Recommendation System

Kaneesha Dawood

Overview

Background

Project Objective

The Dataset

Recommender Models

Key Insights

Evaluation

Background

Key Problem: Having to decide which product to purchase online

- unlimited product choices

- insufficient time

Target user: e-commerce consumer

Project objective

Build a recommender system for the e-commerce user using Amazon's Electronic Product category.

Why?

- To help the user find related products
- To help the user explore new items
- Improve user decision making
- Increase user engagement

Amazon Electronics Dataset

Attributes:

- user_id: unique id for each user
- item_id: unique id for each product
- rating: from 1.0 - 5.0
- Category: specific class of the electronic product category

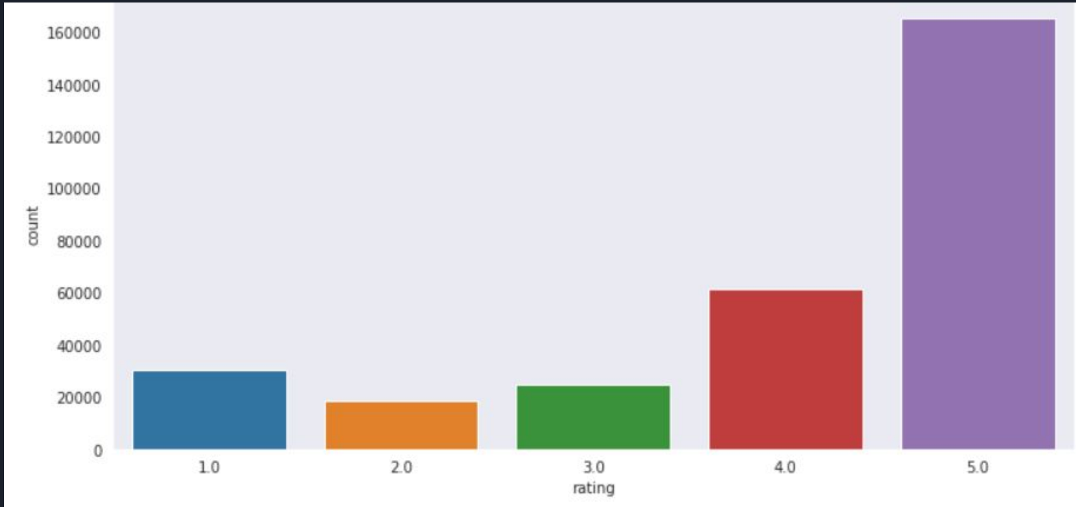
	item_id	user_id	rating	category
0	0	0	5.0	Portable Audio & Video
1	0	1	5.0	Portable Audio & Video
2	0	2	3.0	Portable Audio & Video
3	0	3	1.0	Portable Audio & Video
4	0	4	2.0	Portable Audio & Video

Rating

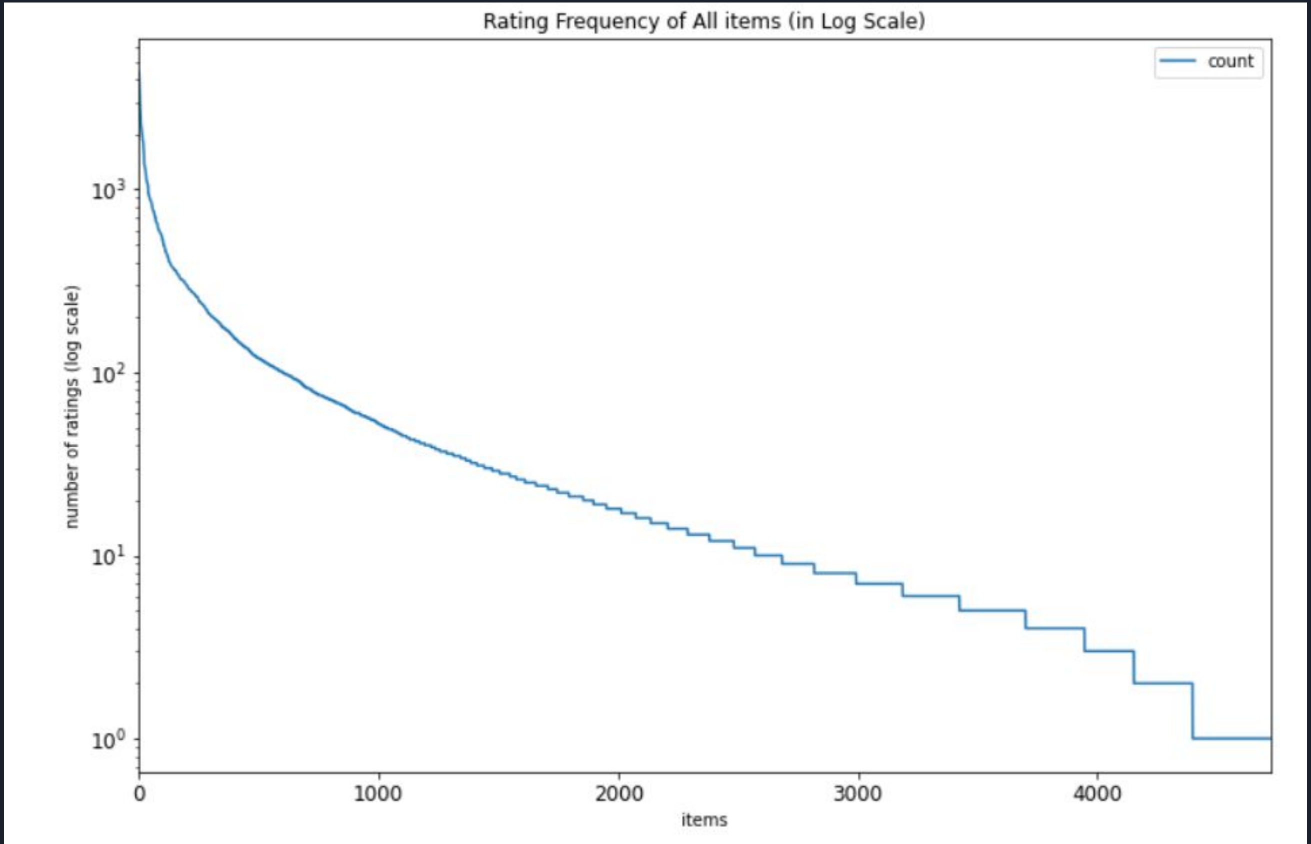
Rating is one of the most common metric used in recommenders.

Rating is used to gain explicit feedback from the user.

Ratings provide an insight into the user's likes and dislikes of a product.



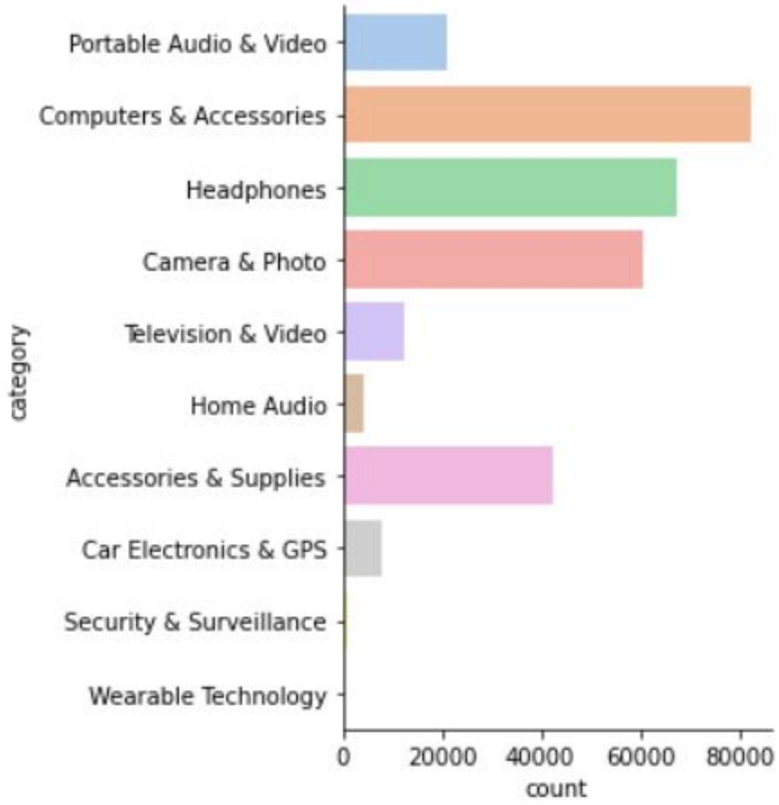
Items Vs User Rating



Category

- High demand for Computers & Accessories (27.4%) and Headphones (22.5%)

- Low demand for Security & Surveillance and Wearable Technology



Train - Test Split

The data is distributed into two sets on a 70:30 ratio:

- Train set
- Test set

```
Shape of training data:  (210000, 4)
Shape of testing data:  (90000, 4)
```

Model 1: User-based Collaborative Filtering

- The User-based collaborative model finds similar users and gives a recommendation based on what other people with similar patterns appreciated. Assumes that similar users will like similar items

- User interactions imply whether the user had a positive or negative experience

- Advantages:

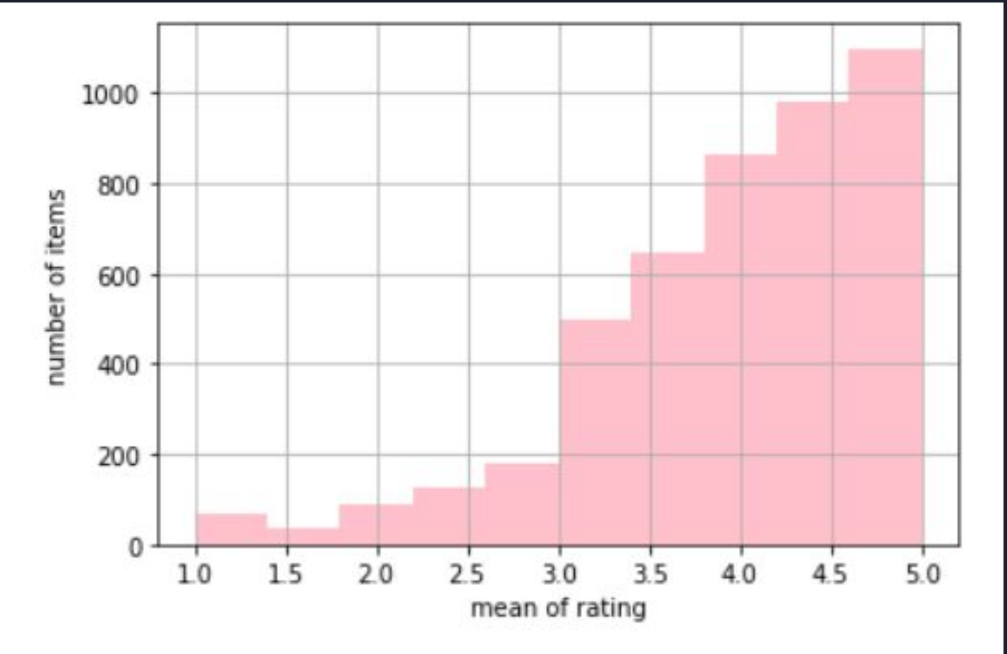
  No Domain knowledge necessary

  Helps users discover new interests

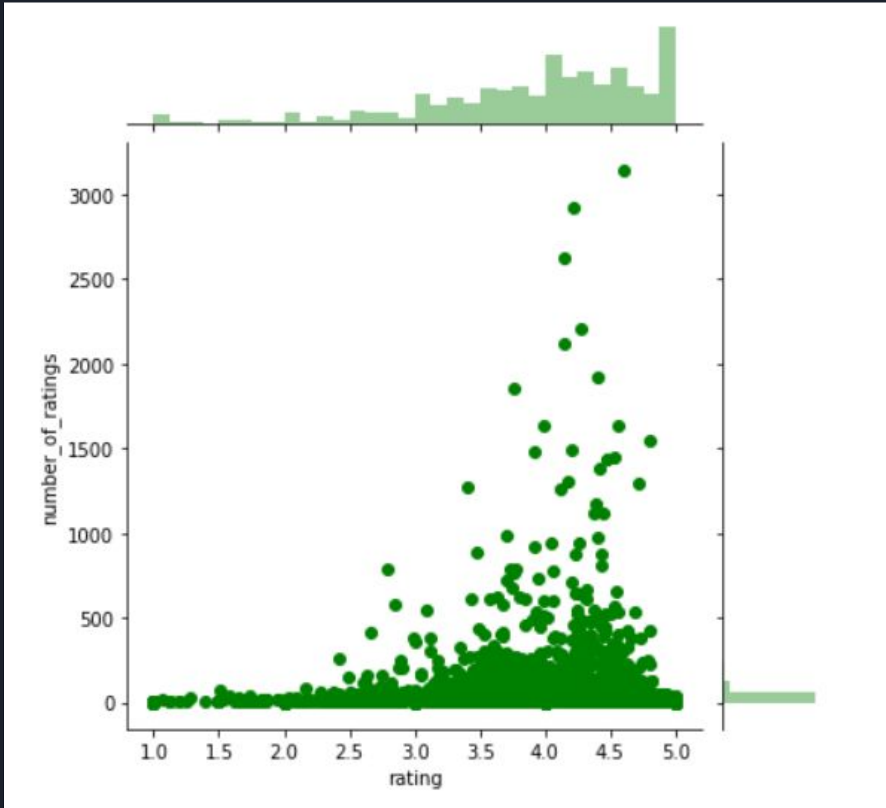
- Disadvantages: The cold Start problem

Approach

- Read the training dataset
- Compute the mean rating of the item
- Calculate the number of ratings per item
- Visualize the distribution of ‘rating’ and ‘number_of_ratings’
- Find the correlation
- Get a threshold of users rated more than 15 items through a User Matrix
- Set an index
- Singular Value decomposition
- Compute Predicted Ratings
- Recommend items to the user



1000
800
600
400
200
0
number of items
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
mean of rating



3000
2500
2000
1500
1000
500
0
number_of_ratings
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
rating

Recommendations

- All three users are given different product recommendations based on their past rating choices

- RMSE : 0.353

Recommended items for user(user_id = 1):

Recommended Items	user_ratings	user_predictions
3189	0.0	2.067316
1670	0.0	2.067316
4559	0.0	2.067316
4385	0.0	2.067316
3698	0.0	2.067316

Recommended items for user(user_id = 2):

Recommended Items	user_ratings	user_predictions
1780	0.0	2.067316
1119	0.0	2.067316
3630	0.0	2.067316
2435	0.0	2.067316
2284	0.0	2.067316

Recommended items for user(user_id = 3):

Recommended Items	user_ratings	user_predictions
314	0.0	-4.911435e-16
897	0.0	-4.911435e-16
1725	0.0	-4.911435e-16
1455	0.0	-6.139294e-16
3630	0.0	-6.139294e-16

Model 2: Popularity based Recommender

Recommends to users based on how popular those items are among other users. It is a non-personalised recommender system and these are based on frequency counts.

Findings: Item_id 2125 is the most popular item

The list of recommendations for the user_id: 10

	user_id	item_id	score	rank
2336	10	2340	3145	1.0
1574	10	1575	2919	2.0
1882	10	1886	2627	3.0
2480	10	2486	2212	4.0
2125	10	2129	2115	5.0

The list of recommendations for the user_id: 100

	user_id	item_id	score	rank
2336	100	2340	3145	1.0
1574	100	1575	2919	2.0
1882	100	1886	2627	3.0
2480	100	2486	2212	4.0
2125	100	2129	2115	5.0

The list of recommendations for the user_id: 150

	user_id	item_id	score	rank
2336	150	2340	3145	1.0
1574	150	1575	2919	2.0
1882	150	1886	2627	3.0
2480	150	2486	2212	4.0
2125	150	2129	2115	5.0

KNN Recommender

Finds the k most similar items to a particular user based on a given distance metric.

Ex: k most similar item for item_id 150

```
Based on item rating, for  7  average rating is  5.0
The first similar item is  3683  average rating is  5.0
The second similar item is  3703  average rating is  5.0
```

	precision	recall	f1-score	support
1	0.09	0.01	0.01	8446
2	0.07	0.00	0.01	5163
3	0.08	0.00	0.01	7146
4	0.21	0.02	0.03	17128
5	0.58	0.97	0.73	52117
accuracy			0.57	90000
macro avg	0.21	0.20	0.16	90000
weighted avg	0.39	0.57	0.43	90000

```
Accuracy:  0.5666
MSE:  2.5934777777777778
```

Evaluation & Recommendations

Metrics:

RMSE

Precision and Recall

Correlation

Best Model: User-based Collaborative Filtering (lower RMSE 0.353)

Thank you!

Github: https://github.com/ktdawood

LinkedIn:
https://www.linkedin.com/in/kaneeshadawood/