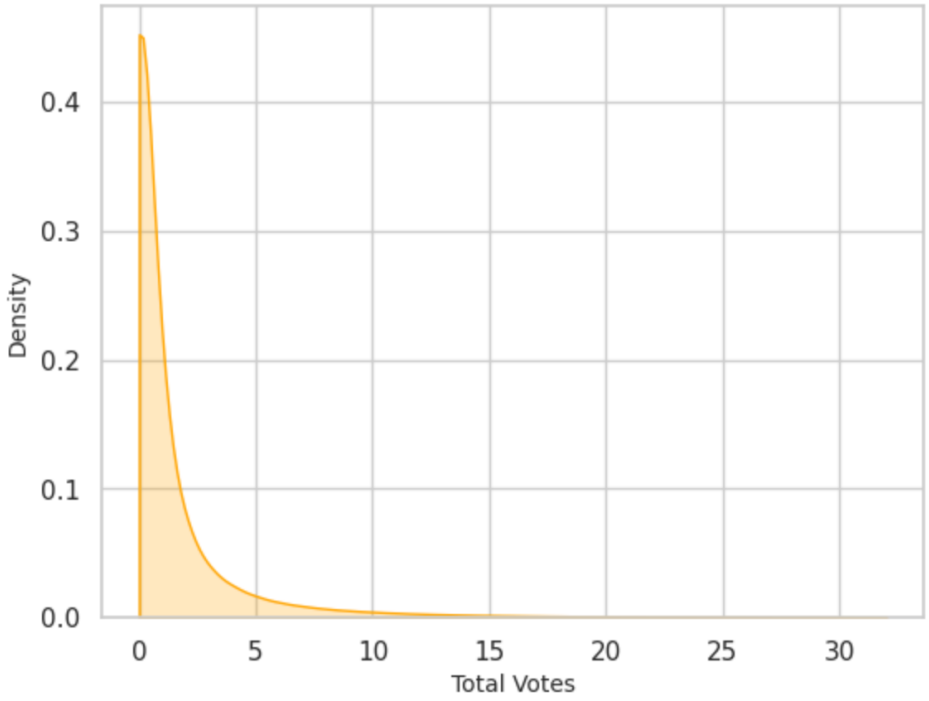

Figure 3: Distribution of Total Votes

4. **Product titles**: The word cloud of product titles shows recurring keywords such as "cable," "HDMI," "wireless," "system," "Apple," "stereo," "wall," "control," "Bluetooth," "remote," "audio," "video," and "speaker." This provides insight into the most commonly reviewed electronic products in the dataset.

Figure 4: Wordcloud of Product Titles

5. **Vine and verified purchase distributions**: For the 'vine' variable, an overwhelming majority of the reviews are not part of the Vine program (3.1 million), while a minuscule number of reviews are part of the program (less than 0.1 million). For the 'verified-purchase' variable, the dataset contains more reviews with verified purchases (approximately 2.6 million) than without verified purchases (around 0.5 million).

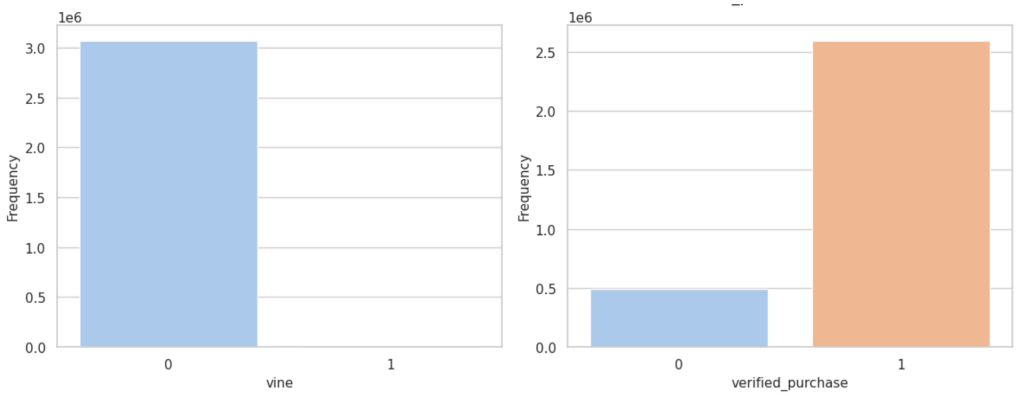

Figure 5: Vine and Verified Purchase Distribution

2.3.2 Bivariate analysis

We examined the relationships between pairs of variables using bivariate analysis. We studied the correlations between numeric variables ('star-rating', 'helpful-votes', 'total-votes', 'vine', and 'verified-purchase'), the distribution of star ratings by verified purchase and Vine program participation, and compare the top 10 product ratings.

1. **Correlation matrix**: A heatmap of the correlation matrix for the selected numerical columns reveals the following notable correlations:

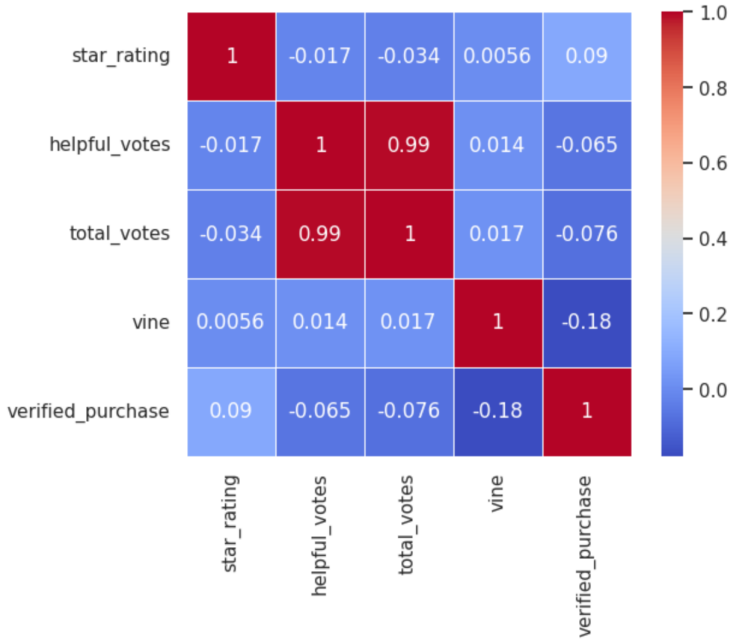

Figure 6: Correlation Matrix Heatmap

Some relevant correlations include:

- Negative correlations between 'star-rating' and 'total-votes' (-0.034) and 'star-rating' and 'verified-purchase' (-0.09)
- A strong positive correlation between 'helpful-votes' and 'total-votes' (0.99)
- Negative correlations between 'helpful-votes' and 'verified-purchase' (-0.065) and 'total-votes' and 'verified-purchase' (-0.076)
- A moderate negative correlation between 'vine' and 'verified-purchase' (-0.18)

2. **Star Ratings by Verified Purchase**: The box-plots show the distribution of star ratings for verified and non-verified purchases. The distribution appears to be similar for both groups, with the non-verified group having a wider concentration of ratings between 4 and 5 and a slightly lower concentration of lower ratings (1 and 2) as compared to the verified group.

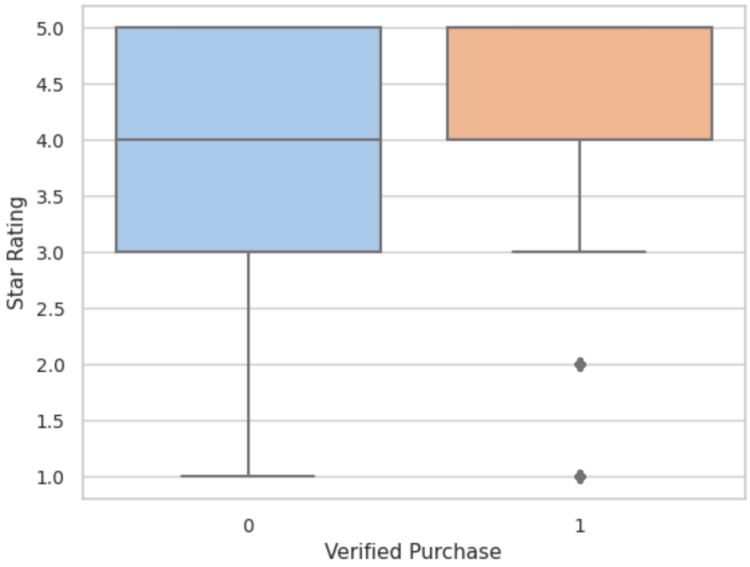

Figure 7: Distribution of Star Ratings by Verified Purchase

3. **Star Ratings by Vine Program Participation**: The box-plots show the distribution of star ratings among Vine program participants and non-participants. Vine participants have a higher concentration of ratings between 4 and 5, and a lower concentration of ratings between 1 and 2 compared to non-participants.

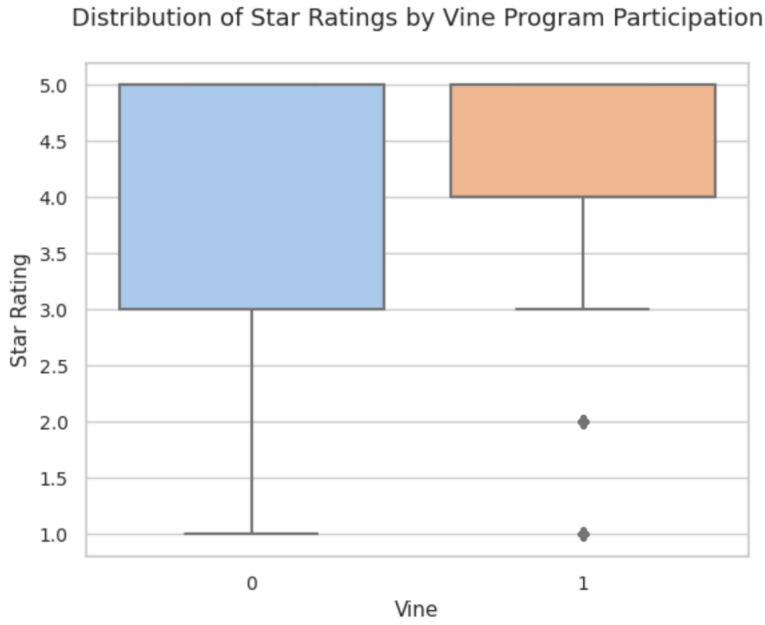

Figure 8: Distribution of Star Ratings by Vine Program Participation

5. **Top 10 Product Reviews**: The most popular product in the dataset has a remarkable 15,334 reviews. This high number of reviews demonstrates a significant level of customer engagement and interest in the product. Furthermore, all products in the top 10 list have thousands of reviews, ranging from 6,062 to 15,334.

	product_id	num_reviews	product_title
0	B003L1ZYYM	15334	AmazonBasics High-Speed HDMI Cable - 6.5 Feet ...
1	B0002L5R78	11166	High Speed HDMI Cable (1.5 Feet) With Ethernet...
2	B003EM8008	9766	Panasonic ErgoFit In-Ear Earbud Headphone
3	B0012S4APK	9359	Cheetah APTMM2B TV Wall Mount for 20-75" TVs u...
4	B0001FTVEK	8793	Sennheiser On-Ear 926MHz Wireless RF Headphone...
5	B000WYVBR0	7835	VideoSecu ML531BE TV Wall Mount for most 22"-5...
6	B0019EHU8G	7586	Mediabridge ULTRA Series HDMI Cable (3 Foot) -...
7	B00F5NE2KG	6688	Bluetooth Speaker, DKnight Magicbox Ultra-Port...
8	B004QK7HI8	6536	Mohu Leaf 30 TV Antenna, Indoor, 30 Mile Range...
9	B00D5Q75RC	6062	Bose SoundLink Mini Bluetooth Speaker

Figure 9: Top 10 Product Reviews

4. **Top 10 Product Ratings**: The violin plot displays the distribution of ratings for the top 10 products, showing a high number of 5-star ratings and a smaller number of 4-star ratings. The average rating for these products is 4.46.

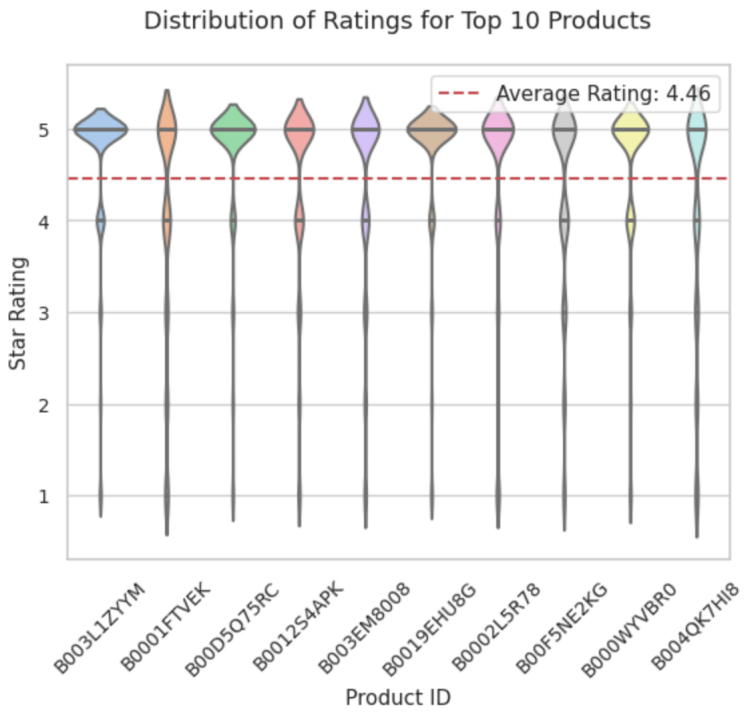

Figure 10: Distribution of Ratings for Top 10 Products

5. **Top 10 reviewers**: The table below lists the top 10 reviewers in the dataset, ranked by the number of reviews they have written. Each row in the table provides the unique customer ID, the total number of reviews written by that customer, and their average star rating. The top reviewers tend to give high ratings, with the average star ratings mostly above 4.0. This suggests that the most active reviewers generally have a positive view of the products they review. Reviewer 30669680 stands out with a lower average rating of 3.47, which could indicate a more critical approach to reviewing products. The high number of reviews from these top reviewers indicates their significant influence on the overall rating and perception of products within the dataset.

	customer_id	num_reviews	avg_star_rating
0	49266466	234	4.307692
1	53075795	198	4.106061
2	30669680	190	3.473684
3	53037408	180	4.861111
4	50820654	171	4.538012
5	52938899	166	4.331325
6	44834233	159	4.798742
7	50027179	158	4.601266
8	39789300	154	4.811688
9	32038204	149	4.704698

Figure 11: Top 10 reviewers

6. **Correlation between Number of Reviews per Customer and Average Star Rating**: The scatter plot on the left illustrates the distribution of the number of reviews per customer, while the box plot on the right shows the average star rating per customer. The scatter plot reveals that the majority of customers have written fewer than 50 reviews, with a few outliers who have written over 150 reviews. The box plot indicates that the average star rating per customer tends to be high, with most ratings falling between 3.5 and 5.0. These plots together suggest that while most customers write only a few reviews, those who are more active in reviewing tend to give high ratings. This could imply that frequent reviewers have a generally positive outlook or are more engaged with products they like. The concentration of high average ratings across the customer base highlights a trend towards positive feedback in the dataset.

Figure 12: Correlation between Number of Reviews per Customer and Average Star Rating

7. **Product parent and number of products**: The table below shows the top product parents along with the number of associated product IDs. Each product parent is a unique identifier that groups multiple related products, providing insight into product families with a high number of variations or versions. These high numbers of product IDs under single product parents suggest a diverse range of product variants within certain product families, reflecting the manufacturer's strategy to cater to varied customer preferences and market segments.

	product_parent	num_product_ids
52888	318698851	51
137194	825631183	44
162300	976432208	38
43152	259482787	37
53958	325003594	34

Figure 13: Product parent and number of products

2.3.3 Time series analysis

1. **Rating Distribution Over Time**: The time series plot illustrates the average star rating of reviews over time from 2000 to 2015.

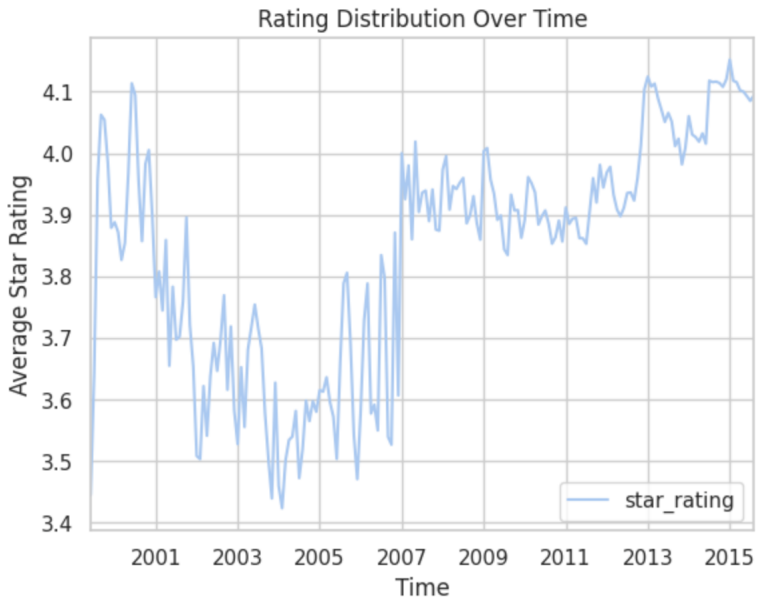

Figure 14: Rating Distribution Over Time

The plot shows notable fluctuations in average star ratings throughout the years:

- In the early 2000s, the average star rating experienced a significant decline, dropping from around 4.1 in 2000 to approximately 3.4 by 2003.
- A gradual increase followed, with some volatility, reaching around 3.9 by 2007.
- From 2008 onwards, there is a more consistent upward trend, peaking near 4.1 in the early 2010s and stabilizing around 4.0 towards the end of the period.

This pattern suggests that the average star rating has generally improved over time, with some periods of decline and recovery. The initial dip and subsequent rise may reflect changes in customer expectations, product quality, or review behavior. The overall upward trend in recent years indicates an increasing level of customer satisfaction.

2. **Number of Reviews per Year**: The bar chart below displays the number of reviews submitted each year from 1999 to 2015. The chart shows a clear increasing trend in the number of reviews over time:

- From 1999 to 2008, the number of reviews per year was relatively low, with a gradual increase starting around 2005.
- A significant surge in the number of reviews can be observed starting from 2009, with a steep increase each subsequent year.
- The peak is reached in 2014 and 2015, with over 700,000 reviews submitted each year.

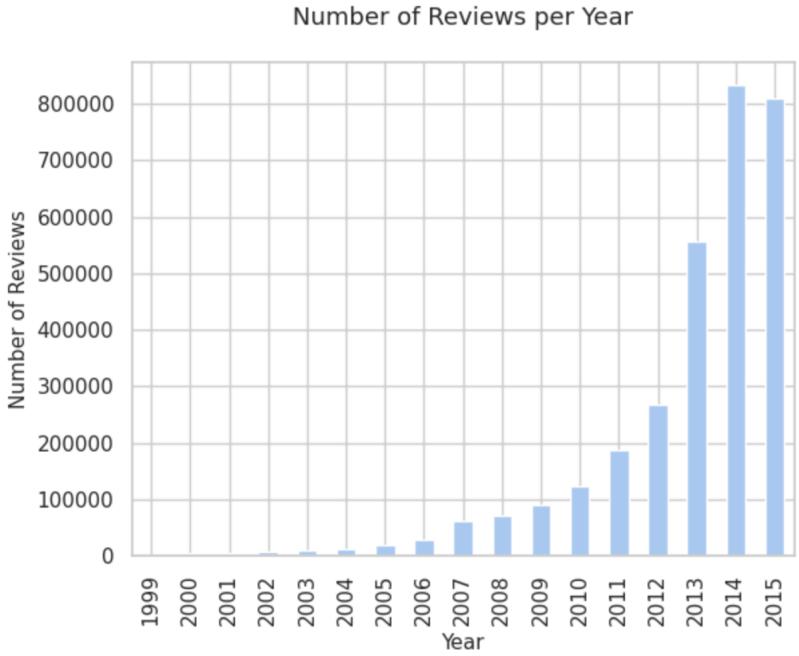

Figure 15: Number of Reviews per Year

This trend indicates growing user engagement and participation in submitting reviews over time. The sharp increase in recent years suggests that more customers are sharing their experiences online, possibly due to the increasing prevalence of e-commerce and the ease of submitting reviews through various platforms.

2.4 Data Quality

2.4.1 Data Completeness and Duplicates

We start by checking the overall structure of the data, looking for duplicates, and assessing the completeness of the dataset. The dataset contains 3,091,024 rows and 15 columns. No duplicate rows were found, indicating that each review entry is unique.

2.4.2 Missing Values

Next, we check for missing values in the dataset. The following columns contain missing values:

- **product_title**: 4 missing values
- **review_headline**: 39 missing values
- **review_body**: 147 missing values
- **review_date**: 24 missing values

These missing values are minimal and can be handled by either imputation or removal without significantly impacting the analysis.

2.4.3 Descriptive Statistics of Numerical Columns

Descriptive statistics provide insights into the distribution and central tendency of numerical columns. These statistics, including mean, median, standard deviation, minimum, and maximum, help in understanding the characteristics of the data.

2.4.4 Unique Values and Data Types

We also examine the number of unique values in each column and check the data types. Some columns, such as **marketplace** and **product_category**, have only one unique value, indicating they do not provide additional information and can be dropped. Additionally, categorical variables like **vine** and **verified_purchase** have been encoded for further analysis.
From the output, we see the following:

- **marketplace**: 1 unique value
- **customer_id**: 2,152,773 unique values
- **review_id**: 3,091,024 unique values
- **product_id**: 185,774 unique values
- **product_parent**: 166,173 unique values
- **product_title**: 167,864 unique values
- **product_category**: 1 unique value
- **star_rating**: 5 unique values
- **helpful_votes**: 895 unique values
- **total_votes**: 934 unique values
- **vine**: 2 unique values
- **verified_purchase**: 2 unique values
- **review_headline**: 1,635,683 unique values
- **review_body**: 2,894,833 unique values
- **review_date**: 5,904 unique values

By addressing data completeness, duplicates, and missing values, we ensure the dataset is clean and ready for further analysis.

3 Data Preparation

In this phase, we will apply knowledge collected in the Data Understanding in order to develop a viable dataset to train the model on.

3.1 Data Selection

In order to understand what data is relevant to us, first we need to address data mining goals, as well as some data mining constraints. We want to implement a recommendation system, thus raising customer satisfaction, enhancing strategic decision-making, and achieving business objectives.

From this perspective, we should be cautious with new data added from other datasets, as it should be coming from the same source (amazon website), with the same anonymization techniques (all IDs should be generated the same way, all categories preserved). In the current overview, this could be a more up-to-date ratings (e.g. review dates of 2024), more countries (not only the US), or other categories (as we are currently working with only Electronics). However the selected dataset contains no description how exactly it was collected, anonymized, and post-processed, resulting

in a high possibility to ruin dataset quality by adding more rows.

On the other hand, extending the dataset horizontally (adding more columns) is also problematic, due to indexes being anonymized, and votes not representing much of a source identifiers.

For all the reasons above, we decided not to enhance the selected dataset with new data from other sources, as we may potentially ruin quality dramatically.

3.2 Data Cleaning

To start the cleaning process, first, we need to understand a bit about what data recommendation systems use in general (reprocessing techniques may vary dramatically). In modern systems there are two techniques, one is personalized systems, and the other is generalized. Intending to elevate customer satisfaction, it makes sense to turn our attention more into a personalized one.

In the context of personalized recommendations (unique for each user), we only need to store the rating the user gave, and to what product it was given, resulting in tuples 'customer_id', 'product_id', and 'star_rating'. However, to make this data more appropriate and suitable for models, we should clear it from outliers, and make some assumptions about 'the usefulness' and 'actuality' of the reviews given. For that reason, it would make sense to store all columns that could be used as filterers in the next steps.

According to *Data Understanding*, we can say that columns 'marketplace', and 'product_category' could be removed entirely as they only contain 1 unique value throughout all the rows. In addition, we can also discard vine due to huge class dis-balance (can't filter by vine=1, and vine=0 as it would change distributions).

In the given architecture, we also decided not to use 'product_title' as it could not be used as a filter due to its variability and 'text' nature.

3.3 Data Construction

In this section, we will make additional columns that would further be helpful in Integration and Formatting parts. As most of our data consists of just indexes and votes, it makes sense to work with them.

The first column we will add is 'helpfulness' - how many people consider a given review helpful ($\frac{\text{helpful\_votes}}{\text{total\_votes}}$). In case of total votes being equal to 0, we will consider helpfulness also 0. We suppose that this column could be used further in filtering to select more truthful reviews (non-emotional, with less bias overall).

The other one we made is 'review_full_text'. We found inconsistencies in reviews as sometimes people write full reviews in 'review_headline' and leave 'review_body' empty. To make it consistent, we decided to combine both columns into one that will represent all text data of review.

The last one is 'review_length', just a number of symbols in 'review_full_text', or more mathematically number of symbols in 'review_headline' and 'review_body' plus one (space symbols that separates them in 'review_full_text'). We have an assumption that longer reviews correlates with the ratings, as possibly people were much more

frustrated or upset with a product, so they gave a lower rating and a more detailed (longer) review to help others.

3.4 Data Integration and Formatting

In this section, we will be focusing on formatting and filtering data, as well as getting additional information into a dataset. These sections are usually considered differently, however, due to reasonable resource management, and how related such sections are, we will do them at the same time. We propose 4 different solutions to data formatting, and one of them requires additional data integration.

The first approach – no filtering at all. Maybe there is a chance that raw data is already good enough for a machine learning model to be trained on. The only thing we did, we dropped rows with null values to make the data 'perfect'-clean.

The second approach – basic filtering based on statistics. Here we decided to remove all ratings with value 3 due to its ambiguity (it is a common technique in recommendation systems). In addition, we will consider only reviews that are considered helpful by other users. This should remove spam reviews and provide a more clear and honest understanding of the products we were working with.

In this case, helpfulness is considered as $\frac{\text{helpful\_votes}}{\text{total\_votes}}$. And where this value will exceed 50%, we will count them as helpful. Lastly, we will not trust any unverified purchases. After all such filtering, we left only 18% of the original dataset, however, we made sure that rating distributions remained the same (we got a bit higher standard deviation, which was expected as we removed all reviews with 3 stars).

The third approach – filtering based on text data. This is the part where we will "integrate" new data into a dataset. When we were working with data, we noticed that some reviews had high ratings with highly negative descriptions, and vice versa. Knowing that, we decided to integrate new data into our dataset - sentiment analysis scores. We used lxyuan/distilbert-base-multilingual-cased-sentiments-student from Hugging face. This model is trained specifically to identify whether text has positive, neutral or negative context. We applied this model on the whole filtered dataset (from second approach) and made new three columns 'positive_score', 'neutral_score', 'negative_score'. Further we filtered data even more by these columns. We decided to view reviews with: high positive score and low rating, high negative score and high rating. After a such analysis, we noticed that reviews with rating 4 and 'negative_score' greater than 0.9 have interesting dynamic. They are positive (judging by rating), but comments actually describes how bad product is. For this inconsistency we decided to filter such rows as they may be misleading. What we find interesting, we did not find such correlations with ratings 5, 1 and 2. It is also worth mentioning that the model we used is not ideal, it has some false positives and true negatives which may skew a bit our results.

The forth approach – filtering based on reviews given. As a last step, we decided to remove users and products with insufficient data. This is also a common technique in recommendation systems, as such users and products would not get good predictions due to low data available, in addition they would make a model less stable overall.

All Datasets made with approaches described above with source code for all methods could be found in our GitHub repository.

4 Modeling

4.1 Selecting the modeling technique

We decided to apply Alternating Least Squares method to our problem. This method is pretty easy to implement, but has some drawbacks:

- most of the implementations are very memory insufficient (especially in our case);
- it suffers from **cold start**. But this problem is solved in the data preparation;
- in most implementations it requires to label items and users by numbers;

therefore, we used the implementation of ALS from pyspark, that is optimized for large datasets and does not use the straight forward approach to do the factorization.

4.2 Generating test design

We took the **70%** of the dataset as training data, **15%** as test data and other **15%** of data as validation set. Test set imitates new users with some initial ratings. Validation set is the same, but used only for tuning parameters. For such simple test idea, we do not need any stratification or other techniques.

4.3 Building the model

ALS in pyspark has some important parameters to set. They are described below with their descriptions and values we tried.

- **rank**. This is the dimensionality of the embedding for users and products. We tried different values from **5** to **80**. **30** showed the best result.
- **maxIter**. This is the number of iterations in the method. The best result was achieved with **15** iterations.
- **regParam**. This is the parameter of regularization on embeddings. We tried different values from **0.5** to **1** (shown in 16). For our specific task we needed a lot of regularization, so the value of **0.7** was the best.

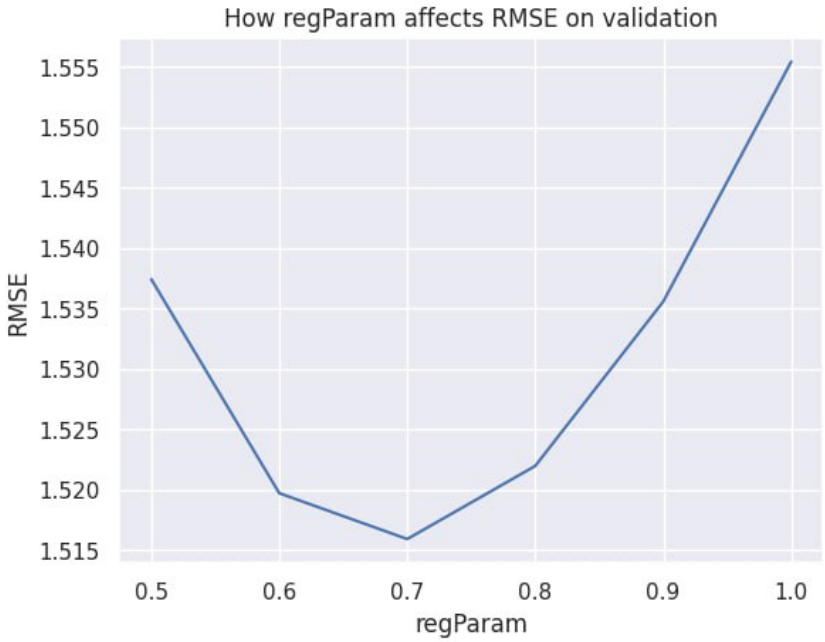

Figure 16: Effect of regParam on RMSE

One of the advantages of the method is its interpretability. ALS, utilizing the ratings of the user, generates the embedding of the user. We can use this embedding with the embedding of any item to produce the prediction on rating using dot product.

4.4 Assessing the model

The model was assesses with **RMSE**. This metric is choosen, because our regressive task is to predict a rating for every pair of users and products in some range. We can therefore assess the model by how it deviates from the answer. Of course, the model assesses the model using only known ratings under the hood.

The best model gives **RMSE=1.60** on test, that is very good, taking into account the fact that every user in training set rated only few products out of hundreds of them available.

5 Evaluation

5.1 Evaluate Results

The trained ALS provide recommendations for users.

Data split	RMSE
Train	1.61
Val	1.55
Test	1.60

Table 1: ALS results

Table 1 shows performance of the model. There are, concerns about how well this model would generalize to real-life scenarios.

5.2 Review Process

The data mining process followed the CRISP-DM methodology quite thoroughly, covering all the key phases from business understanding through data preparation, modeling, and evaluation. However, a few potential gaps or areas for further review can be identified.

Data Collection & Integration:
The dataset used was a pre-collected scraped dataset representing reviews up to 2015. User preferences, product categories, and language styles evolve over time. Deploying a model trained only on this older data could lead to stale or less relevant recommendations.

Data Quality & Relevance:
The main limitation might be that neither the text features nor the time data were directly incorporated into the recommendation algorithm. The models were built primarily on the numerical rating data and user/product IDs.

We also did not explore the possibility of fraudulent product pages. For instance, a seller might update an existing product listing with a completely different product, but retain all the previous reviews. This makes the new product appear more well-reviewed and legitimate to shoppers.

Model Validity:
The ALS model has significant limitations. For one, it does not directly support side features beyond the numerical ratings and user/product IDs. Important information contained in the text of the reviews as well as personal data (which is not present in the dataset) were not utilized by.

Additionally, temporal dynamics reflected in the time data may capture shifts in product popularity or user behavior over time. By exploring other recommendation models, we could potentially find deeper insights and generate more personalized, timely recommendations.

5.3 Determine Next Steps

Based on the assessment results and process review, the decision is to initiate further iterations to improve the recommendation models before putting it to production. Additional iterations are needed to address identified gaps and enhance the overall solution's accuracy and relevance.

Potential areas for further work include:

- scrapping and incorporating more relevant review data;
- revisiting omitted features and introducing new ones;
- investigating alternative recommendation models.

6 Conclusion

In conclusion, this case study showcased a thorough application of the CRISP-DM methodology to develop a recommendation system for Amazon's electronics product reviews. Through the phases of business understanding, data understanding, data preparation, modeling, and evaluation, the team demonstrated their expertise in extracting insights and building predictive models from a large dataset.

While the trained model, ALS, achieved reasonably good performance on the test set, the evaluation highlighted an important limitation – the model are yet insufficient for a real-life application. The reason might be that the model did not directly used some features that otherwise would capture emerging trends.

Furthermore, the dataset is limited by reviews prior to the year 2016. Acquiring and incorporating more recent data would be crucial for maintaining recommendation accuracy and relevance.

In today's competitive online shopping market, providing a truly personalized and excellent user experience is very important. By fixing the identified issues and continuing to improve the recommendation system, Amazon can strengthen its position as the top company.

7 Contributions

	PM	BU	DS	ML	BA	Summation
Daniel Vakhrushev	0.5	0.4	0.1	0	0	1
Polina Zelenskaya	0.1	0	0.9	0	0	1
Ivan Chernakov	0.4	0.6	0	0	0	1
Elina Akimchekova	0	0	0	0	1	1
Roman Voronov	0	0	0	1	0	1
Summation	1	1	1	1	1	

Table 2: Each team member contributions table