A Recommendation System for ModCloth

Newton Tran

What Are Recommendation Systems?

- Type of machine learning that utilizes data to suggest items most relevant to the user
- Two main types: collaborative filtering and content-based
  - Former uses user similarity, whereas the latter is exclusive to the item's features (i.e. focuses on a specific user's preferences rather than group or (type)
- Two types of collaborative filtering: user=based and item-based
  - In the former, if any two given users have similar preferences or behaviors, they are likely to prefer items
  - In the latter, there is more focus on items that users have already interacted with

Why Recommendation Systems?

- With so many choices, recommendation systems can streamline the customer experience
- From a business standpoint, they can boost sales since they:
  - Increase personalization for the consumer → higher customer retention
  - Lead to more cross-selling
  - Lower marketing costs

Problem Statement

- ModCloth is a brand that specializes in vintage-inspired women’s apparel, particularly in dresses. The goal is to build a recommendation system that recommends a user three products.

Data

- **Addressing Marketing Bias in Product Recommendations**
  Mengting Wan, Jianmo Ni, Rishabh Misra, Julian McAuley, *WSDM*, 2020
- Under `df_modcloth.csv`
- The data contains over 99,893 reviews of 1020 unique products sold from ModCloth, from 44,783 unique users
- Data spans over 9 years from 2010 – 2019
- Five numerical features: `item_id, rating, size, year, split`
- Seven object-type features: `user_id, timestamp, fit, user_attr, model_attr, category, brand`

Missing Values

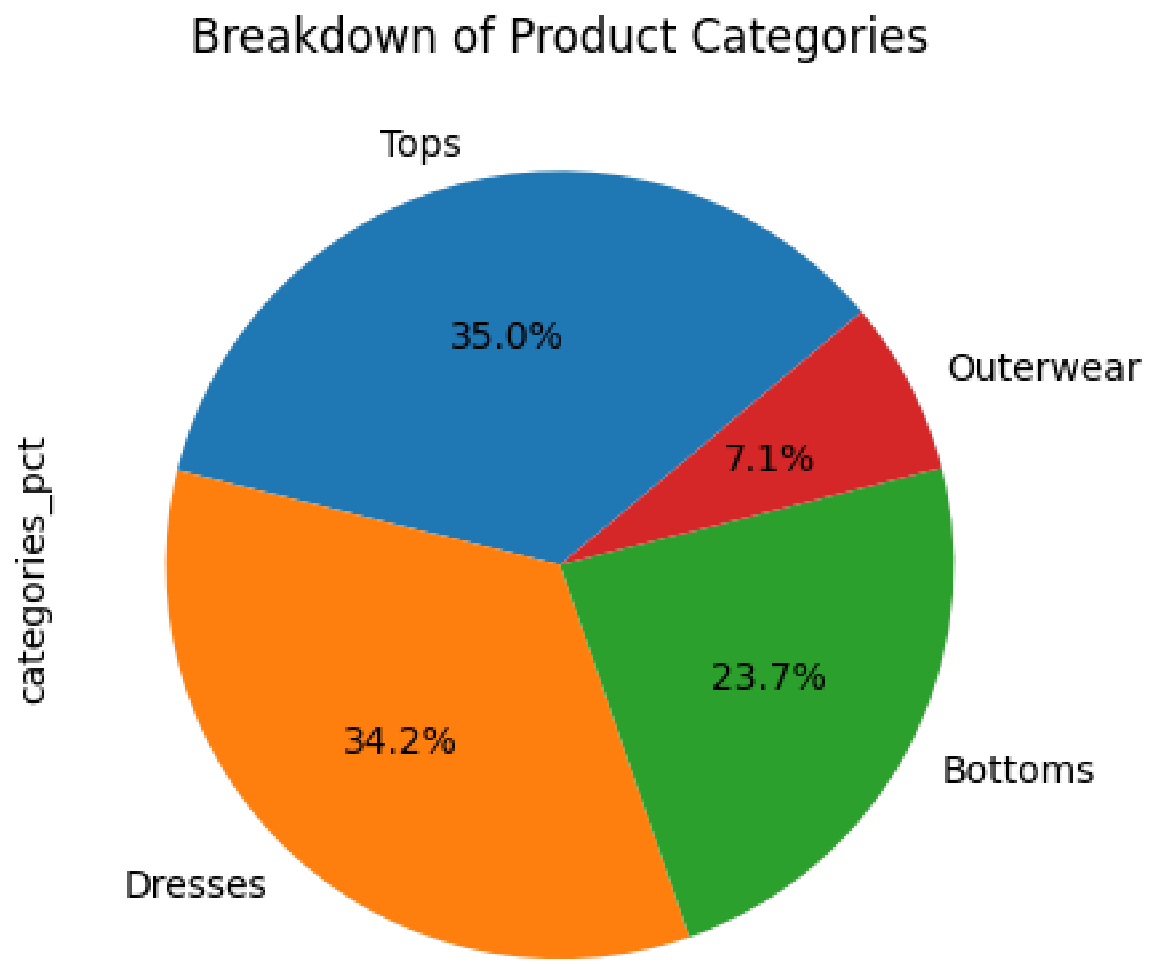
- `brand`, `size`, `fit`, `user_attr`, and `user_id` contained missing values

	column_name	percent_missing
brand	brand	74.059243
size	size	21.783308
fit	fit	18.525823
user_attr	user_attr	8.375962
user_id	user_id	0.001001
item_id	item_id	0.000000
rating	rating	0.000000
timestamp	timestamp	0.000000
model_attr	model_attr	0.000000
category	category	0.000000
year	year	0.000000
split	split	0.000000

Exploratory Data Analysis

Product Categories

Tops, bottoms, and dresses are the predominant categories



Summary Statistics of the Number of Ratings Per Unique Item

- The average number of ratings is about 98, but this starkly contrasts to the median of 17 ratings
- This also widely varies with a standard deviation of around 216 ratings

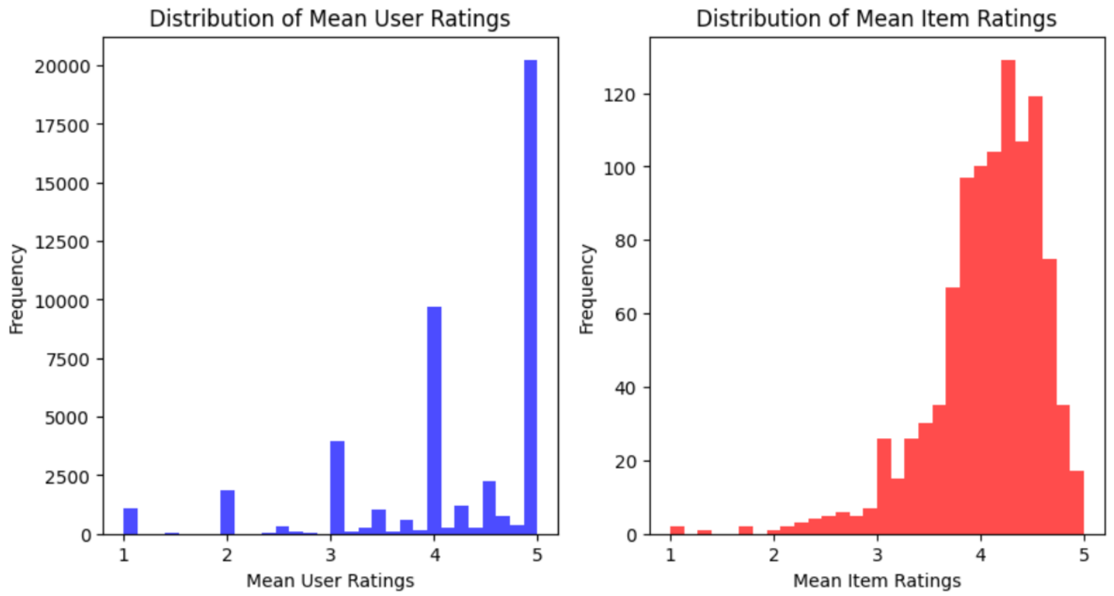
	count
count	1020.000000
mean	97.934314
std	216.416612
min	1.000000
25%	8.750000
50%	17.000000
75%	66.000000
max	1887.000000

Average Rating For Each User

- The average rating that a user gives is around 4, which is close to the median of about 4.17
- This varies by about a standard deviation of 0.54

	rating
count	1020.000000
mean	4.070268
std	0.540718
min	1.000000
25%	3.820599
50%	4.167749
75%	4.452922
max	5.000000

Distributions of Mean User Ratings and Mean Item Ratings



- Both share skewness, however we note that the ratings are concentrated between 4 and 5

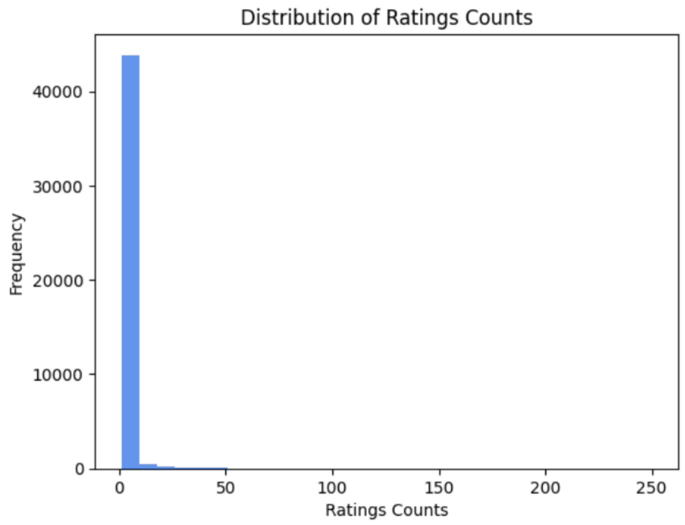
Summary Statistics of the Number of Ratings a User Has Submitted

- The number of ratings submitted ranges from 1 to 250
- On average, there are about 2 reviews submitted from any given user
- This varies by about 6 to 7 reviews

	rating
count	44783.000000
mean	2.230579
std	6.548969
min	1.000000
25%	1.000000
50%	1.000000
75%	2.000000
max	250.000000

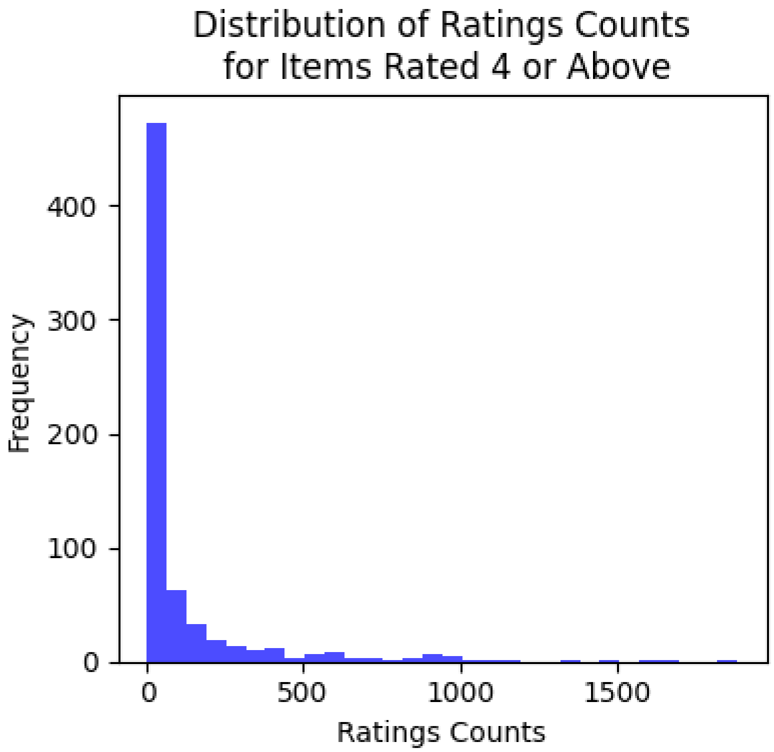
Distribution of the Number of Ratings a User Has Submitted

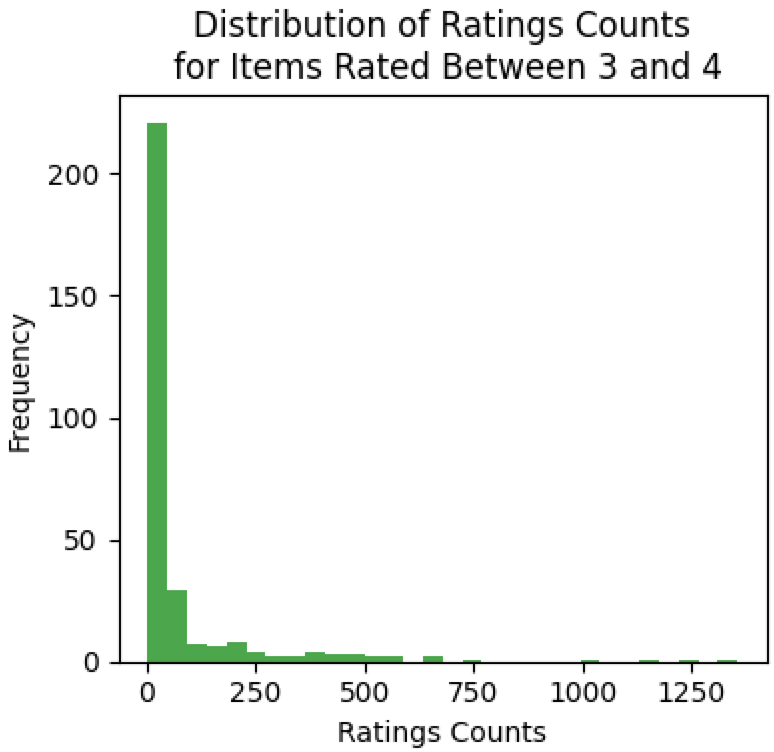
- We see that most users have submitted around 2 reviews
- Extreme right-skewness

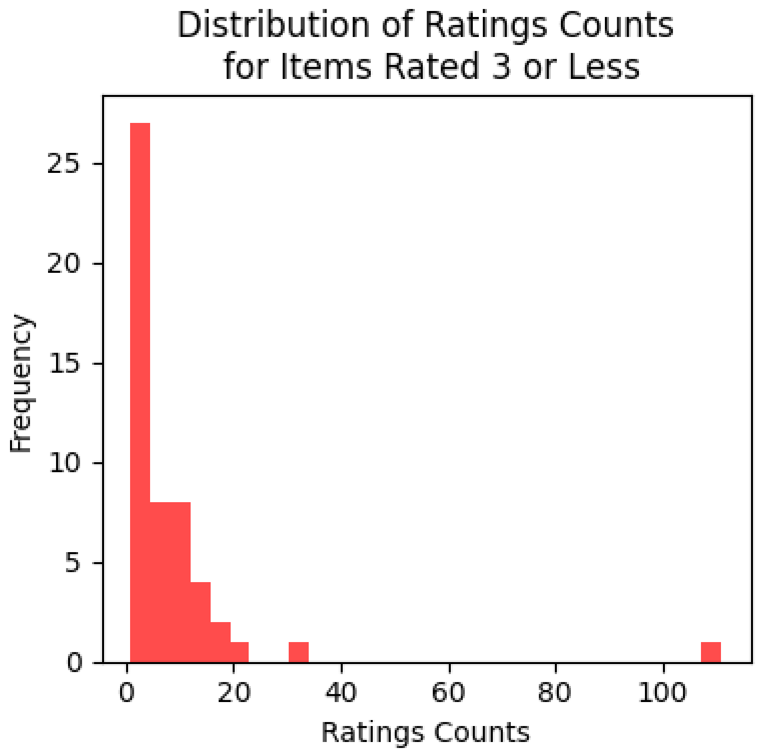


The Number of Ratings Counts, Given the Item's Rating

- In all three cases, they all show right-skewness







Data Wrangling

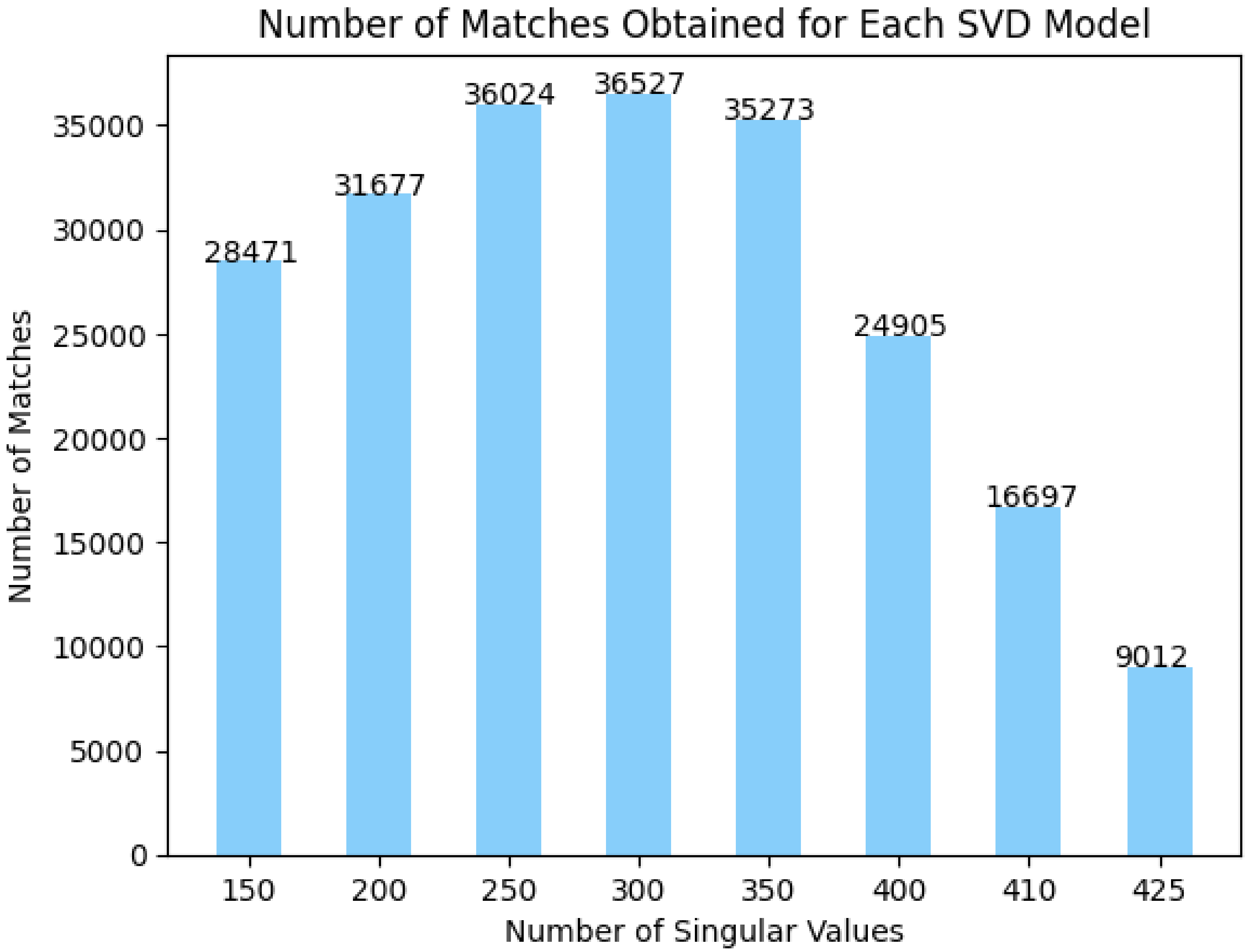
- Four features were extracted: `item_id`, `user_id`, `rating`, and `category`
- Imputed one missing value of `user_id` with ‘Unknown’
- Establish a notion of what is considered a “good” item
  - An item’s average rating was 4 or above
  - The item had at least 24 reviews
- After filtering the data, the dataset was reduced to 93,915 reviews (rows), containing 435 unique items and 43,470 unique users
  - Subsequently saved as `df_modcloth_filtered.csv`

Data Wrangling

- A separate dataset contained the most popular items through basic aggregation, which was grouped by `item_id`, and then sorted by average item rating and number of ratings, each in ascending order, was stored as `pop_items.csv`
- A new feature, `weighted_vals`, was created by multiplying the average item rating by the number of ratings, then by a float between 700 and 1600 if the average item rating was 4 or above (else divided by 1000), and then finally added by a random float generated by 0 to 1
  - Ensures mapping between the item and its respective `weighted_vals` is bijective

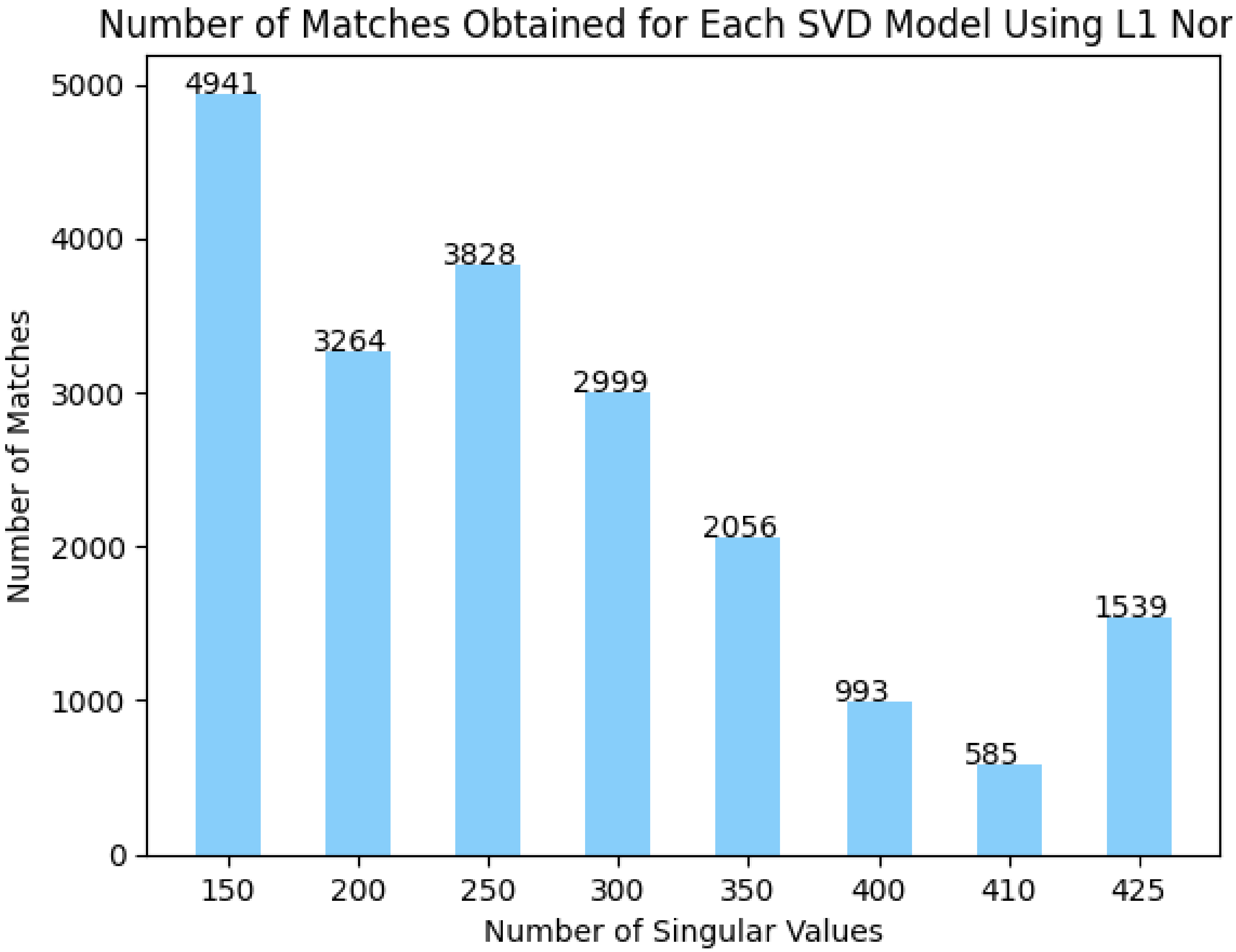
Modeling

- The recommendation system is split into two cases:
  - A user's average rating is 4 or above. In this case, we will utilize item-based collaborative filtering
  - Else, a user's average rating is below 4. In this case, we will only give generic recommendations based on the most popular items
- For the first case, for model training and validation, a 50-50 stratified train-test split was done on the filtered dataset
- A utility matrix was created off from the training dataset
  - Since the ratings were rank-based, Spearman's correlation was preferred
- Singular value decomposition (SVD) was then applied to the utility matrix
- A k-nearest neighbors model of k = 10 was then constructed from that SVD-reduced utility matrix
  - k = 10 was chosen since we wanted to capture a sufficient, but not overly-excessive number of similar items, while also considering the model might capture a product that was already purchased by the user



Model Metrics

- A simple metric in determining the best model is to see how many matches were obtained between the model and the user's actual purchases



Model Metrics

- By default, the k-nearest neighbors model used L2 norm (Euclidean distance), but we wanted to see how the number of matches would change by changing this hyperparameter to L1 norm (Manhattan distance)

Number of Singular Values	RMSE
150	0.128778485257336
200	0.1019257880438485
250	0.08059295336375218
300	0.061961617379721494
350	0.04433289670423865
400	0.02500844225436355
410	0.020314003417670517
425	0.011552216980782718

- The root mean squared error (RMSE) between the original utility matrix and its SVD-reduced version was also calculated

Model Metrics

Conclusion

- Data showed strong signs of skewness, which influenced the results of our recommendation system since items rated 4 or above were heavily emphasized, whereas those rated below 4 were heavily penalized
  - It’s possible that, for example, items rated between 3.7 to 3.9 might be “good” items
- It’s hard to establish a solid notion of what is considered a “good” item since ratings are highly subjective and inconsistent
- We can further understand the behavior of our recommendation system – more so the bias of our data - by applying sentiment analysis on the size and fit features

References

- Addressing Marketing Bias in Product Recommendations
  Mengting Wan, Jianmo Ni, Rishabh Misra, Julian McAuley
  WSDM, 2020

- IBM. “Collaborative Filtering.” Ibm.com, 21 Mar. 2024, www.ibm.com/think/topics/collaborative-filtering.

- “Think | IBM.” Ibm.com, 2024, www.ibm.com/think/topievdelo.com/amazons-recommendation-algorithm-drives-35-of-its-sales/cs/collaborative-filtering. Accessed 16 Dec. 2024.

- Evdelo. “Amazon’s Recommendation Algorithm Drives 35% of Its Sales.” Evdelo, 2020, evdelo.com/amazons-recommendation-algorithm-drives-35-of-its-sales/.

- Whiting, Rick. “Researchers Solve Netflix Challenge, Win $1 Million Prize | CRN.” Crn.com, 2024, www.crn.com/news/applications-os/220100498/researchers-solve-netflix-challenge-win-1-million-prize.