CMPT-732 Project Report - Amazon Product Analyzation

December 2023

1 Problem definition

There are mainly two goals for this project.

- Make customized recommendations to customers and sellers
- Analyzing the reviews for product improvement

There are some challenges to accomplish these goals

- The size of data sets are big
- memory leaks if ruining all category at same time
- There are many fields in each json object
- Some fields might be missing
- Need to combine big data sets/ intermediate data frames together
- Compatibility when incorporating third party keyword extraction package.

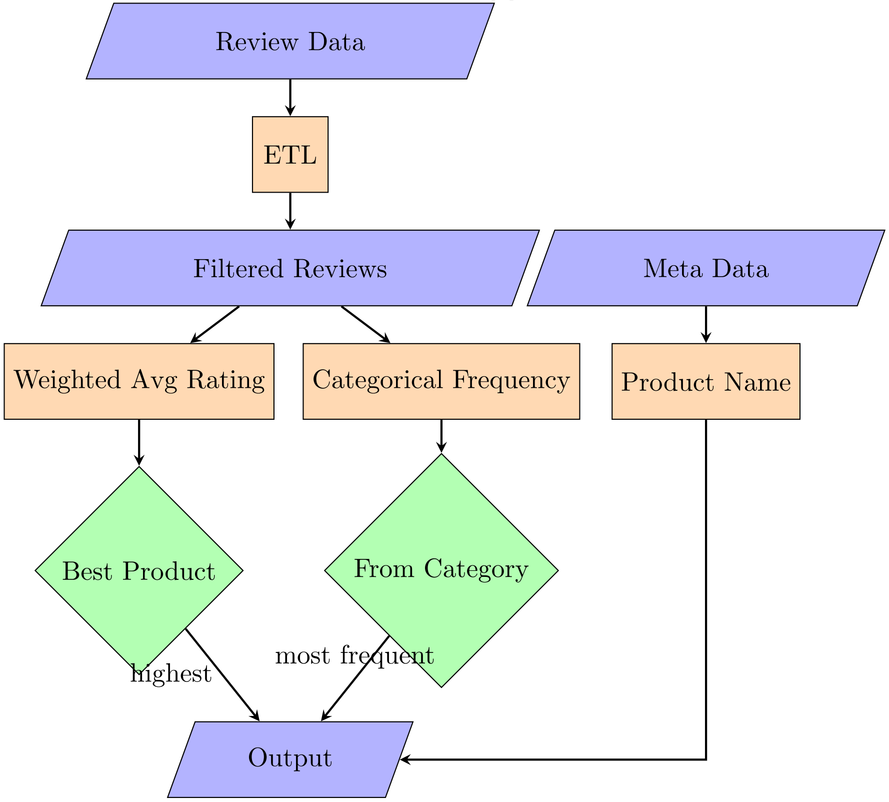
2 Methodology

2.1 ETL

We downloaded the raw data from the data set website data set website as compress file with lines of json object.
We use the PySpark json package to read the .json file. And then filtering out unverified reviews. We also transform the null value from missing field to 0s or empty list for later processing. Finally cache the dataframe or write it to json file for later applications.

2.2 Product recommendation

Flow Chart for Recommendation System [2]



2.3 Review Analyzing

2.3.1 Key Word Extraction

We used rake-nltk 1.0.6 package to perform key word extraction. rake is short for Rapid Automatic Keyword Extraction algorithm. This algorithm is a domain independent keyword extraction algorithm which tries to determine key phrases in a body of text by analyzing the frequency of word appearance and its co-occurance with other words in the text.[1]

We classified reviews into ”GOOD” or ”BAD” by splitting on overall rating greater than 4 and less than 3. And filtering out neutral reviews which are less important than reviews with strong opinion. This also helps to improve our run-time to about 1 minute per category.

2.3.2 Predicting Number of Purchases

The goal of the part is to predict number of purchases of each product from product meta data and ratings we get from previous part. And we are using the predicted value to help the sellers understand that which product they should restock. In this part, we utilized the PySpark.ML package. We performed feature engineering to transform large categorical data **brand** into continuous average of purchases among the same brand. We tested GBT regressor, random forest regressor, and linear regressor. The result of prediction is cached as .csv file for future visualization.

2.4 Dashboard

Our aim is to help both the customer as well as sellers on Amazon and visualize it in such a way that it's easy to understand and yet is very factual. We have built a dashboard using Streamlit. This dashboard has visualizations that we built using Plotly express module. We have done 2 types of visualization here, one focusing on sellers and the other on the customers. For sellers we have plotted pie-charts which depict a) The customer preferences based on the three categories, and b) the Top 100 customer preferences for every category. We have incorporated Streamlit-word cloud to build a word cloud from the keywords we extracted from the reviews. For Customers, we have plotted a scatter plot that depicts the products with their weighted average. This will help the customers decide which product they should buy.

3 Problems

3.1 Faster Null Value Replacement

After reading the json file, there are null values in the missing fields, for example, reviews with no pictures, or comments. Our first attempt to solve this problem is by using user-defined functions to replace the null values, but this can be time-costly. So we used PySpark coalesce function which return value of the first column that is not null, and also we have used fillna() method to replace the NUll values with 0.

3.2 Key Word Extraction

Error when toPandas() function, which is cause by null values. This behaviour is different than PySpark sql dataframe, so we also performed ETL on meta data before extracting key word frequency.

3.3 Feature Selection and Engineering

When we want to predict the number of purchase of each product by their meta data and ratings. The **brand** categorical feature is very large in number, which increase the training time and possibility of over fitting. So instead of using **brand** directly, we created another feature which is the averaged purchases among the same brand. With this new feature, we will be able to test with linear regression.

4 Results

4.1 Visualizations

4.1.1 Top products per category

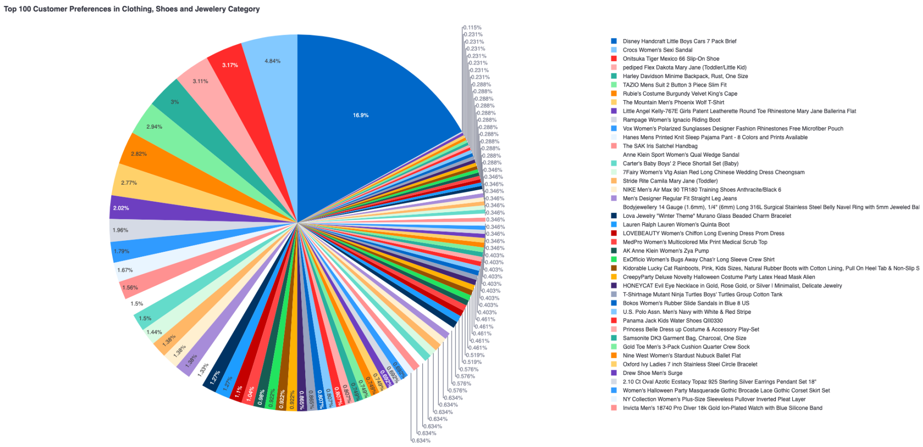


Figure 1: Top-100 Products in Clothing, Shoes and Jewellery

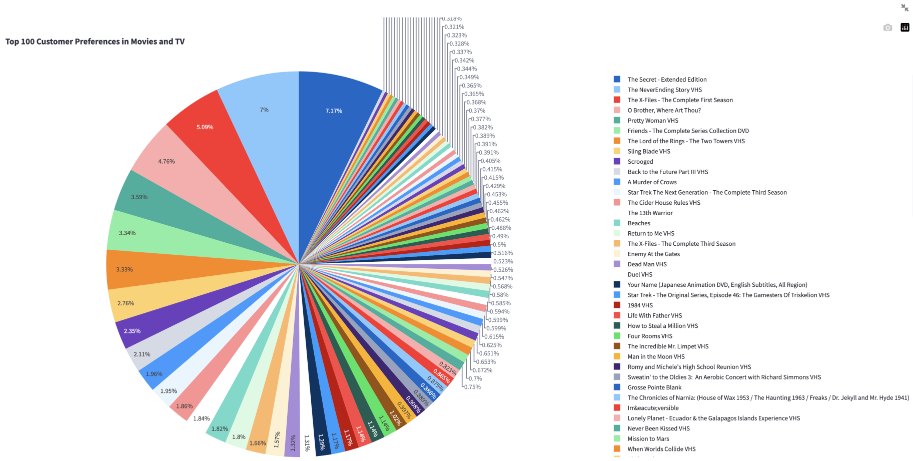


Figure 2: Top-100 Products in Movies and TVs

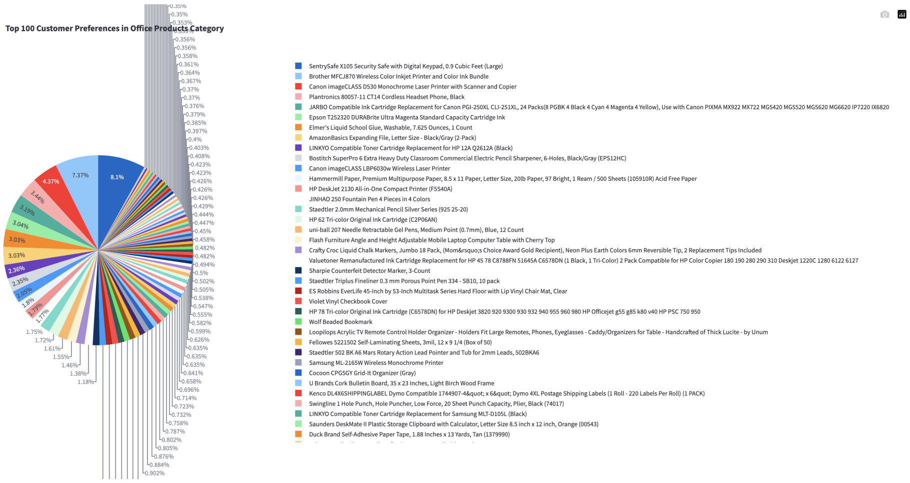


Figure 3: Top-100 Products in Office Products

4.1.2 Percentage of Purchases from Categories

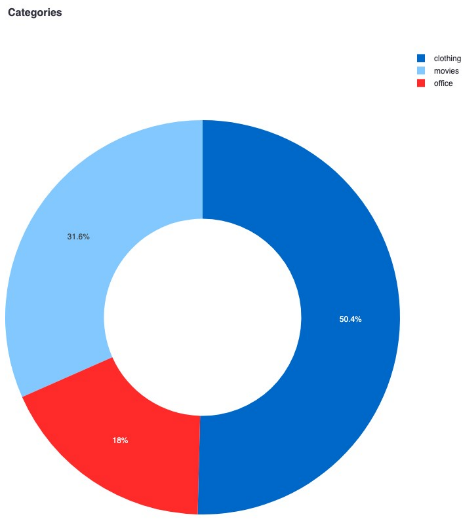
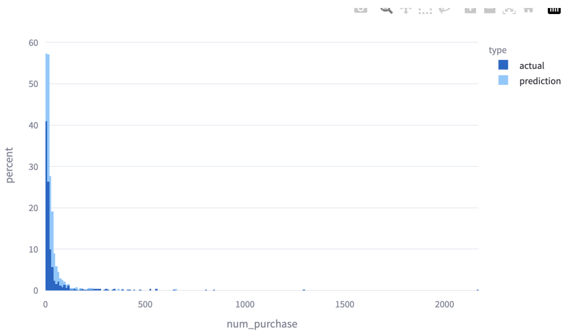
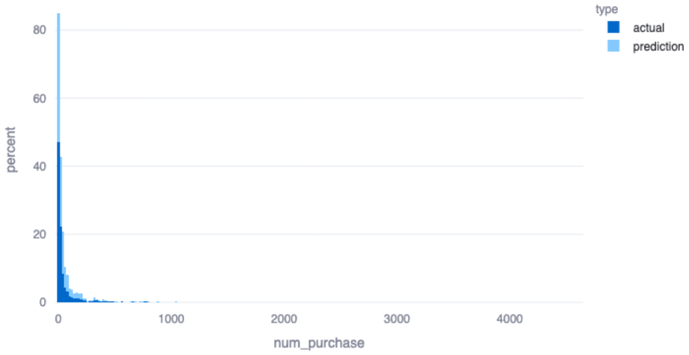
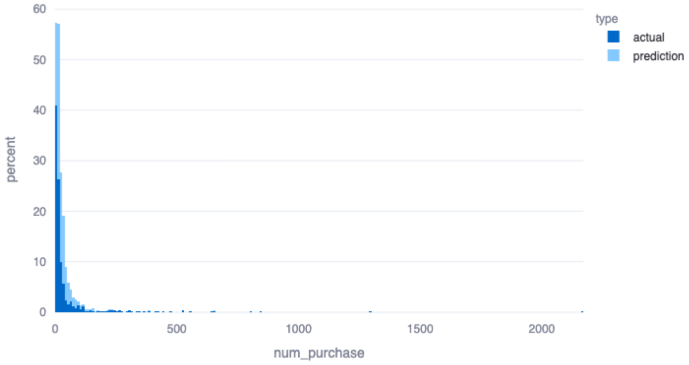


Figure 4: Pie Chart of Purchase Percentage of Selected Number of Customers

4.1.3 Two-Colour Histogram of Prediction VS Actual Purchases for Each Category







4.1.4 Word Cloud for Review Texts

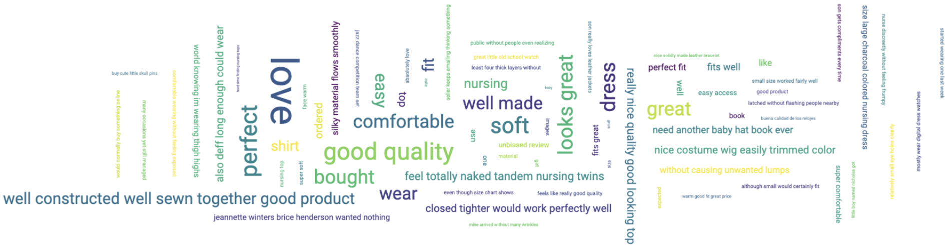


Figure 5: Clothing

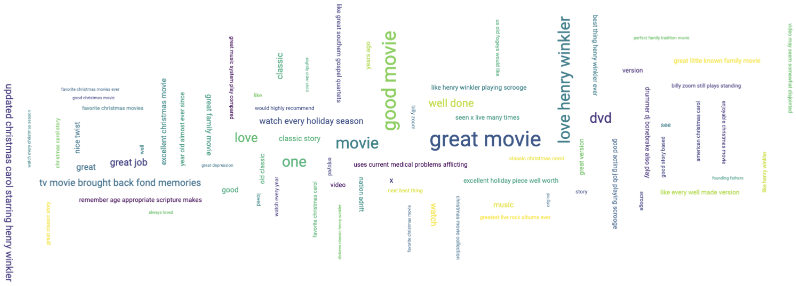


Figure 6: Movies and TVs

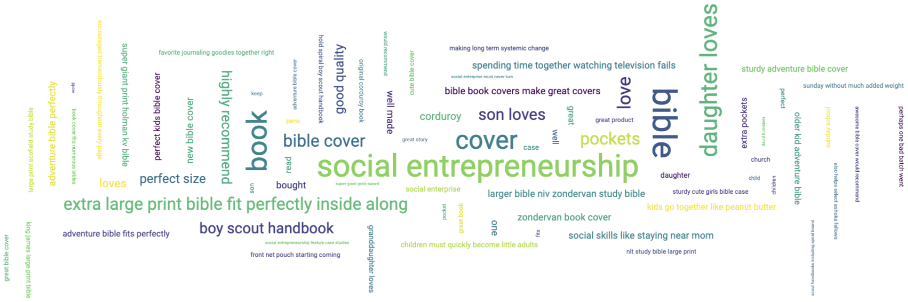


Figure 7: Office Produts

5 Summary

- Getting the data: downloaded. 0
- ETL: 2
- Problem: 3
- Algorithmic work: 3
- Bigness/parallelization: 2
- UI: 3
- Visualization: 5
- Technologies: 2

6 Reference

1. https://pypi.org/project/rake-nltk/
2. https://docs.streamlit.io/library/api-reference
3. https://github.com/rezaho/streamlit-wordcloud streamlit
4. https://plotly.com/python/plotly-express/
5. LaTeX Graphics using TikZ: A Tutorial for Beginners (Part 3)—Creating Flowcharts - Overleaf, Online LaTeX Editor