PERSONALIZED E-COMMERCE RECOMMENDATION SYSTEM

MSA 8010 Data Programming

December 11, 2024

TEAM: 5-Star Recommenders
Akshara Ravishankar
Komal Naidu
Mayadir Sandoval
Nivethitha Pandi
Sandra Soy Kipsang

Georgia State University | J. MACK ROBINSON COLLEGE OF BUSINESS

Robinson

5 Star Recommenders

Akshara Ravishankar

Mayadir Sandoval

Sandra Soy Kipsang

Komal Naidu

Nivethitha Pandi

FRAMEWORK/CONTENT

1. Business problem
2. Proposed Solution Overview
3. Exploratory Data Analysis (EDA)
4. Data Preprocessing
5. ML Models
6. Evaluation and choosing best model
7. Incorporation into business

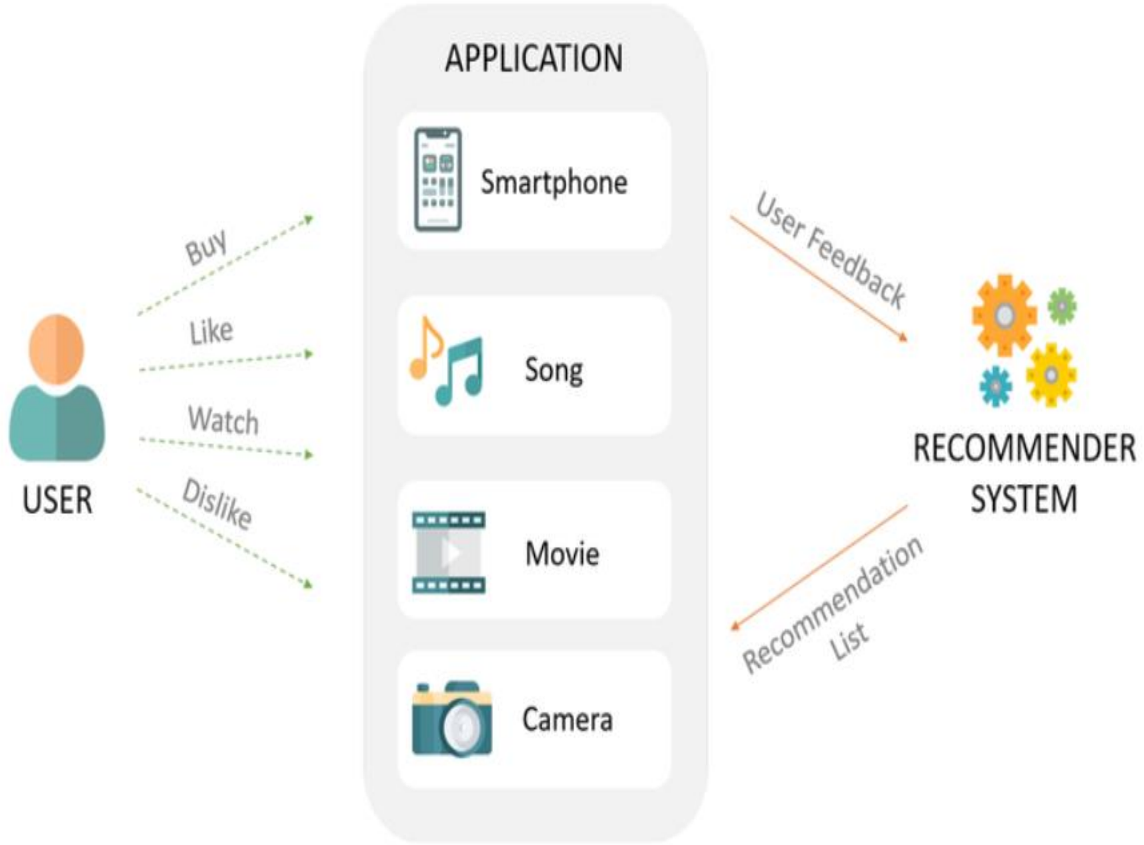
BUSINESS PROBLEM:

Are you a confused customer overwhelmed by too many choices?

- E-commerce platforms like “Olist” face challenges like **high customer churn** and **low conversion rates,** impacting continuous revenue.
- Customers churn because they don’t find content or products they like.

PROPOSED SOLUTION

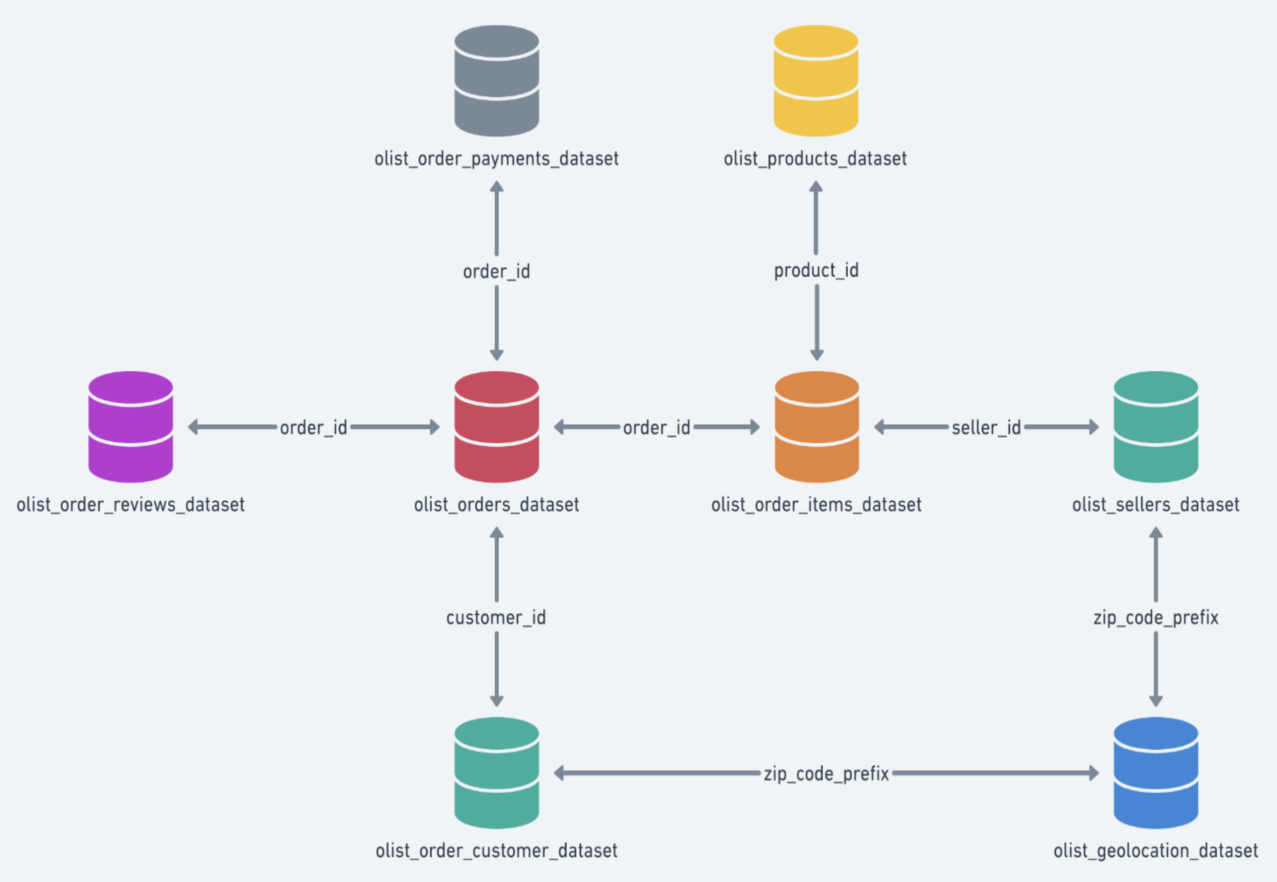
PERSONALISED PRODUCT RECOMMENDATIONS



Why recommendation system is necessary?

- A ranked list of recommended products tailored to each user, enhances the user experience and encourages repeat visits
- Recommendations account for **10-30% of e-commerce revenue** by driving upselling and cross-selling.
- Promotes lesser-known products that might match specific user preferences

Dataset Overview



Olist is a Brazilian e-commerce platform that functions as a marketplace, connecting small and medium-sized businesses with customers by allowing them to list their products on various online marketplaces.

Features: 39 Columns

Number of data points: 117329

Target Features: User id(object), product id(object), review id(object).

https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce?select=olist_order_items_dataset.csv

Brazilian E-Commerce Public Dataset by Olist

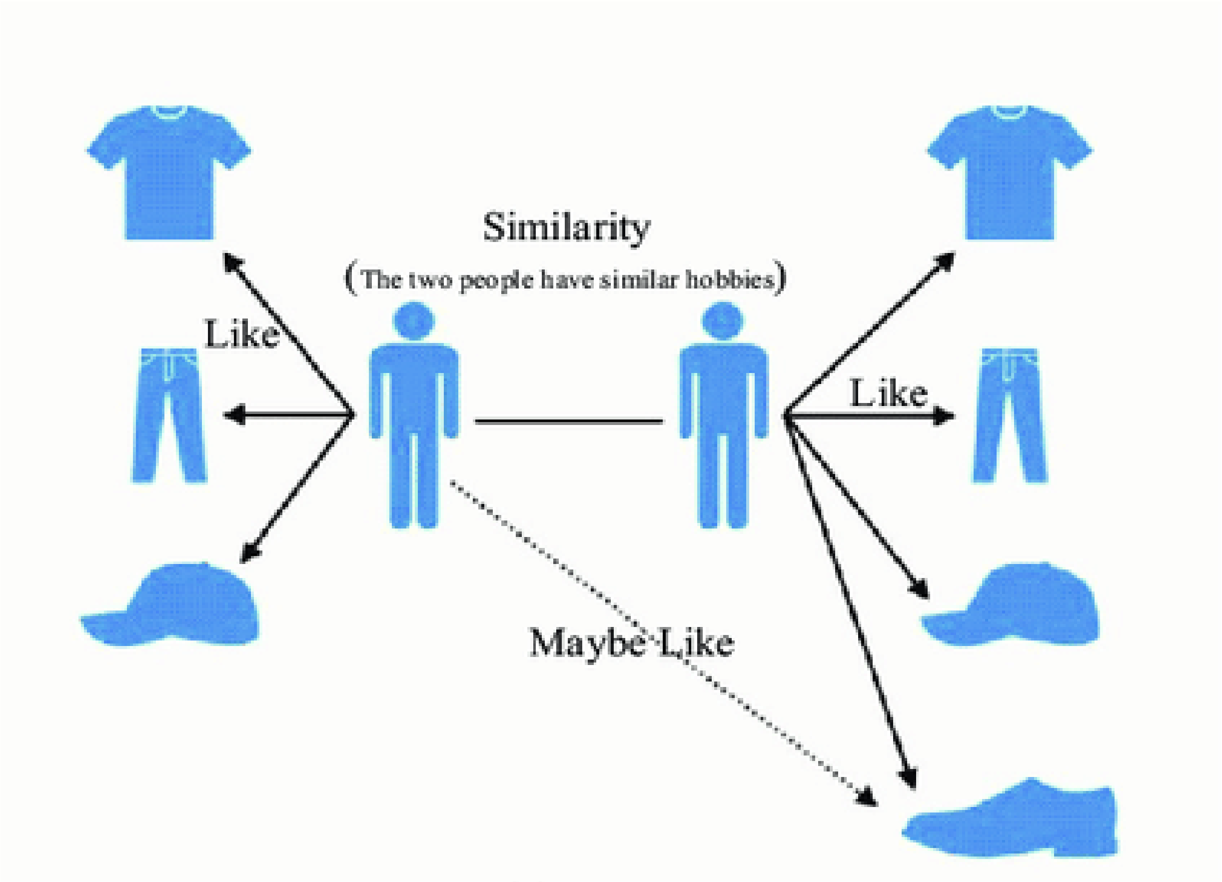
COLLABORATIVE FILTERING

User Based Collaborative Filtering

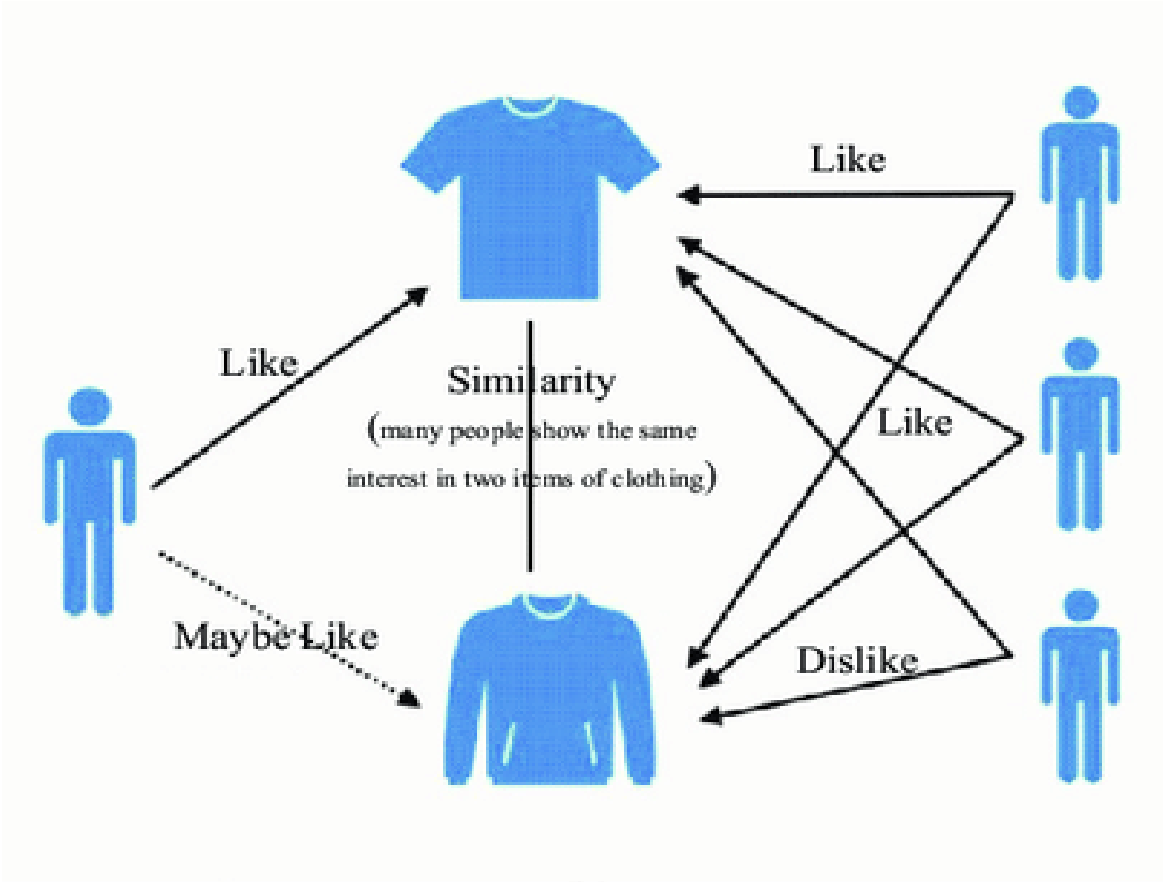
- Based on user’s neighborhood

Item Based Collaborative Filtering

- Based on item’s similarity



(a)



(b)

CF MODELING CYCLE

EDA

Data Pre-processing

ML Model Testing

Hyperparameter tuning

Model Evaluation

Final Model Selection

Provide product Recommendations

Summary Statistics

```python
from skimpy import skim
skim(df)
```

skimpy summary

Data Summary

dataframe	Values
Number of rows	117329
Number of columns	39

Data Types

Column Type	Count
string	23
float64	10
int32	6

number

column_name	NA	NA %	mean	sd	p0	p25	p50	p75	p100	hist
payment_sequential	0	0	1.1	0.73	1	1	1	1	29	
payment_installments	0	0	2.9	2.8	0	1	2	4	24	
payment_value	0	0	170	270	0	61	110	190	14000	
customer_zip_code_prefix	0	0	35000	30000	1000	11000	24000	59000	100000	
review_score	0	0	4	1.4	1	4	5	5	5	
order_item_id	0	0	1.2	0.68	1	1	1	1	21	
price	0	0	120	180	0.85	40	75	130	6700	
freight_value	0	0	20	16	0	13	16	21	410	
product_name_lenght	1695	1.44	49	10	5	42	52	57	76	
product_description_lenght	1695	1.44	790	650	4	350	600	980	4000	
product_photos_qty	1695	1.44	2.2	1.7	1	1	1	3	20	
product_weight_g	20	0.02	2100	3800	0	300	700	1800	40000	
product_length_cm	20	0.02	30	16	7	18	25	38	100	
product_height_cm	20	0.02	17	13	2	8	13	20	100	
product_width_cm	20	0.02	23	12	6	15	20	30	120	
seller_zip_code_prefix	0	0	24000	28000	1000	6400	14000	28000	100000	

string

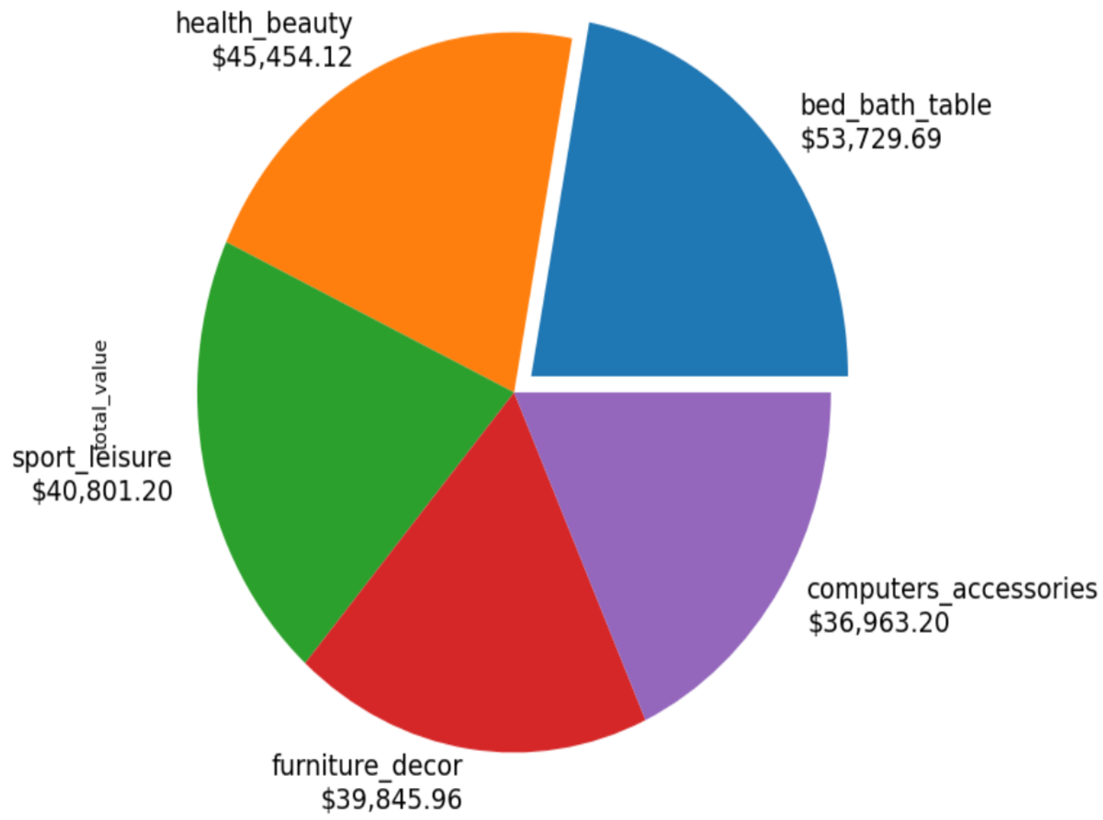
column_name	NA	NA %	words per row	total words
order_id	0	0	1	117329
customer_id	0	0	1	117329
order_status	0	0	1	117329
order_purchase_timestamp	0	0	2	234658
order_approved_at	15	0.01	2	234628
order_delivered_carrier_date	1235	1.05	2	232188
order_delivered_customer_date	2471	2.11	2	229716
order_estimated_delivery_date	0	0	2	234658
payment_type	0	0	1	117329
customer_unique_id	0	0	1	117329
customer_city	0	0	1.8	205733
customer_state	0	0	1	117329
review_id	0	0	1	117329
review_comment_title	103437	88.16	0.25	28929
review_comment_message	67650	57.66	5.1	603059
review_creation_date	0	0	2	234658
review_answer_timestamp	0	0	2	234658
product_id	0	0	1	117329
seller_id	0	0	1	117329
shipping_limit_date	0	0	2	234658
product_category	1695	1.44	0.99	115680
seller_city	0	0	1.7	201631
seller_state	0	0	1	117329

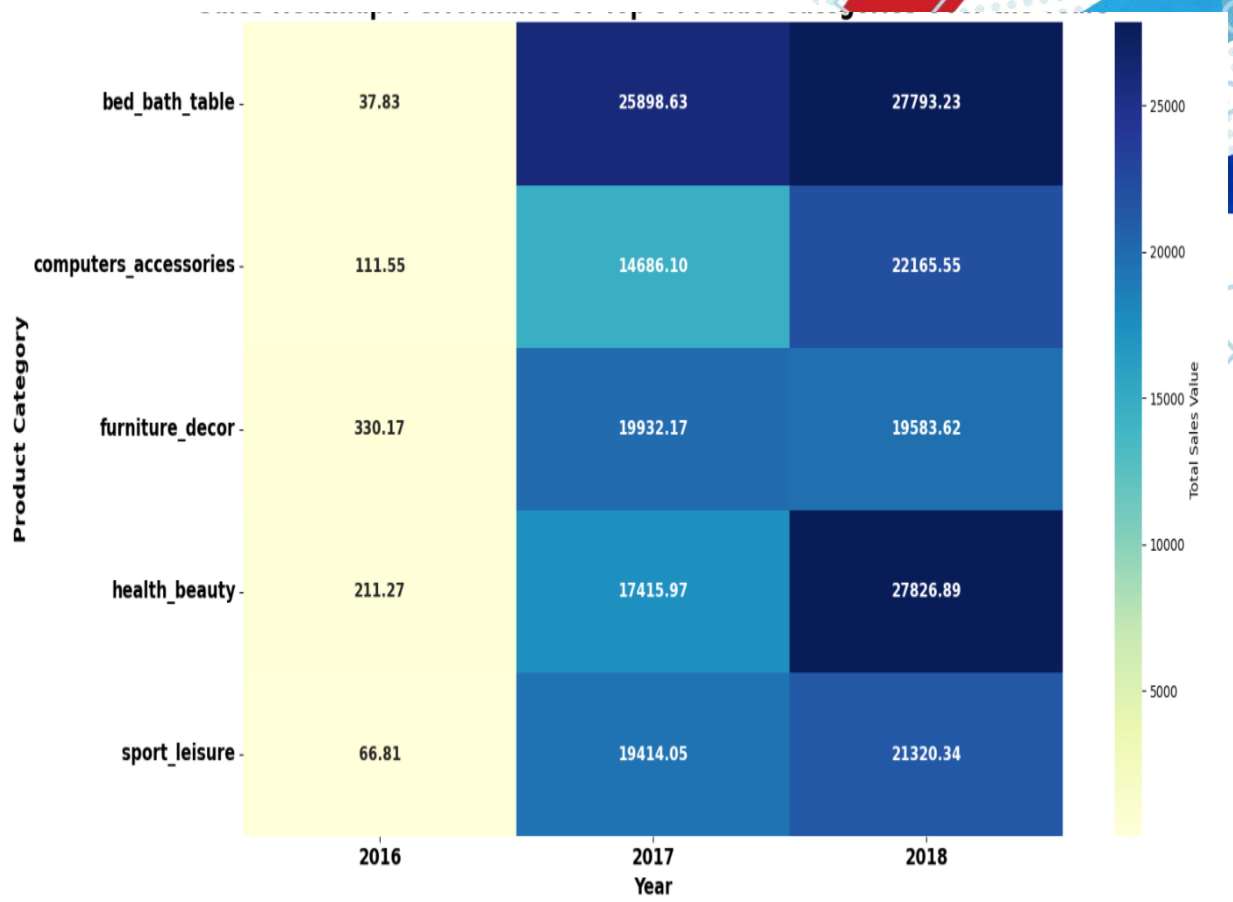
End

EDA Highlights

	Missing Count	Missing Percentage
review_comment_title	103437	88.159790
review_comment_message	67650	57.658379
order_delivered_customer_date	2471	2.106044
product_category	1695	1.444656
product_name_lenght	1695	1.444656
product_description_lenght	1695	1.444656
product_photos_qty	1695	1.444656
order_delivered_carrier_date	1235	1.052596
product_length_cm	20	0.017046
product_weight_g	20	0.017046
product_height_cm	20	0.017046
product_width_cm	20	0.017046
order_approved_at	15	0.012785

EDA Highlights





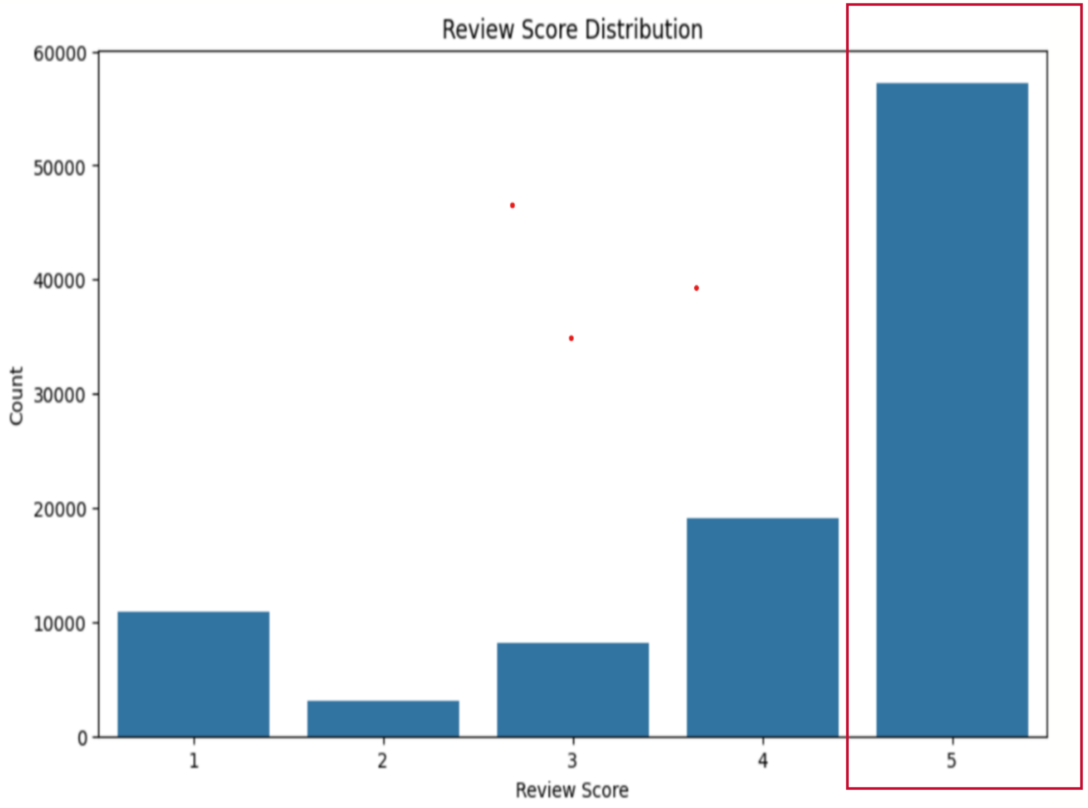
Top 5 Product Category by Total Sales Value

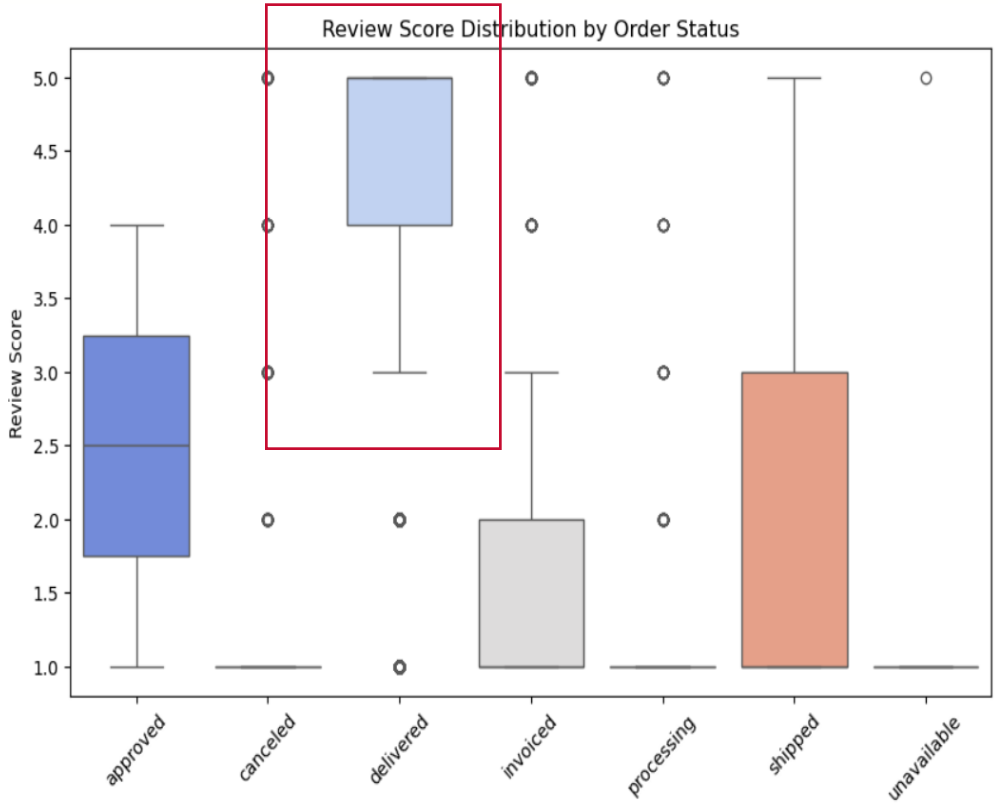
Sales Heatmap: Performance of Top 5 Product Categories Over the Years

EDA Highlights

Review Score

- The distribution is skewed
- Over 50% of customers have given a 5-star rating followed by a little below 20 % with 4-star ratings





EDA Highlights

Most Interacted Customers

- The top-most customer interactions are on the lower range
- Indicates less interaction – data sparsity

DATA PREPROCESSING

1. **Build a User – Item Interaction Matrix using explicit feedback (ratings)**

2. **Reducing Sparsity:**

We have retained only interactions of users that had **atleast 1 review**, and items that were **ordered atleast 10** times.

```
Shape of final_ratings_matrix:  (24014, 420)
given_num_of_ratings =  24212
possible_num_of_ratings =  10085880
density: 0.24%
```

product_id	162	306	397	498	613	664	731	836	854	860	...	31130	31150	31188	31192	31213	31453	31458	31502	31527	31684
customer_unique_id																					
21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
37	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
44	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

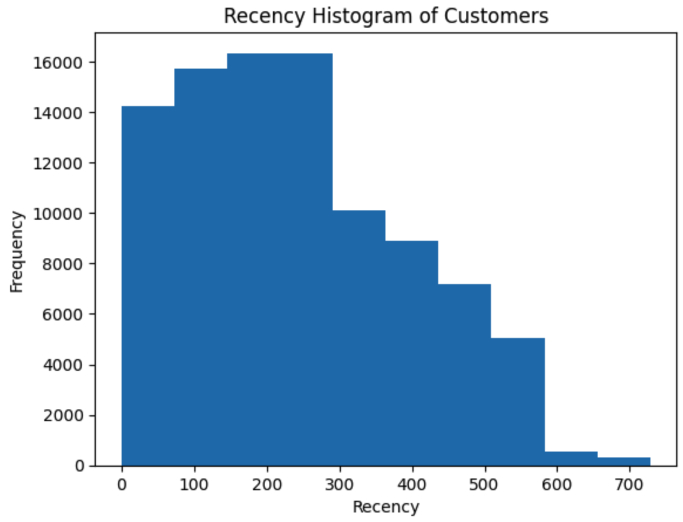
FEATURE ENGINEERING

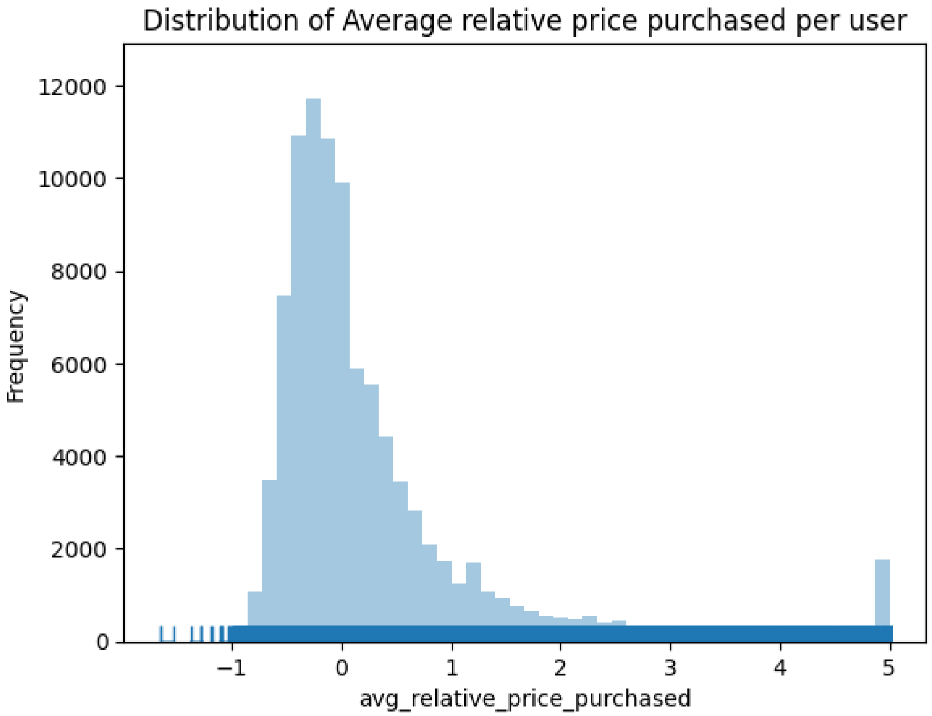
User Features:

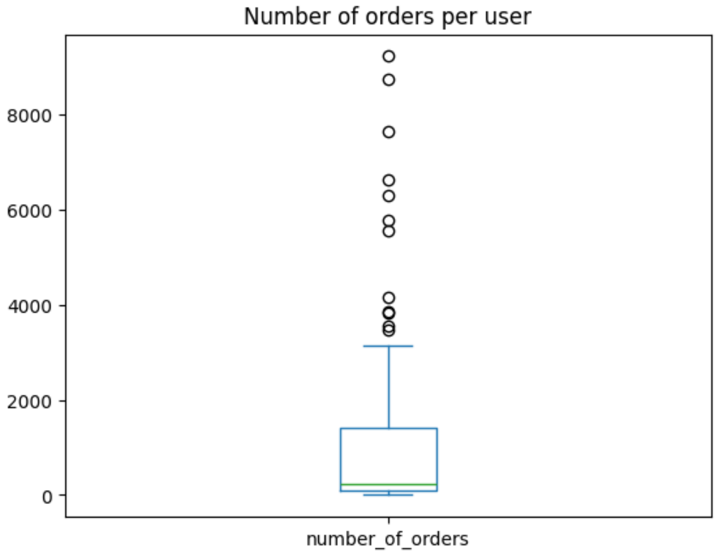
Mean rating, Recency of order, Average price bought

Product Features:

Relative price of product based on products in same category







Georgia State University | J. MACK ROBINSON COLLEGE OF BUSINESS

Robinson

ML Models Tested

MODEL	SVD	SVD++	ALS
PROPERTIES	• decomposes the user-item interaction matrix into three smaller matrices • captures latent factors that influence user-item relationships	• can considers both explicit ratings and implicit feedback • incorporates user and item biases improving accuracy	• decomposes interaction matrix into user latent factors & item latent factors. • optimizes user, item matrices alternately
REASON FOR CHOOSING	• captures Latent Features • scalable for Explicit Feedback	• more accurate recommendations • captures complex relationships	• robustness with sparsity • regularization for Better Generalization

Initial Iterations

SVD (Singular Value Decomposition) & SVD++

- a matrix factorization technique, which reduces the number of latent features
- decomposes into 3 other matrices: R=U·S·VT

U: User latent feature matrix.

S: Diagonal matrix of singular values (importance weights of latent factors).

VT: Item latent feature matrix.

SVD

1. Data Normalization
2. Sparse Matrix Representation
3. keep only the top 50 latent features
4. Split to Train-Test

RMSE: 0.129
Precision at 5: 0.294
Recall at 5: 0.300

```
RMSE_test = accuracy.rmse(predictions)
RMSE_test
✓ 0.0s

RMSE: 0.129013
```

SVD++: (extension of SVD)

1. From "Surprise" Library
2. Data Normalization
3. Sparse Matrix Representation
4. Split to Train-Test

Evaluating RMSE, MAE of algorithm SVDpp on 5 split(s).

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	1.3184	1.3190	1.3185	1.3174	1.3152	1.3177	0.0014
MAE (testset)	1.0323	1.0349	1.0300	1.0316	1.0279	1.0313	0.0023

ALS (ALTERNATING LEAST SQUARES) MODEL FOR RECOMMENDATIONS

Model Testing

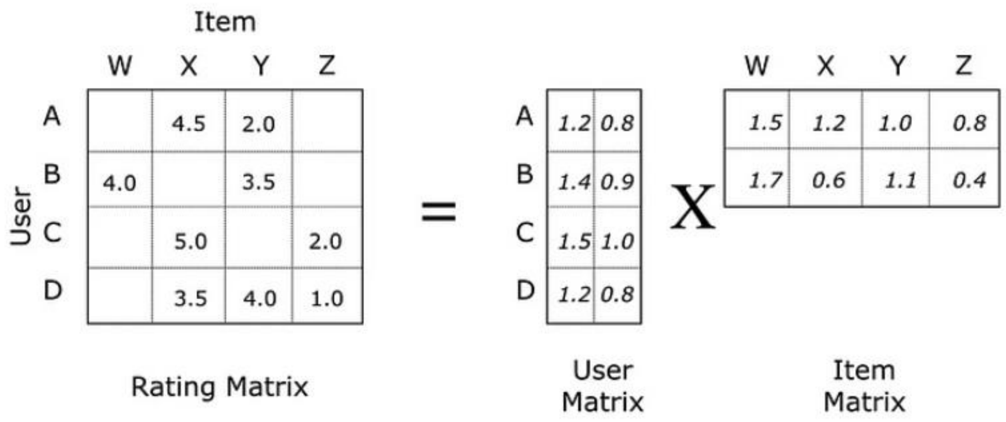
PySpark : Python API for Apache Spark for real-time, large-scale data processing in a distributed environment.

Hyperparameter tuning:

Rank: 30, MaxIter: 10 RegParam: 0.05

Evaluation:

- RMSE: 0.2314
- MAE: 0.0122
- Mean Average Precision: 0.618

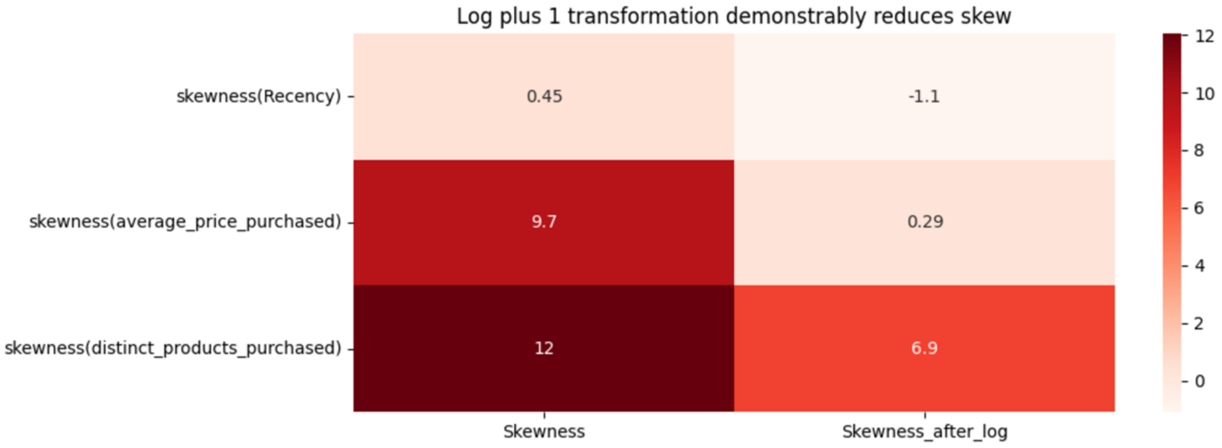


Ref: https://medium.com/@brunoborges_38708/recommender-system-using-als-in-pyspark-10329e1d1ee1

CLUSTERING CUSTOMERS BASED ON ORDER DETAILS

Pre-processing:

- Reducing skew (Log transformation)
- Standardize data (Standard scaler)
- Principal Component Analysis (PCA) to reduce dimensionality



Model Testing

- K-Means clustering model
- Evaluation: Silhouette score

$$s(\boldsymbol{o}) = \frac{b(\boldsymbol{o}) - a(\boldsymbol{o})}{\max\{a(\boldsymbol{o}), b(\boldsymbol{o})\}}$$

- *s(o)* is the silhouette coefficient of data point **o**

- *a(o)* is the *average distance* between **o** and all the other data points in same cluster

-b(o) is the *minimum average distance* from **o** to all clusters to which **o** does not belong

-Ranges between -1 to 1.

Silhouette score =0.7167

MODEL EVALUATION

Root Mean Square Error (RMSE)
Calculates the square root of the average squared differences between predicted and actual ratings

- In predicting explicit feedback (e.g., ratings), RMSE is the most reliable metrics to assess accuracy
- provides a more detailed view of errors, since it penalizes larger errors more
- Expected in the range 0.8 to 1.3 on a 5 point scale

Interpretation & Results:

Evaluation Metric	SVD	SVD++	ALS
RMSE	0.1290	1.3175	0.2314

Based on the above RMSE score, we deem the **SVD++** model to be performing better, as the RMSE falls within range.
We use the SVD++ model to generate product recommendations.

GENERATING RECOMMENDATIONS

- Recommendations are generated for all the users, suggesting products with highest ratings.

- Recommendation function accepts 'Customer ID' as input and generates top 'K' recommendations.

```
+------------------------+--------------------+
|customer_unique_id_index|     recommendations|
+------------------------+--------------------+
|                      26|[{99, 1.22042E-4}...|
|                      27|[{99, 5.904459E-9...|
|                      28|[{99, 4.9281876E-...|
|                      31|[{99, 1.2435938E-...|
|                      34|[{214, 1.7780683E...|
|                      44|[{84, 2.318453E-9...|
|                      53|[{122, 0.90258265...|
|                      65|[{363, 0.0}, {362...|
|                      76|[{36, 0.001375694...|
|                      78|[{36, 5.123226E-1...|
+------------------------+--------------------+
```

```
Top 10 product recommendations for user 1911:
[18217, 8418, 29079, 13783, 26023, 7893, 6782, 13440, 12917, 18475]
```

```
Top 10 product recommendations for user 3333:
[29547, 15647, 10545, 12917, 21925, 10019, 13783, 28729, 20967, 27230]
```

INCORPORATION INTO BUSINESS

- **OLIST** should integrate the Recommendation Model into their backend Systems, linking product catalogs and user profiles for **REAL-TIME** recommendations
- Deploy the model via APIs to ensure seamless interaction with the websites front-end
- Create REAL-TIME data pipeline to process user interactions

CAPACITY REQUIREMENTS

- Data Storage and Processing
- Computational Resources
- Real-time Performance
- Monitoring through Dashboarding

REFERENCES

[1] https://medium.com/@brunoborges_38708/recommender-system-using-als-in-pyspark-10329e1d1ee1

[2] https://towardsdatascience.com/recommendation-system-matrix-factorization-d61978660b4b

[3] https://medium.com/@eliasah/deep-dive-into-matrix-factorization-for-recommender-systems-from-basics-to-implementation-79e4f1ea1660

[4] https://tatevkarenaslanyan.medium.com/using-customer-and-product-features-in-recommender-systems-2734258873cf#:~:text=Matrix%20Factorization,and%20Item%20Factor%20matrices%2C%20respectively.

[5] https://haneulkim.medium.com/matrix-factorization-part-i-from-derivation-of-stochastic-gradient-descent-step-to-learning-17a7d8975965

Thank You

Georgia State University | J. MACK ROBINSON COLLEGE OF BUSINESS

Robinson