OLIST

Recommendation System

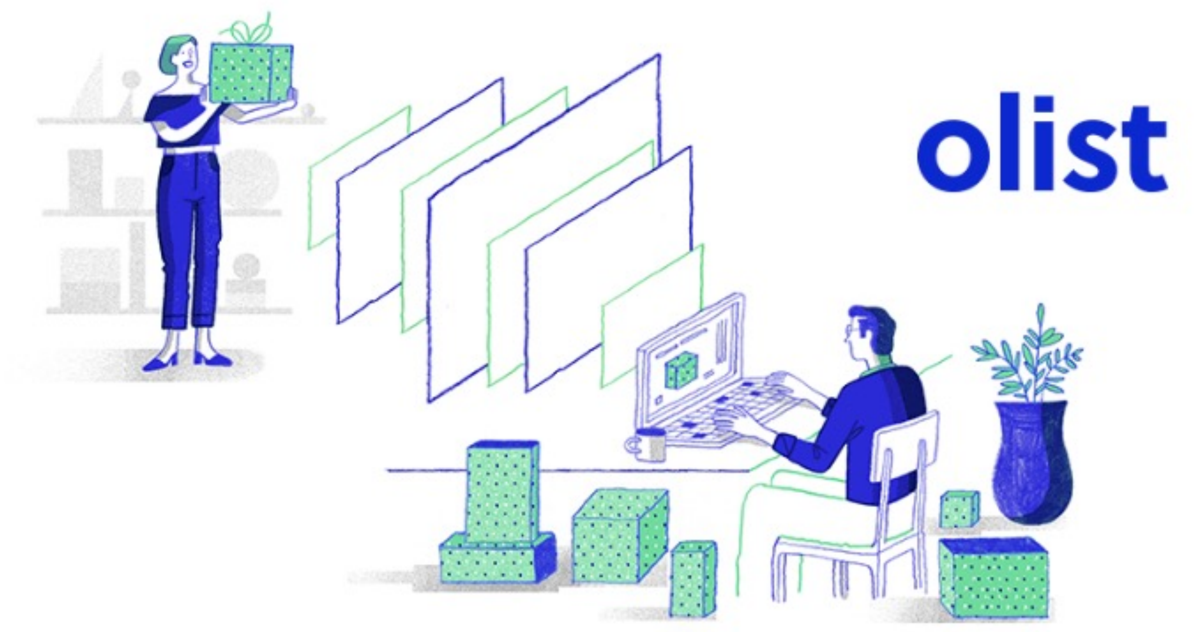
Satria F. B**aga**skara
Dibimbing Data Scientist Batch 18

CONTENTS

OBJECTIVE

1. Olist needs recommendation system to increase sales by **give nearest suggestion products based on customer interest and preferences.**
2. The available data is enough for user-based data like product id, customer id and product's rating.

PROBLEM BACKGROUND

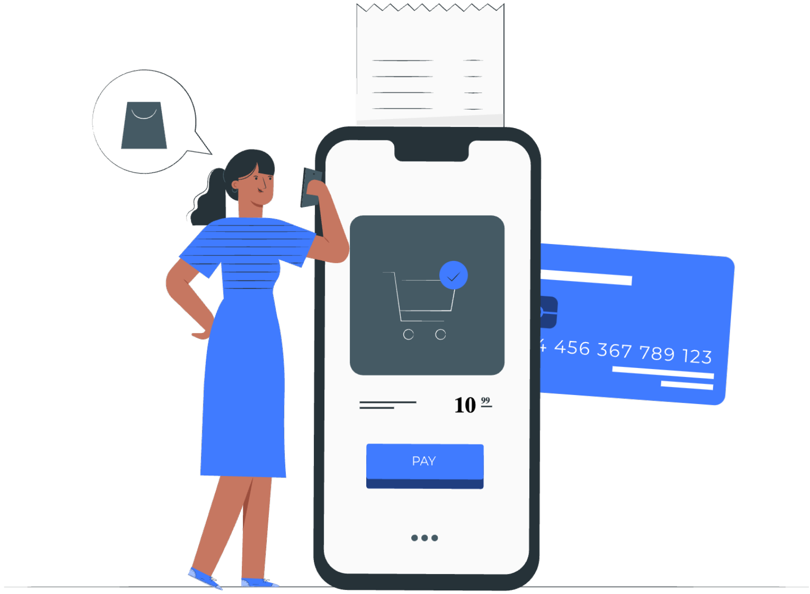


- Recommendation system is important for e-commerce companies.
- Too many options of products makes confuses
- Simply suggest most likeable of customer’s interest and needs

Dataset source: https://www.kaggle.com/datasets/olistbr/brazilian-ecommercev

COLLABORATIVE FILTERING

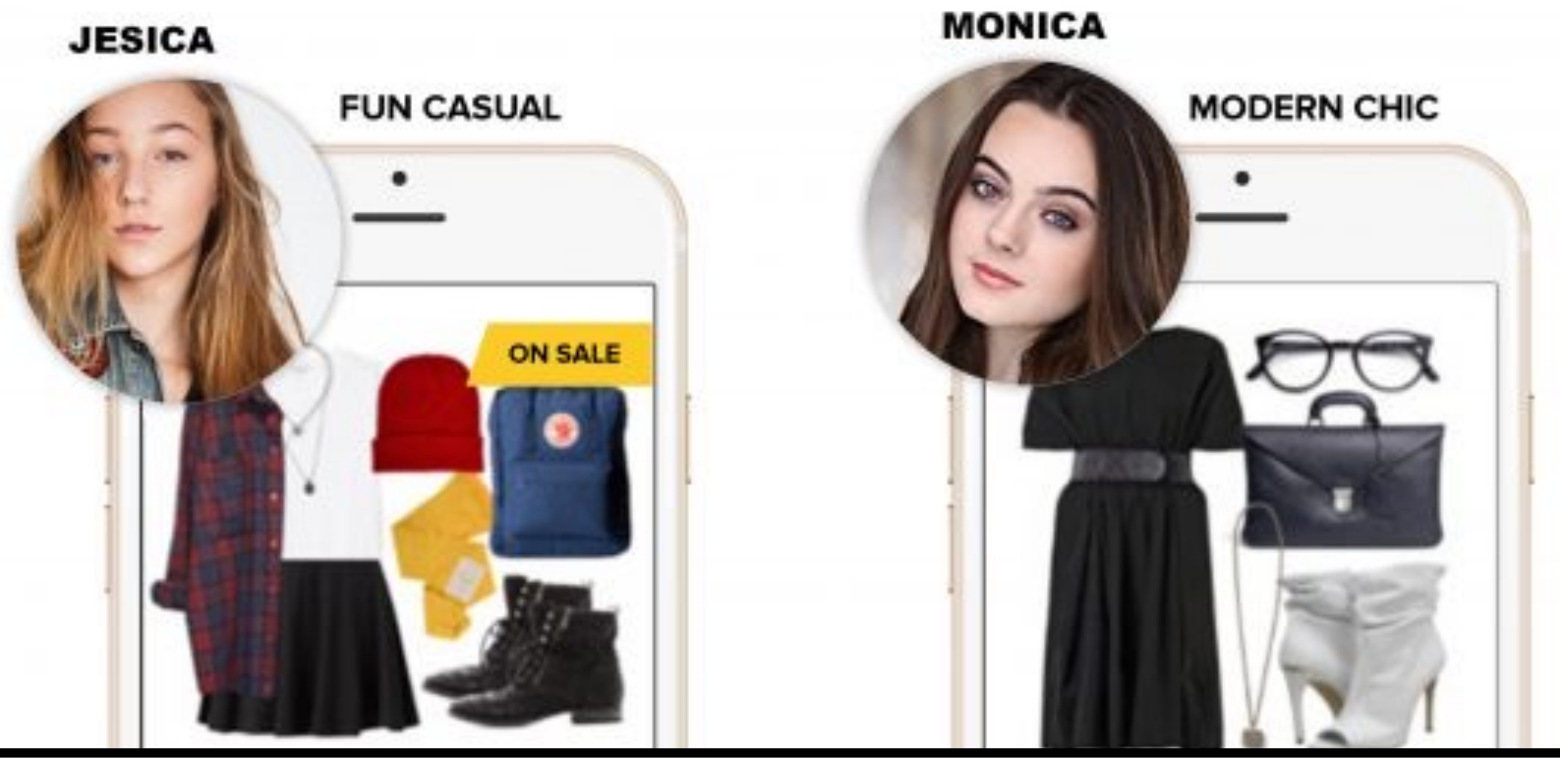
User-based Interest



WHAT KIND OF PRODUCT THAT I LIKE?

COLLABORATIVE FILTERING

User-based Interest



Giving Recommendation based on their interest and preferences

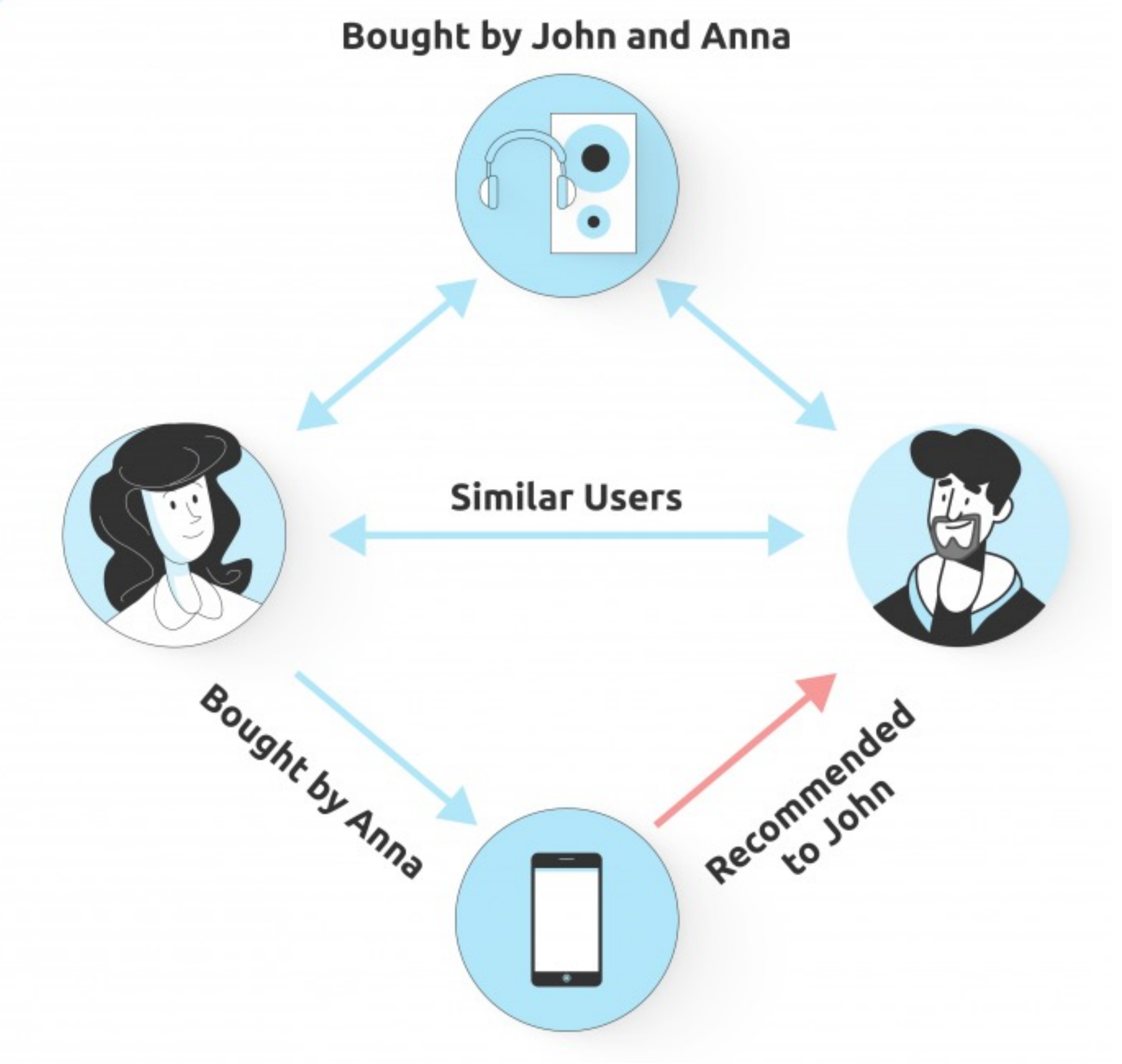
(Personalized Purchasing Experience)

https://nextlevel.ai/portfolio-view/personalized-shopping/

COLLABORATIVE FILTERING

User-based Interest

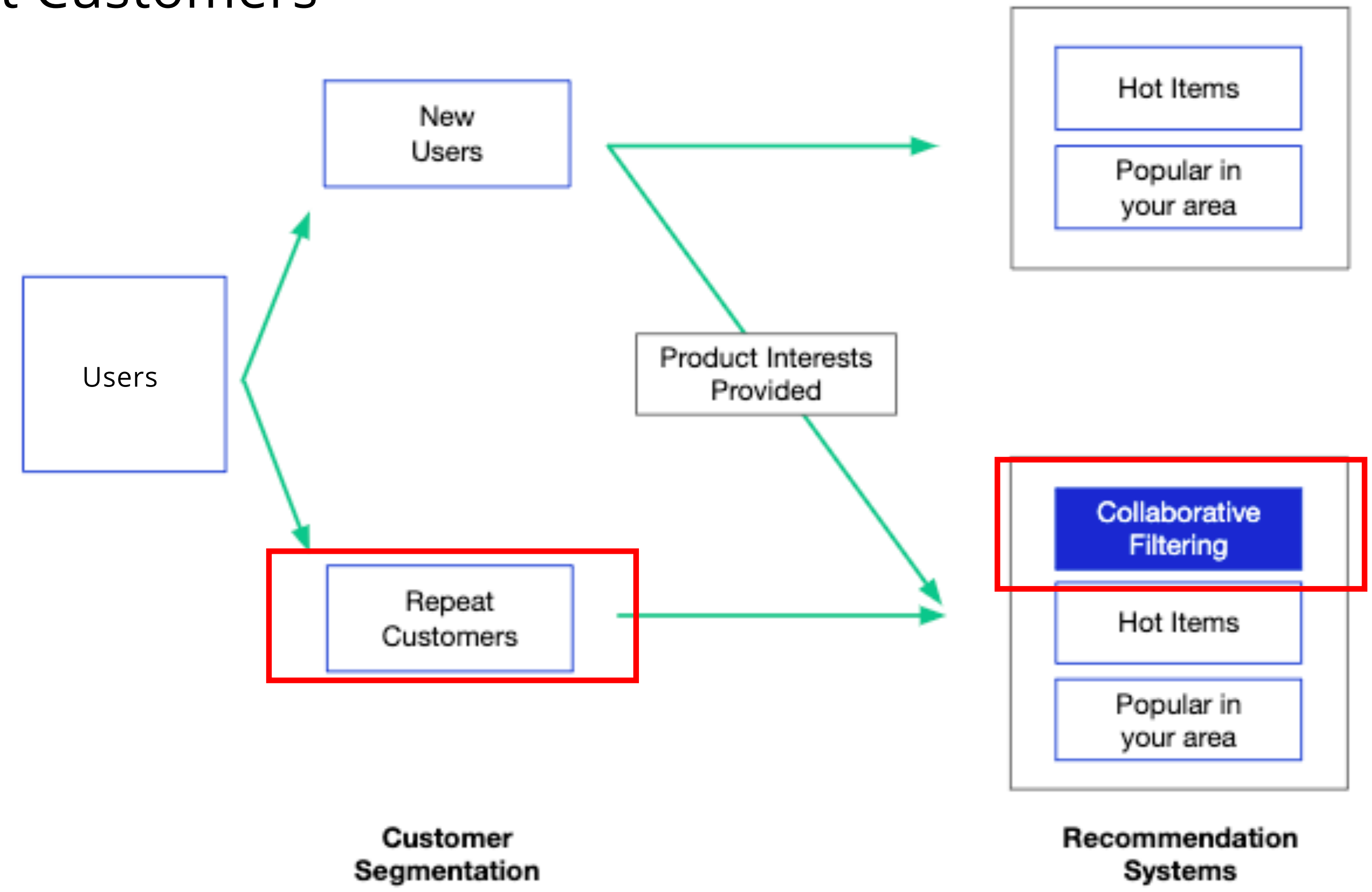
Similar Users Interest
Collaborative Filtering



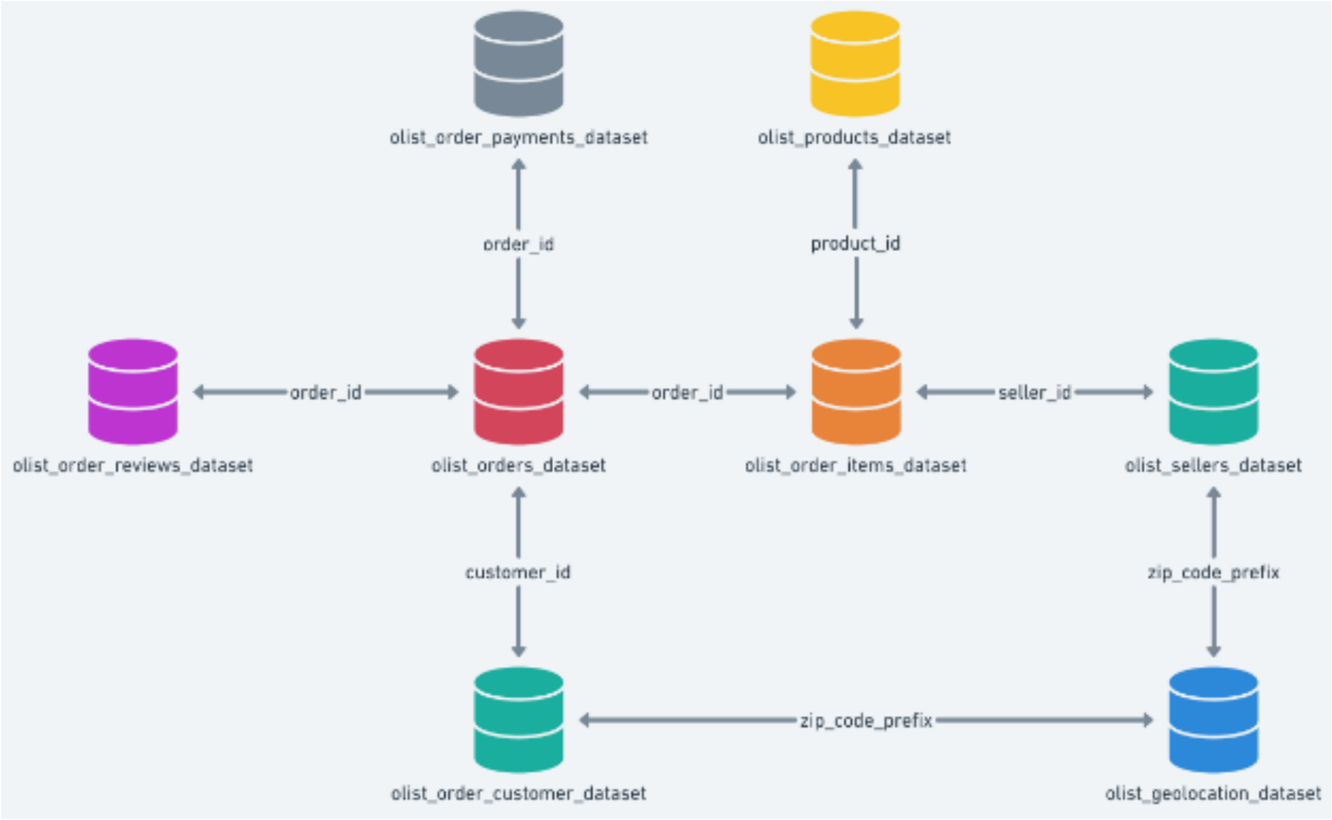
https://influencermarketinghub.com/ecommerce-recommendation-system/

COLLABORATIVE FILTERING

for Repeat Customers

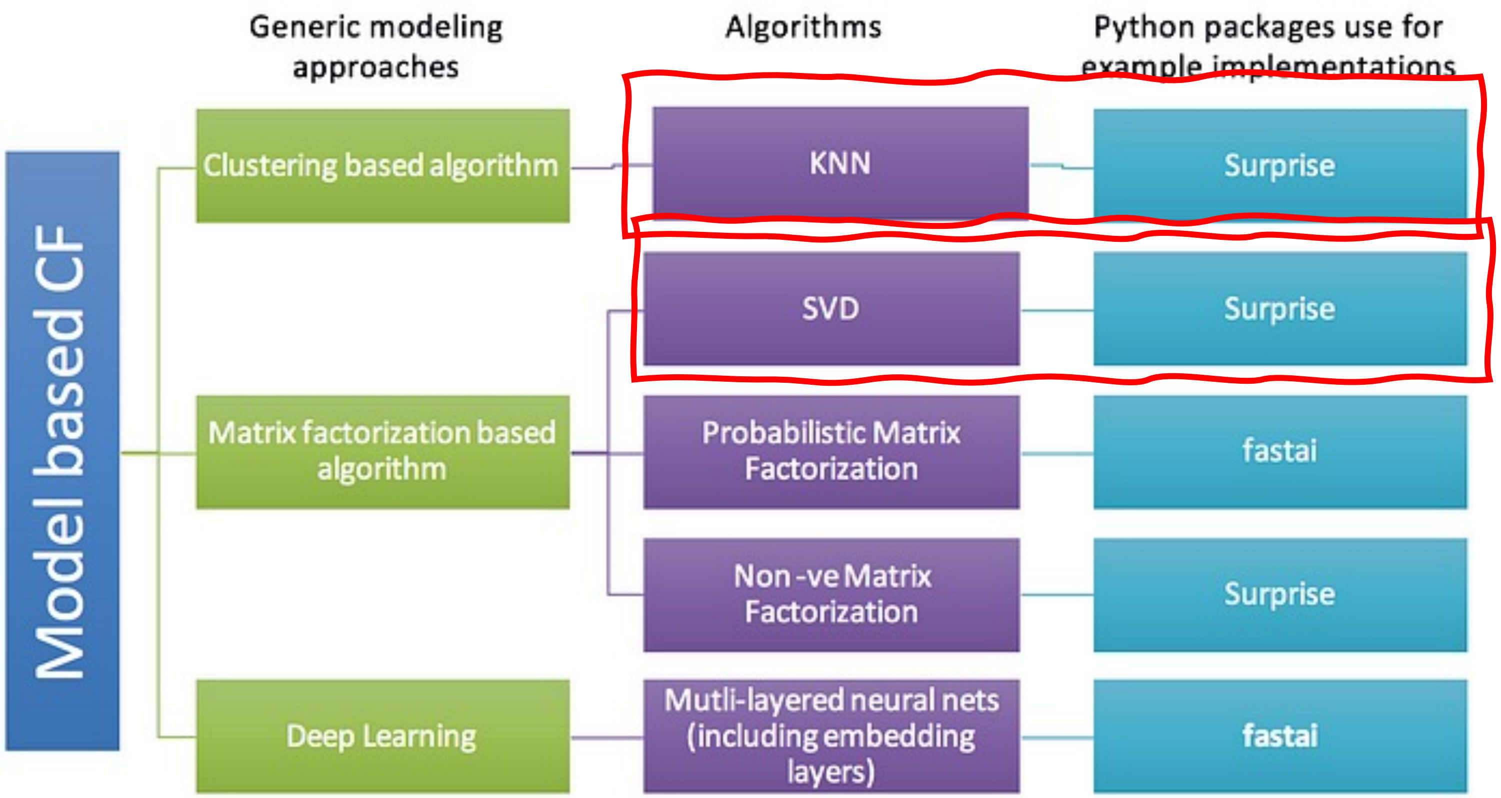


DATA SCHEME
Olist



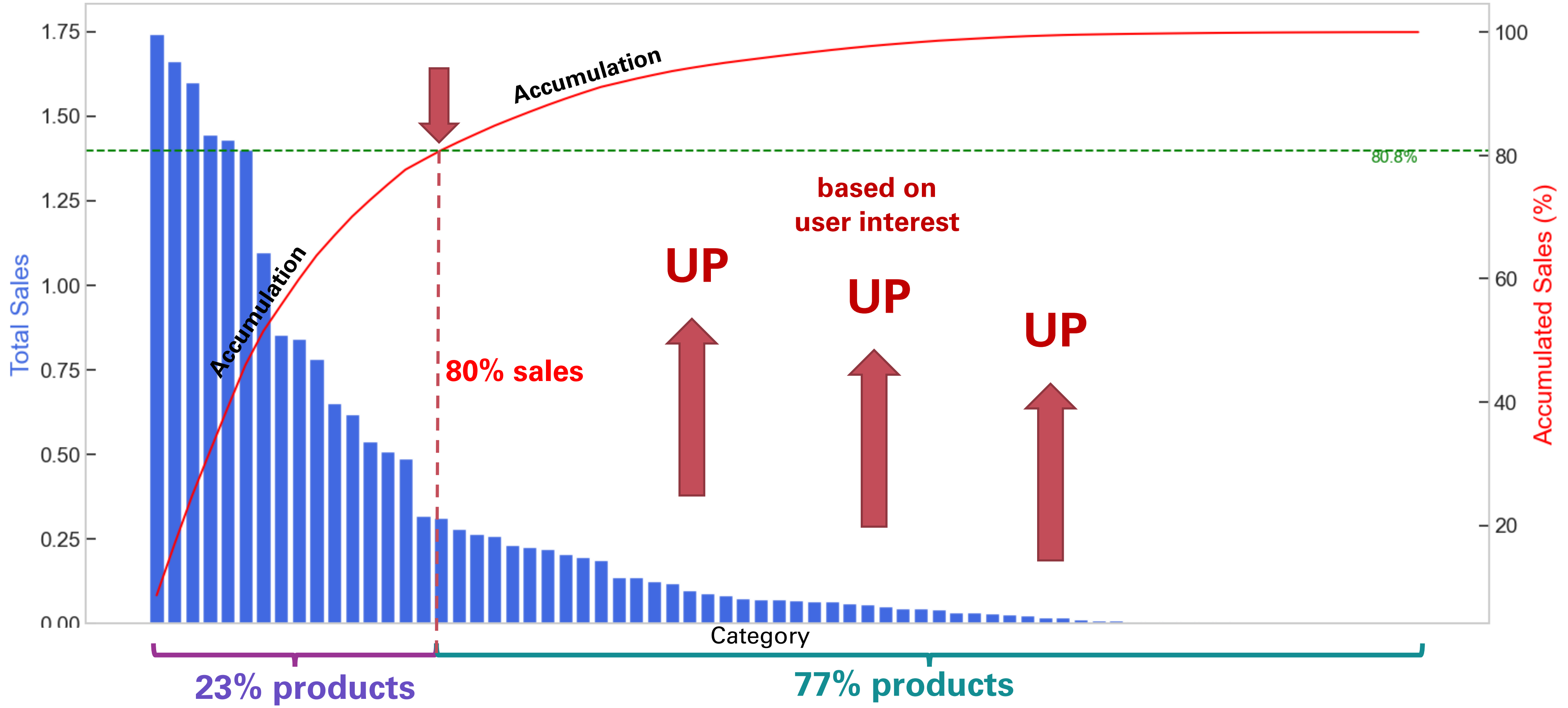
COLLABORATIVE FILTERING

using Surprise Library



TOP 17 CATEGORIES FROM 72 CATEGORIES

EDA



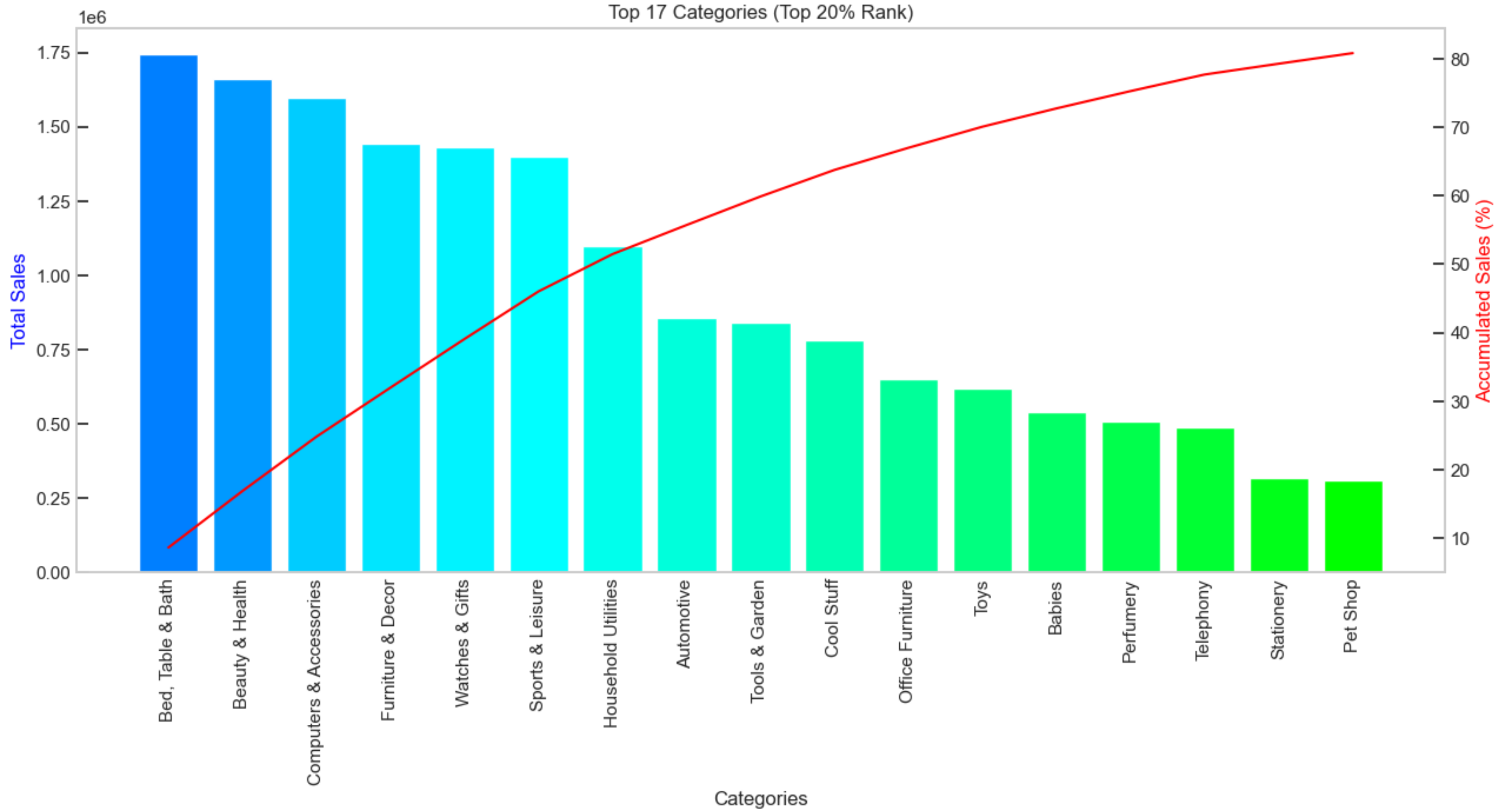
23% products contribute **80% sales** which align with the pareto & long-tail

Hence there're still a lot of opportunities to recommend other products

https://www.nngroup.com/articles/long-tail/

TOP 17 CATEGORIES (TOP 20% RANK)

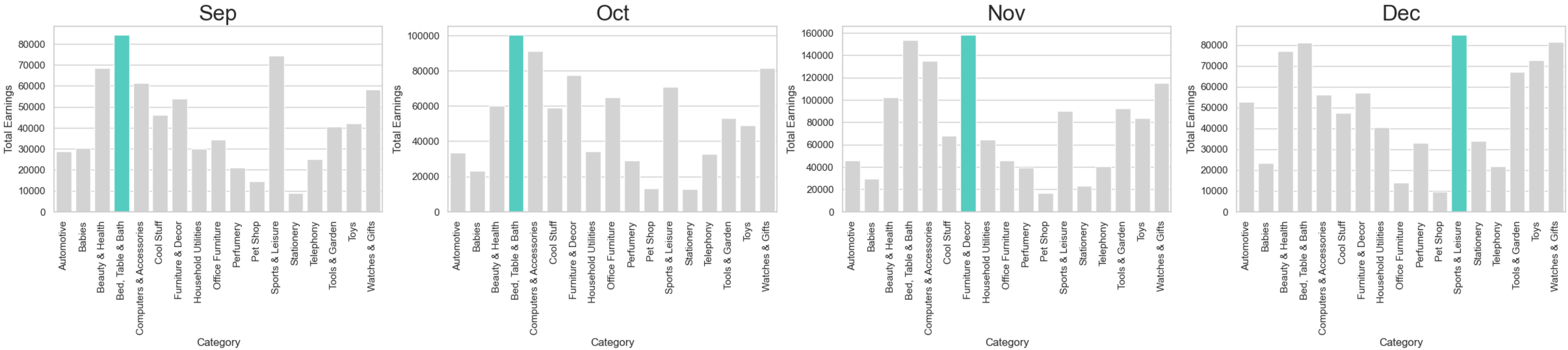
EDA



Top 3 are **Bed, Table & Bath, Beauty & Health,** and **Computer & Accessories**
Next top 3 are **Furniture & Décor, Watches & Gifts,** and **Sports & leisure**

MONTHLY BEST SELLING CATEGORIES

EDA

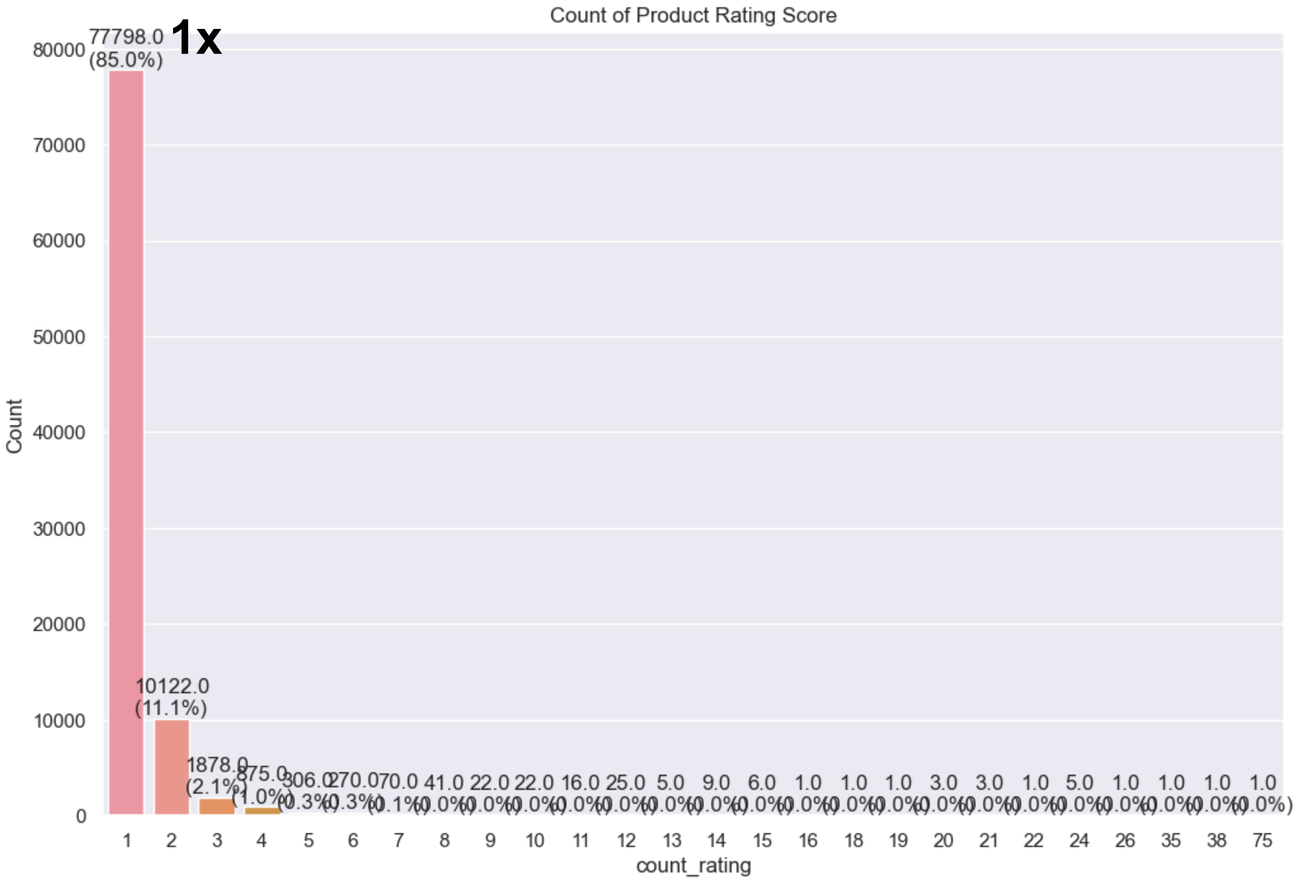


3x consecutively on **Computer & Accessories**
4x times on **Bed Table & Bath**
3x times on **Beauty & Health**

Hence monthly top category product could use for first-time customer since they still don't have their purchase data.

COUNT OF PRODUCT RATING SCORE

EDA



Mostly Customer **Rated** only **1 Times** with almost **85%**

It seems likely rate by First-Time Customer, and could be Repeat Customer don’t want to give rating

RATING SCORE OF PRODUCT

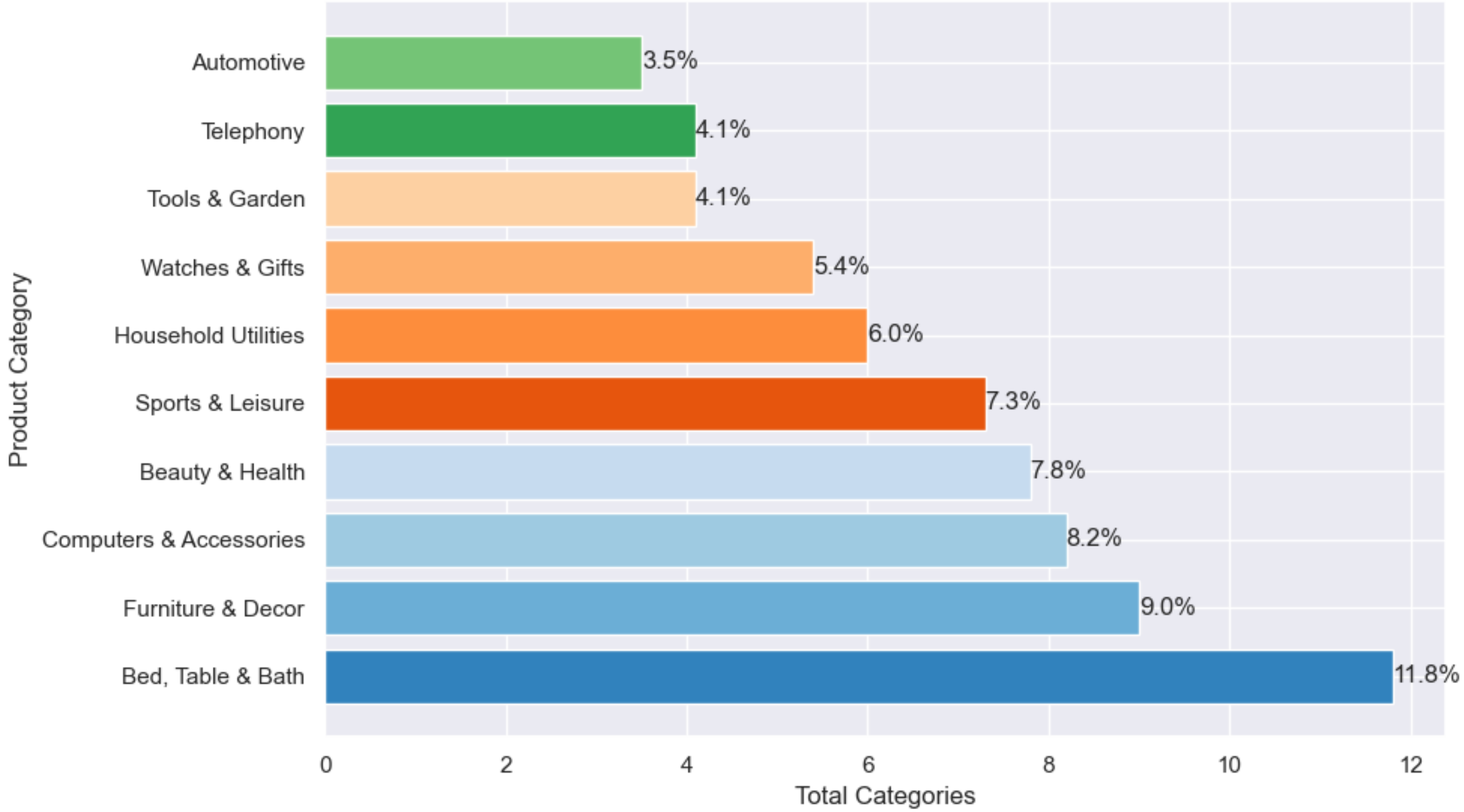
EDA

5 Rating Score with mostly **57%**

except *for* ***1 star score*** *had high score, furthermore I would inspect on the further step.*

TOTAL PRODUCT CATEGORY WITH 1 SCORE

EDA

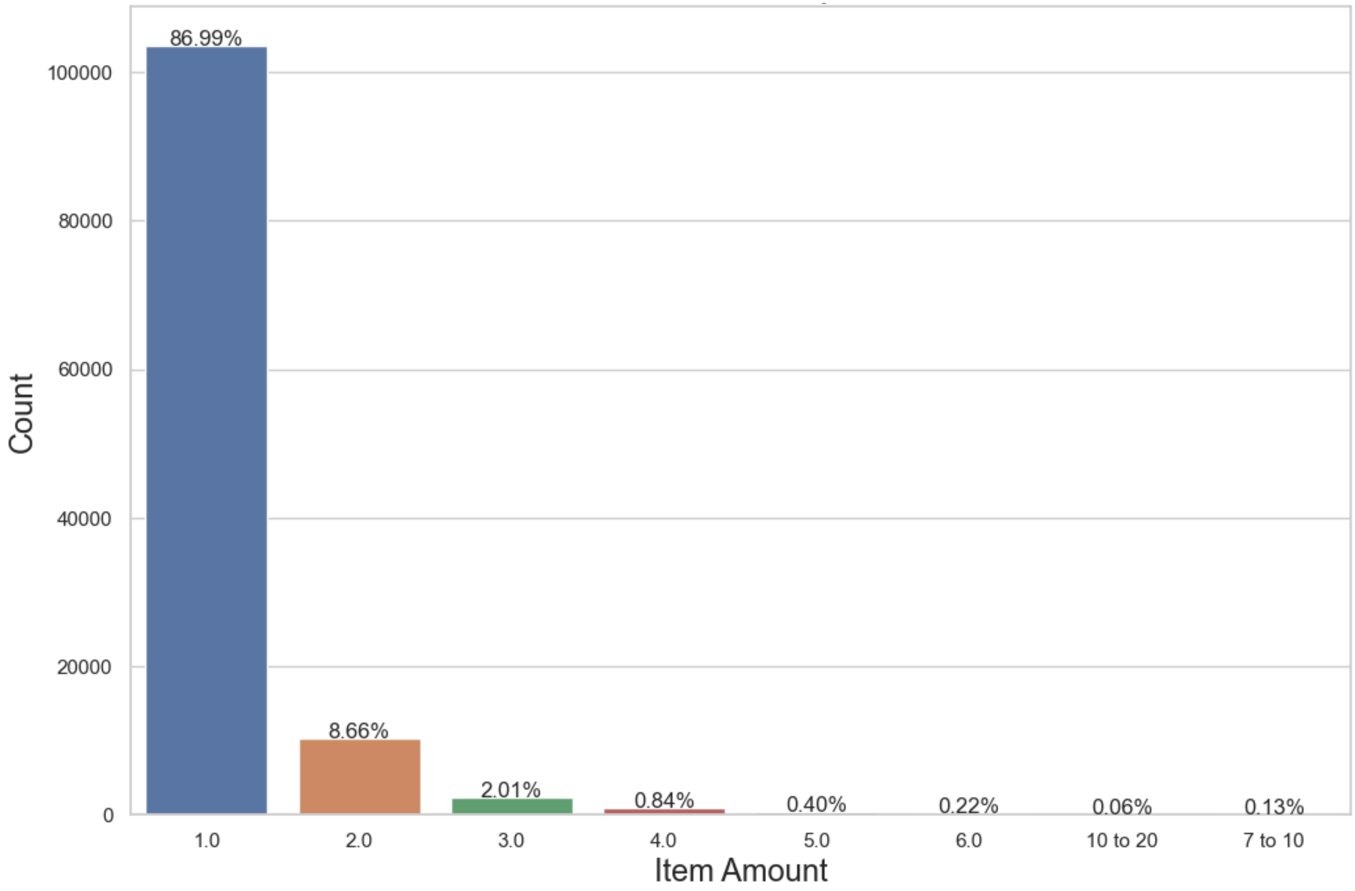


Unfortunately, some of **Top Category Product** has **Rated** with **1 Score**

Hence there's need service improvement

DISTRIBUTION OF PRODUCT PER ORDER

EDA

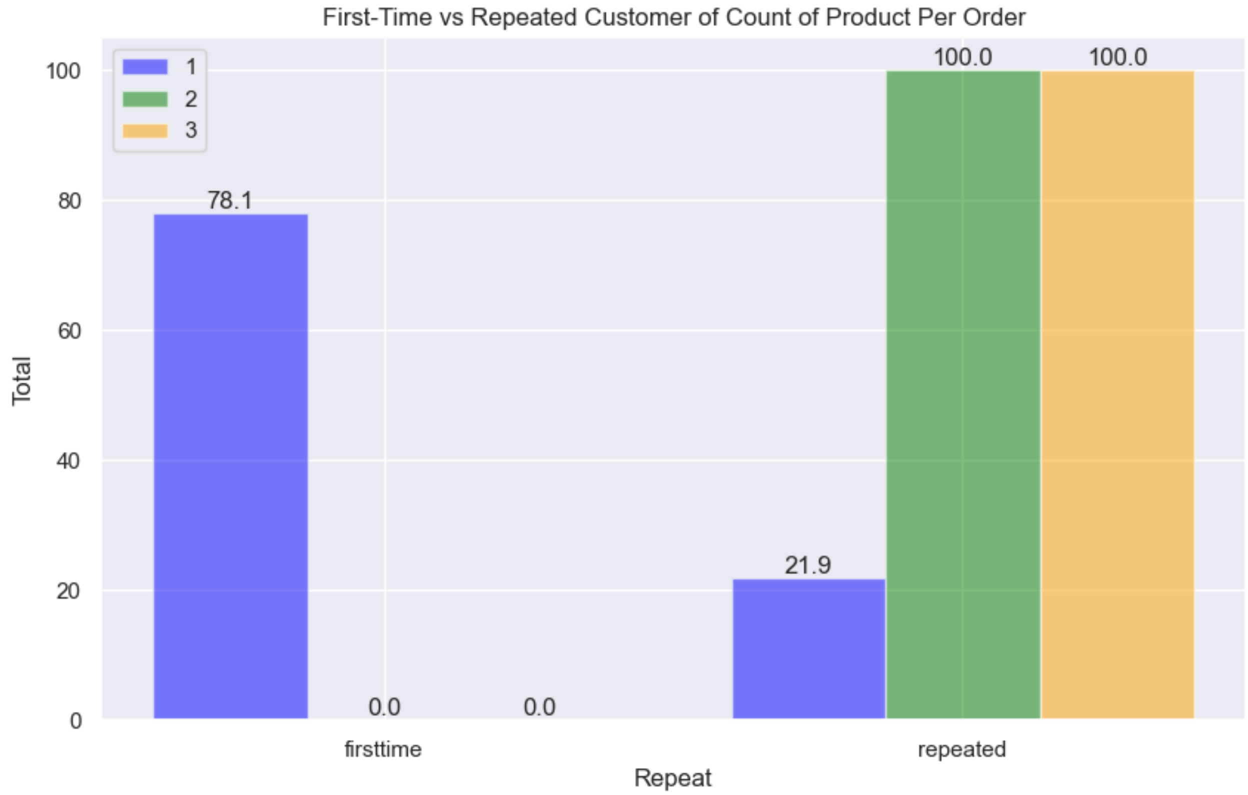


Almost **87% order** is only bought **1 product**

Hence it align with the given product ratings count that still 1x times

FIRST-TIME VS REPEATER:
HOW MANY PRODUCT PER ORDER?

EDA



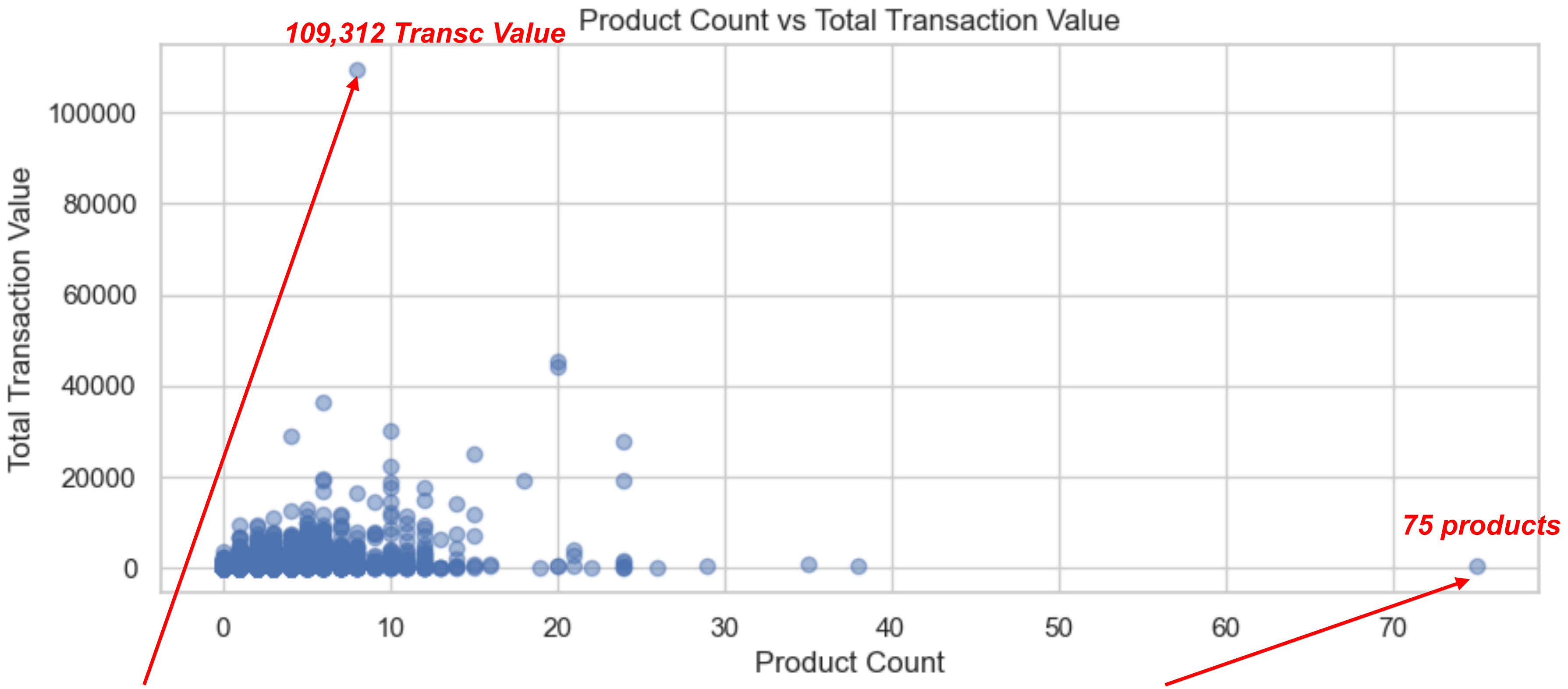
78% First-Time Customer only **buy 1 product**

vs

100% Repeat Customer buy **2-3 products**

HIGHEST PURCHASE AMOUNT OF PRODUCT & HIGHEST TRANSACTION VALUE OF PRODUCT

EDA

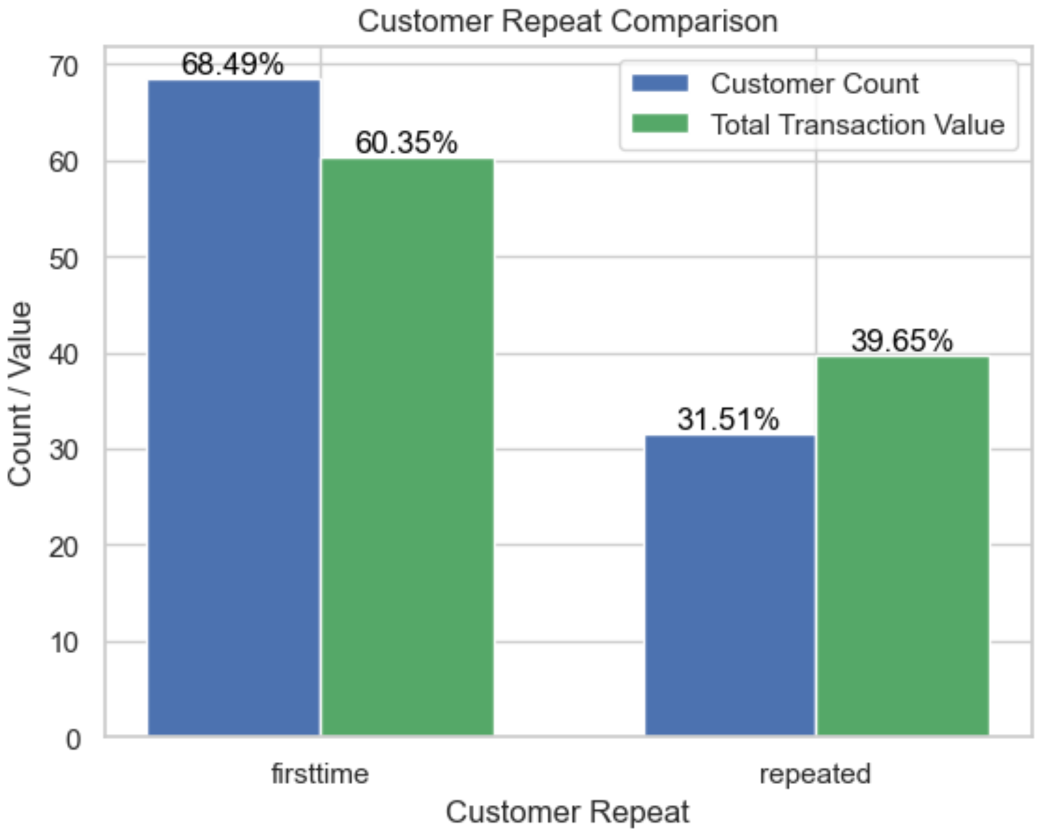


Surprisely, the customer bought 8x **landline telephone** using credit card
Need further analysis to validate this.

Most of them included to 17 Top Products such as furniture & décor; bed, table & bath; and household utilities using voucher

TRANSACTION VALUE: REPEAT VS FIRST-TIME CUSTOMER

EDA

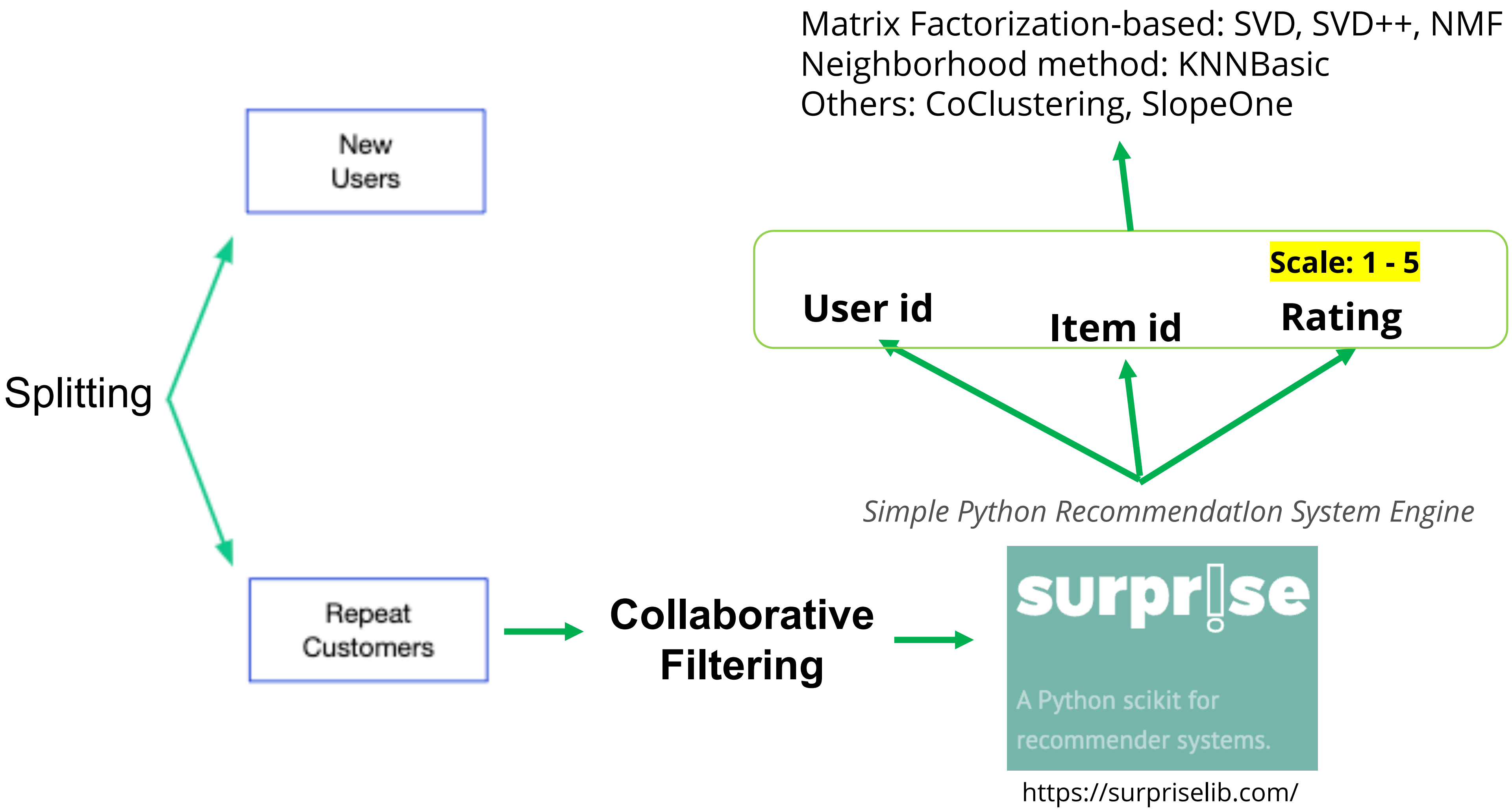


Repeat customers made repeat purchases of the product
by seeing total transaction value is higher than their amount.

Hence there’re still lot of opportunities to gain more selling on repeat customer and first-time customer

PREPARING DATAFRAME

Recommendation System



MODELLING RESULT

Recommendation System

7 Models on Surprise Library

	normalpred	SVD	SVD++	NMF	KNNBasic	CoClustering	SlopeOne
RMSE train	1.953044	1.523386	1.519055	1.527452	0.302666	0.409888	0.035610
RMSE test	1.939800	1.513225	1.513225	1.513092	1.470767	1.414831	1.370222

Selected Model

Base Model (SVD++)

```
RMSE train      1.519055
RMSE test       1.513225
RMSE diff       0.005830
Name: SVD++, dtype: float64
```

SVD++ as the **best model** with **lowest RMSE** & **lowest difference** between RMSE trainset and testset

Tuning

Tuned Model

```
Train               Test
RMSE: 0.3210        RMSE: 0.3248
MAE:  0.2593        MAE:  0.2616
```

After **tuning** the model using **GridSearchCV**, the model has **improved** almost **5x** significantly decreased of RMSE for both trainset & testset

EVALUATING RECOMMENDER

Recommendation System

Predicted Actual Error

Joni Lexung Car

Best Prediction

	uid	iid	rui	est	details	Iu	Ui	err
3188	4f82360423c19895d2a8183e6d8ec701	bf66f5d110af02b50af37038afe90bd9	1.0	1.0	{'was_impossible': False}	2	2	0.0
100	a133f658a4d462264b8d4d8cbb282393	7c6fbb3a5346dfd607386155c6f628f8	5.0	5.0	{'was_impossible': False}	2	3	0.0
2123	b0d60f871dec79cae101d9e74c816407	65223c26538a2226610efc437e488b77	1.0	1.0	{'was_impossible': False}	2	2	0.0

Top: Joni bought Lexung Car with **Actual Rating 1** Score otherwise the **Predicted Rating 1** Score too, Hence the **difference/error is 0**. Meanwhile this Car was rated by 2 person.

Predicted Actual Error

Kamila Cupid Doll

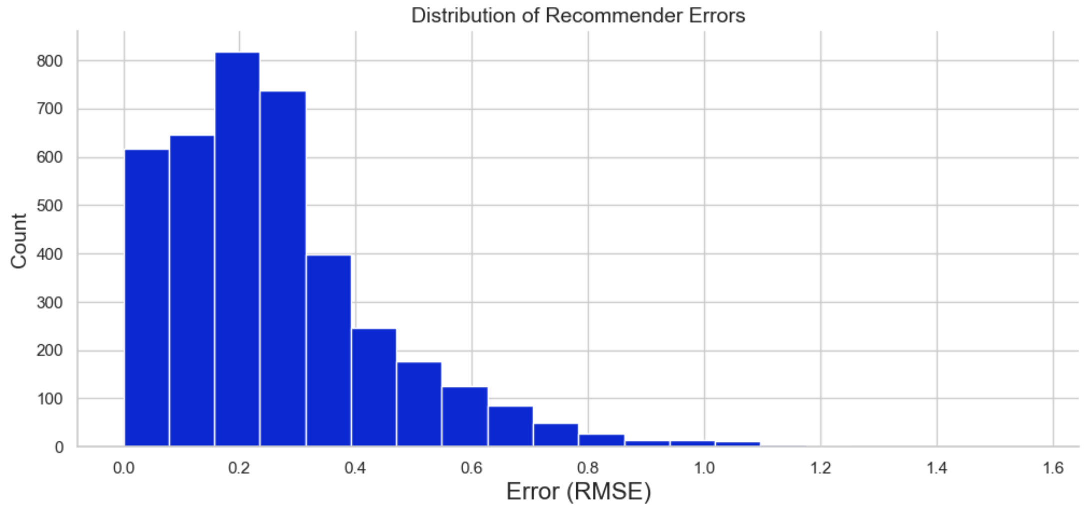
Worst Prediction

	uid	iid	rui	est	details	Iu	Ui	err
2568	c49760f3652f57abae1908ae0a452350	87d780fa7d2cf3710aa02dc4ca8db985	1.0	2.076481	{'was_impossible': False}	1	11	1.076481
618	50ad7151ad494370e8dc57a57351d17e	f71973c922ccaab05514a36a8bc741b8	1.0	2.081859	{'was_impossible': False}	1	9	1.081859
260	4f88e6285f805c9a823ba23291a65ca3	f77dd338d9f75229a09cbb9a18fd0c9a	1.0	2.091208	{'was_impossible': False}	2	8	1.091208

Bottom: Kamila bought Cupid Doll with **Actual Rating 1** Score otherwise the **Predicted Rating 2.1** Score, Hence the **difference/error is 1.1**. Meanwhile this Doll was rated by 11 person.

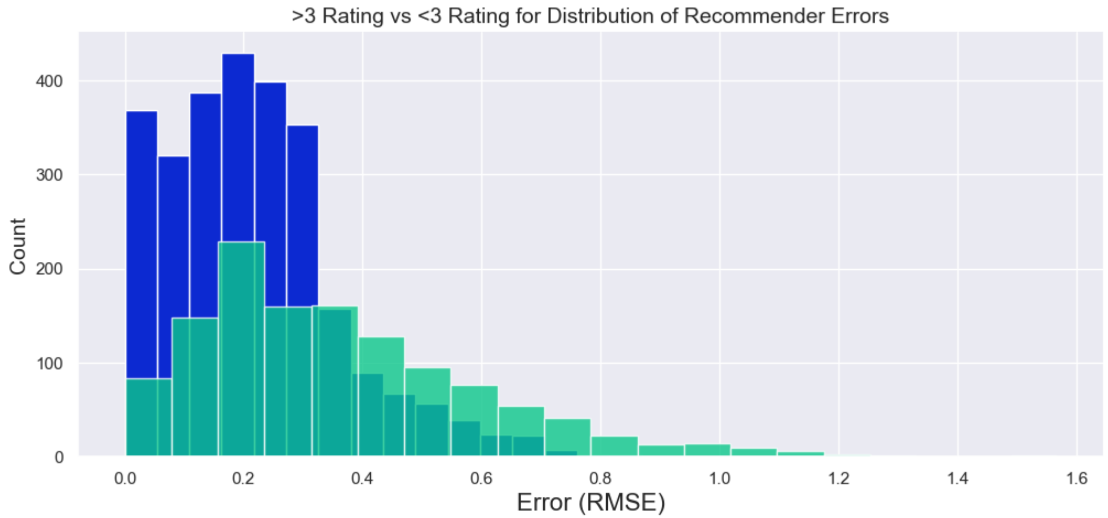
EVALUATING RECOMMENDER

Recommendation System



The model has the **good performance** so far for **all rating score**

For all rating, the model has RMSE with skewed right distribution



The model has the **good performance** so far for **above and below 3 rating score**

For above 3 rating score, the model has RMSE with skewed right distribution, meanwhile for below 3 rating score the model has same distribution but more long tail and slightly lower head

SIMPLE RECOMMENDER: FIRST-TIME CUSTOMER

Recommendation System

First-Timer Customers → Cold-start Problem

Top 5 Popular Products Recommendation

```
Top 5 items:
                         product_id product_category_name
0  aca2eb7d00ea1a7b8ebd4e68314663af     Furniture & Decor
1  99a4788cb24856965c36a24e339b6058     Bed, Table & Bath
2  422879e10f46682990de24d770e7f83d        Tools & Garden
3  389d119b48cf3043d311335e499d9c6b        Tools & Garden
4  368c6c730842d78016ad823897a372db        Tools & Garden
```

Top 5 Products in **State** Recommendation

```
Customerid: 11e379501f73ec65c77def13ec8ae678
Top 5 trending products around you (MG, belo horizonte):
                         product_id    product_category_name
0  d1c427060a0f73f6b889a5c7c61f2ac4  Computers & Accessories
1  389d119b48cf3043d311335e499d9c6b           Tools & Garden
2  99a4788cb24856965c36a24e339b6058        Bed, Table & Bath
3  422879e10f46682990de24d770e7f83d           Tools & Garden
4  3dd2a17168ec895c781a9191c1e95ad7  Computers & Accessories
```

CONCLUSION & RECOMMENDATIONS

Conclusion:

The analysis reveals three key problems for the Olist. Firstly, a significant **80% of transaction value** is generated from **only 17 categories of products (20%),** indicating there's still **potential products** in the remaining **34 categories**. Secondly, there is a **urgent issue** with **bad ratings** in the **top 17 categories**, which may lead to **customer dissatisfaction** and impact **customer retention**. Thirdly, there're still **78% first-time customer** that **bought 1 product** instead of repeat customer that bought **2-3 products**

Recommendations:

1. Implement a recommendation system that leverages **collaborative filtering**, specifically the SVD++ algorithm. This approach will provide **personalized product recommendations** based on user-to-product data, **improving** the overall **customer experience** and driving sales.
2. Try to **solve the issue of bad ratings** by identifying the underlying causes. Causes might be from product quality, delivery, customer support, or other factors affecting customer satisfaction. Resolving these issues would help **retain loyal customers**.
3. Since there're still a lot first-time customer than repeat customer**, overcome** the **cold-start problem** for **first-time customers** with two strategies. Firstly, implement a **Top 10 Popular Products Recommendation** to guide customers towards popular products. Secondly, implement a **Top 10 Product in State Recommendation** to offer products that are **specifically popular** in the **customer's state**. These approaches would increase the **likelihood of repeat purchases** and contributing data collection from the customer.

REFERENCES

1. https://surpriselib.com/
2. https://www.moyak.com/papers/collaborative-filtering.html
3. https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada
4. https://developers.google.com/machine-learning/recommendation/collaborative/basics
5. https://towardsdatascience.com/various-implementations-of-collaborative-filtering-100385c6dfe0
6. https://towardsdatascience.com/recsys-series-part-4-the-7-variants-of-matrix-factorization-for-collaborative-filtering-368754e4fab5
7. https://towardsdatascience.com/collaborative-filtering-and-embeddings-part-1-63b00b9739ce
8. https://towardsdatascience.com/various-implementations-of-collaborative-filtering-100385c6dfe0
9. https://www.ethanrosenthal.com/2015/11/02/intro-to-collaborative-filtering/
10. https://www.stat.ipb.ac.id/main/wp-content/uploads/2021/01/Introduction-to-RecSys_BERT4Rec-IPB.pdf
11. https://medium.datadriveninvestor.com/how-to-build-a-recommendation-system-for-purchase-data-step-by-step-d6d7a78800b6
12. https://towardsdatascience.com/building-and-testing-recommender-systems-with-surprise
13. https://towardsdatascience.com/a-simple-approach-to-building-a-recommendation-system-d0f4de1a1f50e-step-by-step-d4ba702ef80b
14. https://celikkam.medium.com/surprise-recommender-library-few-lines-for-everything-d2de3ceac3d4
15. https://towardsdatascience.com/recommendation-system-series-part-1-an-executive-guide-to-building-recommendation-system-608f83e2630a

THANKS!

Linkedin: linkedin.com/in/satriabagaskara/
Github: github.com/dataaga