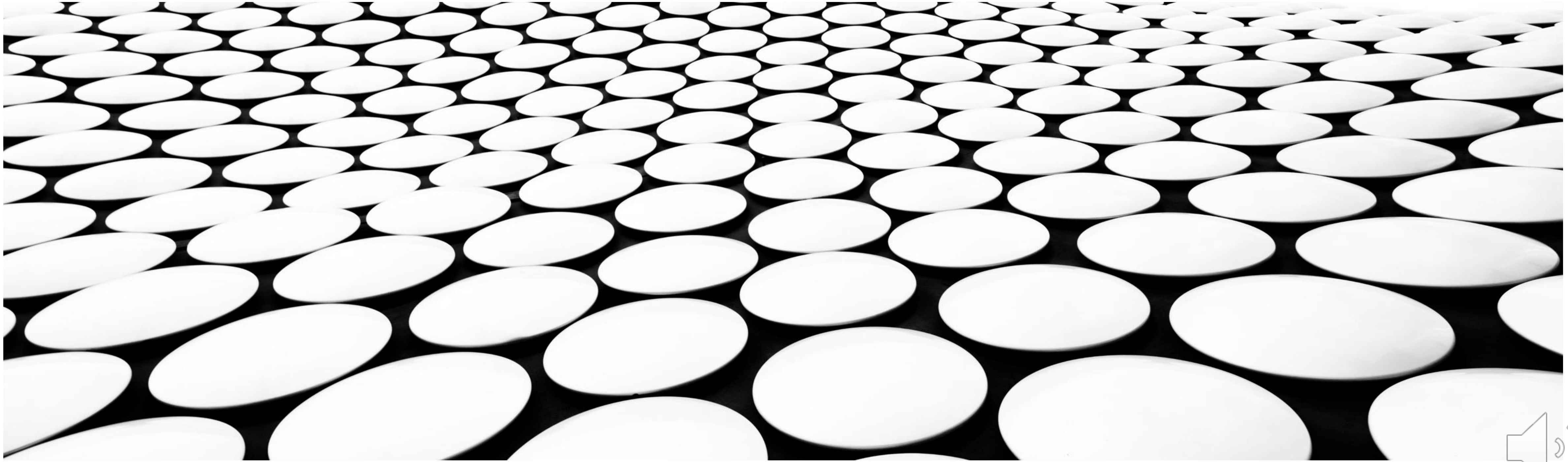
GITHUB REPOSITORY:
https://github.com/nicovy/ecommerce

# E-COMMERCE
# SEGMENTATION AND RECOMMENDER SYSTEM

NICOLAS VEAS



## CONTEXT

- Understanding the audience of an e-commerce platform is crucial for effective marketing, as it enables the use of personalized strategies.
- Developing a customer segmentation model allows us to understand the different profiles and preferences of the e-commerce customers.
- This enhances customer engagement and supports implementing a recommender system that could help improve the shopping experience by providing personalized suggestions that a specific customer will likely purchase.
- In this project, I'll use unsupervised learning models to build a customer segmentation and a recommender system.
- The data used is from kaggle from the following link:
  - https://www.kaggle.com/datasets/shrishtimanja/ecommerce-dataset-for-data-analysis

## OBJECTIVE

- \- To segment customers into distinct groups
- \- To build a recommender system that recommend products to customers

# DATA INFORMATION

55000 Observations
13 Variables
4 Numerical Variables
8 Categorical Variables
Discount name have missing values
No duplicated Values

Contains the following Variables

- CID (Customer ID): A unique identifier for each customer.
- TID (Transaction ID): A unique identifier for each transaction.
- Gender: The gender of the customer, categorized as Male or Female.
- Age Group: Age group of the customer, divided into several ranges.
- Purchase Date: The timestamp of when the transaction took place.
- Product Category: The category of the product purchased, such as Electronics, Apparel, etc.
- Discount Availed: Indicates whether the customer availed any discount (Yes/No).
- Discount Name: Name of the discount applied (e.g., FESTIVE50).
- Discount Amount (INR): The amount of discount availed by the customer.
- Gross Amount: The total amount before applying any discount.
- Net Amount: The final amount after applying the discount.
- Purchase Method: The payment method used (e.g., Credit Card, Debit Card, etc.).
- Location: The city where the purchase took place.

# DATA OVERVIEW

| | count | unique | top | freq |
|---|---|---|---|---|
| **Gender** | 55000 | 3 | Female | 18454 |
| **Age Group** | 55000 | 5 | 25-45 | 22010 |
| **Purchase Date** | 55000 | 54988 | 04/07/2022 11:45:29 | 2 |
| **Product Category** | 55000 | 9 | Electronics | 16574 |
| **Discount Availed** | 55000 | 2 | No | 27585 |
| **Discount Name** | 27415 | 5 | NEWYEARS | 8135 |
| **Purchase Method** | 55000 | 8 | Credit Card | 22096 |
| **Location** | 55000 | 14 | Mumbai | 11197 |

- Mean discount of 136.986796 vs median of 0, which suggests the data is right-skewed
- Gross Amount and Net Amount are also skewed to the right
- There are negatives in Net Amount, which probably was mistyped
- The data contains 29071 unique customers with a maximum of 8 purchases per customer
- Gender is divided into 3 categories and majority of females
- Most customers are in the age group of 25-45 y/o
- Electronics is the top category among 9 categories
- 50.15% of the purchases were made without discount
- Credit card was the preferred purchase method among 8 different methods
- Purchases were made from 14 different locations, all locations are in India

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **CID** | 55000.0 | 5.512456e+05 | 2.606033e+05 | 1.000090e+05 | 3.237170e+05 | 5.500885e+05 | 7.769558e+05 | 9.999960e+05 |
| **TID** | 55000.0 | 5.504740e+09 | 2.594534e+09 | 1.000163e+09 | 3.252604e+09 | 5.498383e+09 | 7.747933e+09 | 9.999393e+09 |
| **Discount Amount (INR)** | 55000.0 | 1.369868e+02 | 1.653755e+02 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 2.741150e+02 | 5.000000e+02 |
| **Gross Amount** | 55000.0 | 3.012937e+03 | 1.718431e+03 | 1.364543e+02 | 1.562111e+03 | 2.954266e+03 | 4.342222e+03 | 8.394826e+03 |
| **Net Amount** | 55000.0 | 2.875950e+03 | 1.726128e+03 | -3.511198e+02 | 1.429552e+03 | 2.814911e+03 | 4.211408e+03 | 8.394826e+03 |

# DATA CLEANING & TRANSFORMATION

- Corrected Data types
- Handled missing values
  - Missing discount name replaced by 0
- Cleaned duplicates
- Adjusted negative net values
  - Replaced negatives by 0
- Adjusted discount amounts
  - It shouldn’t be more than the gross amount
- Set purchases to 2 decimals

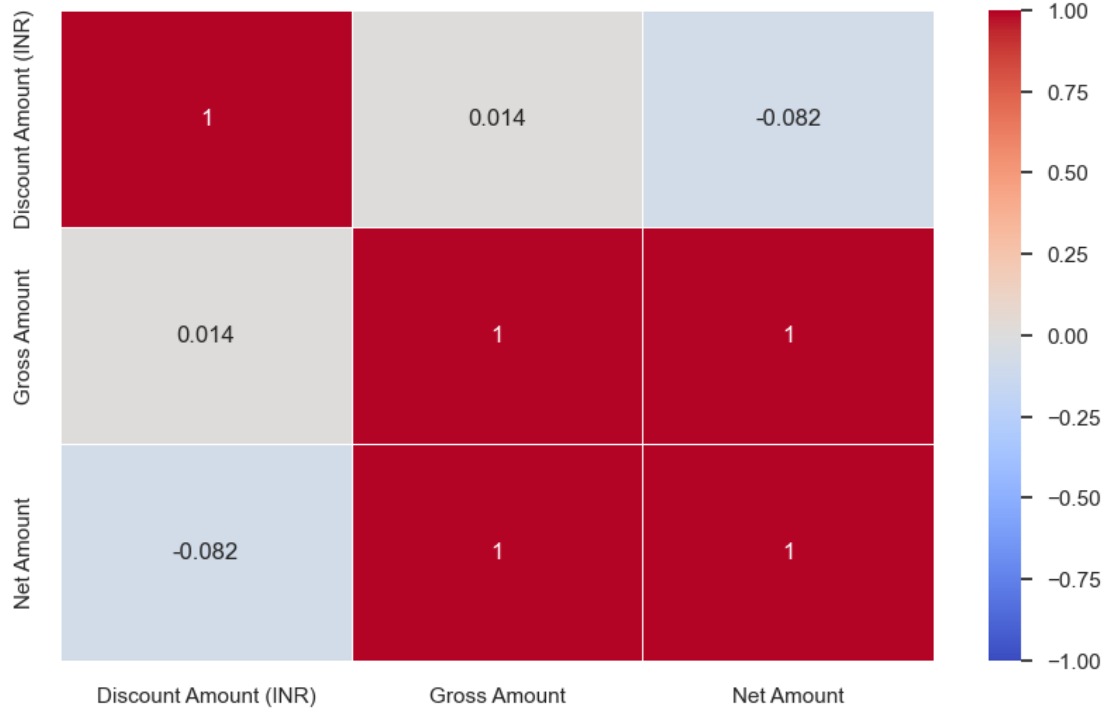
# FEATURE ENGINEERING

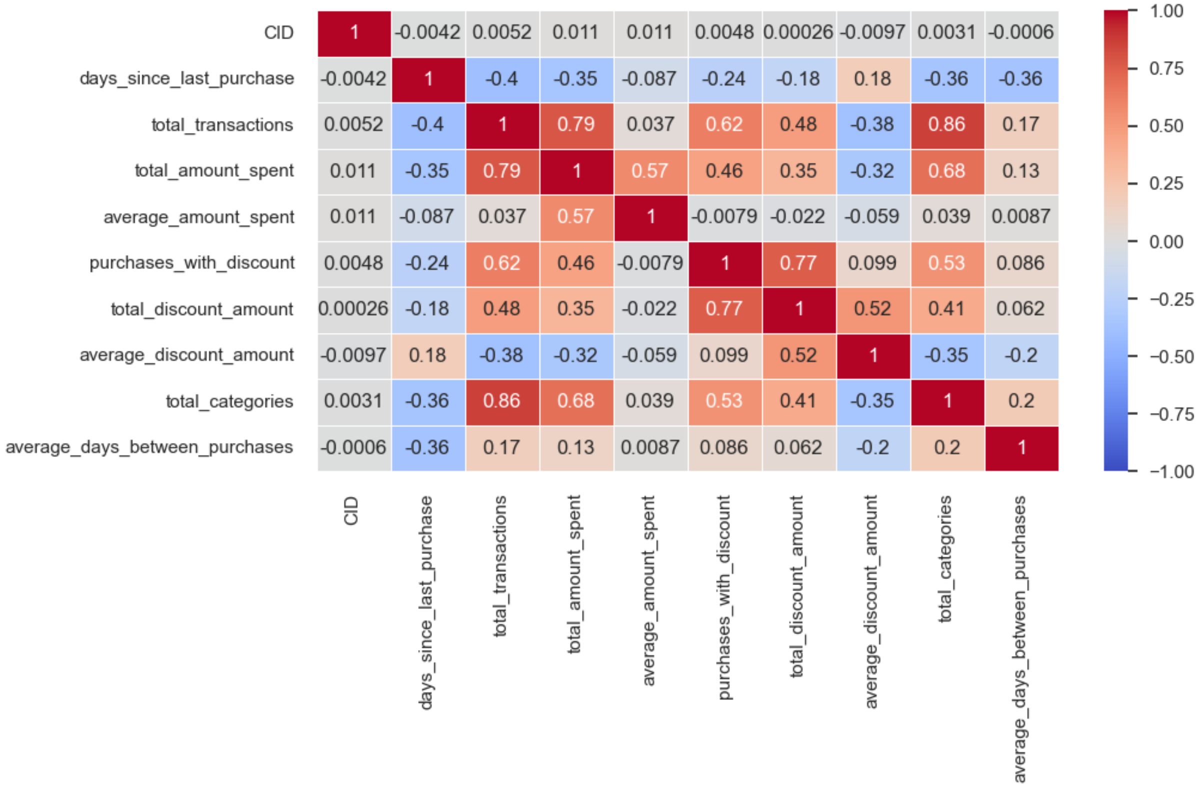
For clustering purposes, we created a table in the following way

- CID: id of the customer
- gender: customer gender
- age_group: customer age group
- days_since_last_purchase: the number of days since last purchase
- total_transactions: the total number of transactions
- total_spent: the total amount spent in all transactions
- avg_spent: the average spent per transaction
- purchases_with_discount: The number of purchases with discount
- total_discount: the total discounts in all transactions
- avg_discount: the average discount per transaction
- product_categories: count of unique purchased categories
- favorite_product_category: favorite product category for each customer
- avg_days_between_purchases: the average days between purchases
- purchase_method: favorite purchase method for each customer
- purchase_location: favorite purchase location

| # | Column | Non-Null Count | Dtype |
|---|---|---|---|
| 0 | days_since_last_purchase | 19478 non-null | int64 |
| 1 | gender | 19478 non-null | category |
| 2 | age_group | 19478 non-null | category |
| 3 | total_transactions | 19478 non-null | int64 |
| 4 | total_amount_spent | 19478 non-null | float64 |
| 5 | average_amount_spent | 19478 non-null | float64 |
| 6 | purchases_with_discount | 19478 non-null | int64 |
| 7 | total_discount_amount | 19478 non-null | float64 |
| 8 | average_discount_amount | 19478 non-null | float64 |
| 9 | total_categories | 19478 non-null | int64 |
| 10 | favorite_product_category | 19478 non-null | category |
| 11 | average_days_between_purchases | 19478 non-null | float64 |
| 12 | favorite_purchase_method | 19478 non-null | category |
| 13 | favorite_location | 19478 non-null | category |
| 14 | cluster | 19478 non-null | int32 |

# EXPLORATORY DATA ANALYSIS
# CORRELATION HEATMAP



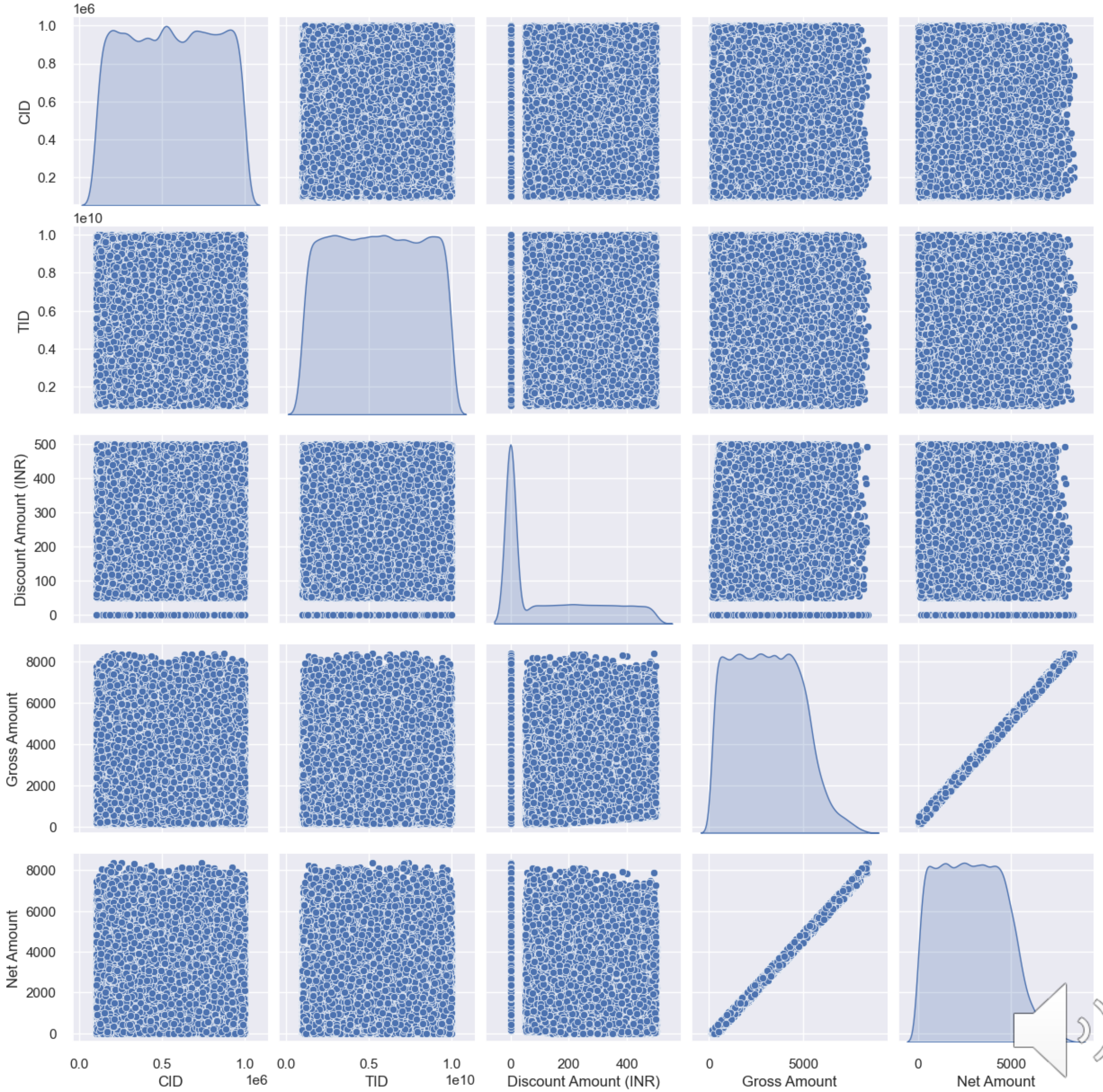




- Gross and net are highly correlated as expected
- Discount is very low correlated to gross and net amounts

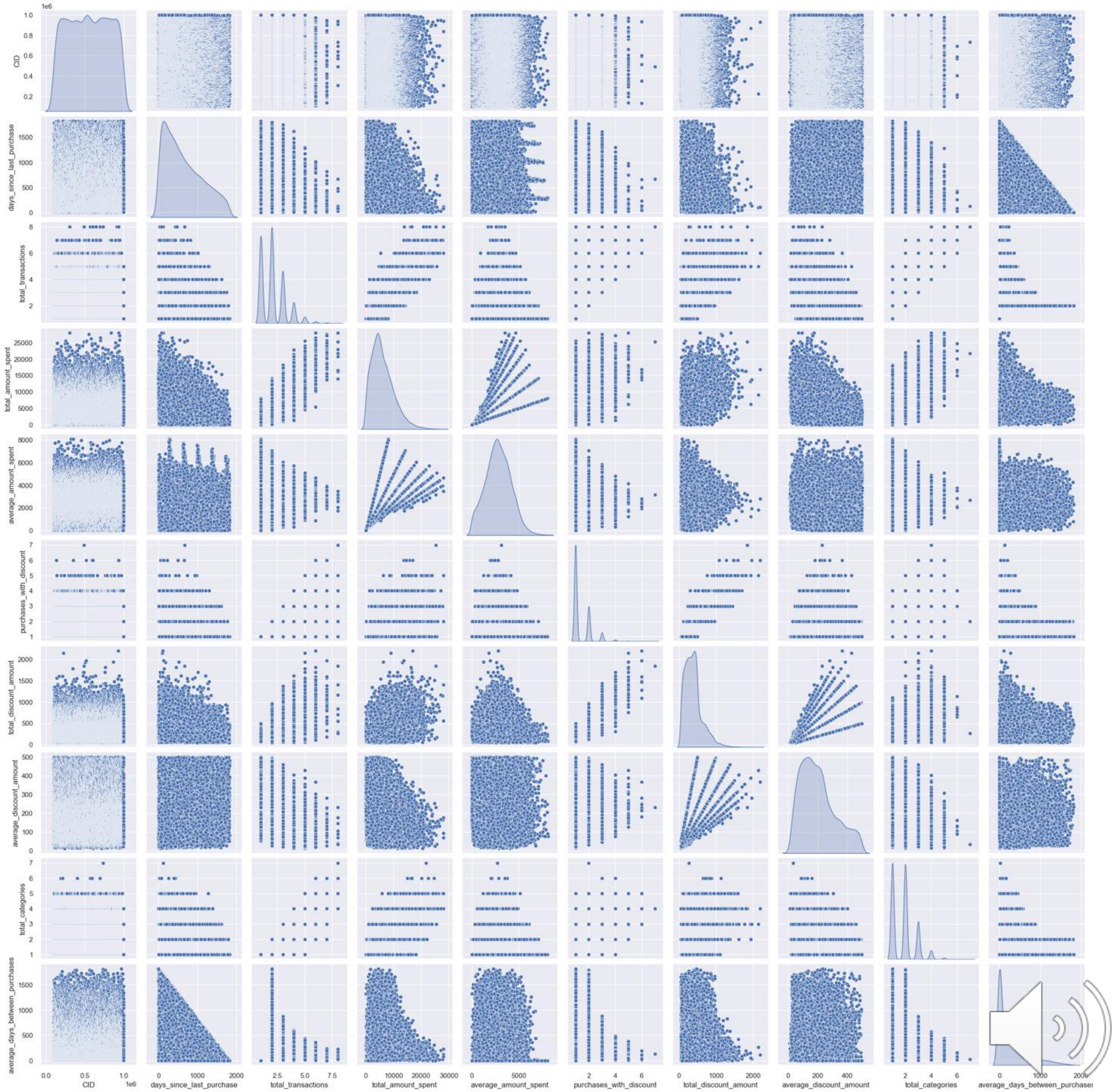
- Total categories are highly correlated with total transactions, which means that usually, customers shop from different categories
- Total transactions are highly correlated with the total amount spent
- Average discount and total discount are also highly correlated
- The total amount spent and total categories are highly correla

# EXPLORATORY DATA ANALYSIS SCATTER PLOT

- Gross and Net amounts are highly correlated with a linear relationship
- For discount amount, we can identify 2 different groups, purchases with 0 discount and purchases with discount.

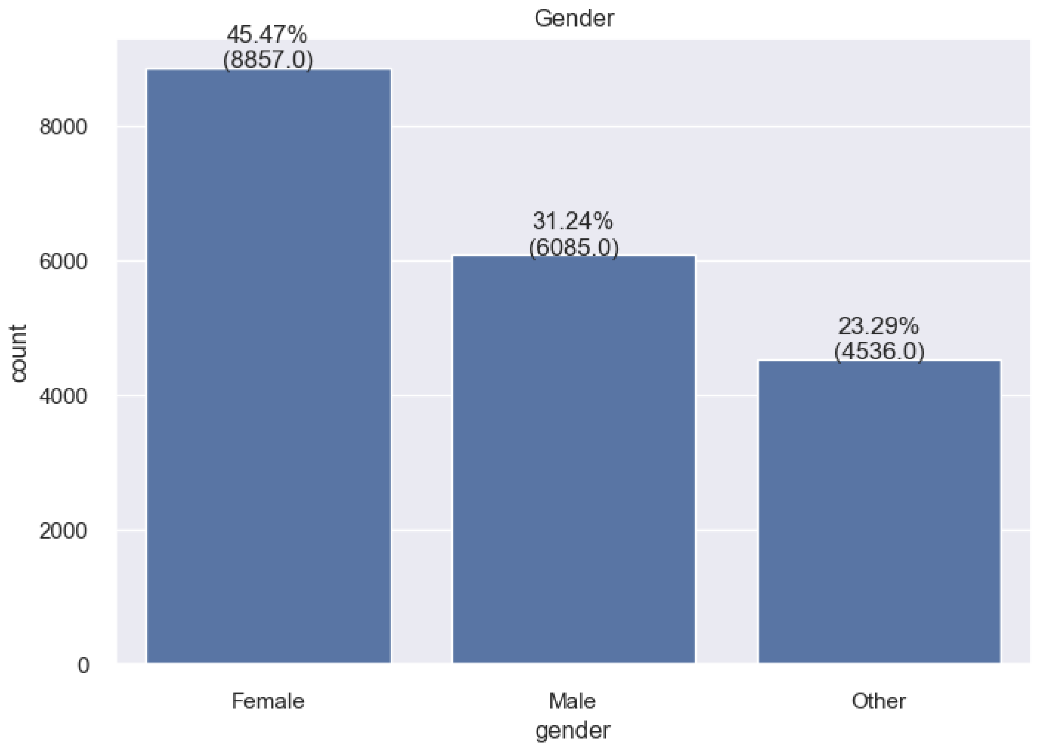


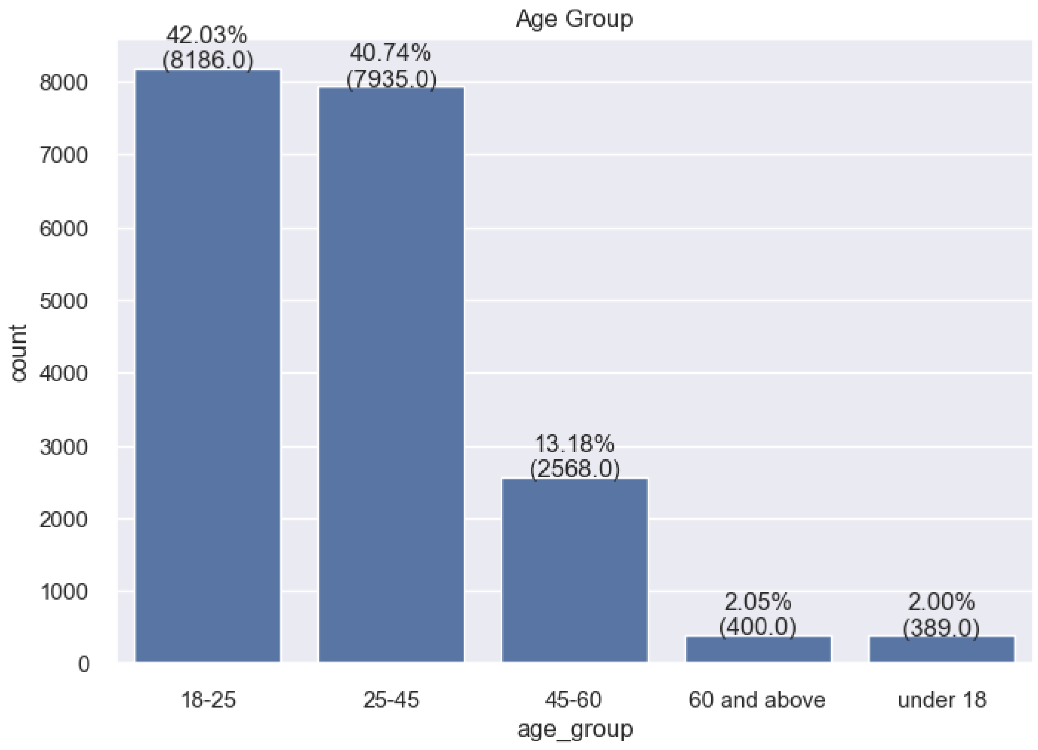
# EXPLORATORY DATA ANALYSIS
# SCATTER PLOT




- People with more transactions tend to buy more often
- We can appreciate linear relation between total_amount and average_amount, and between total_discount and average_discount

# EXPLORATORY DATA ANALYSIS
# CATEGORICAL

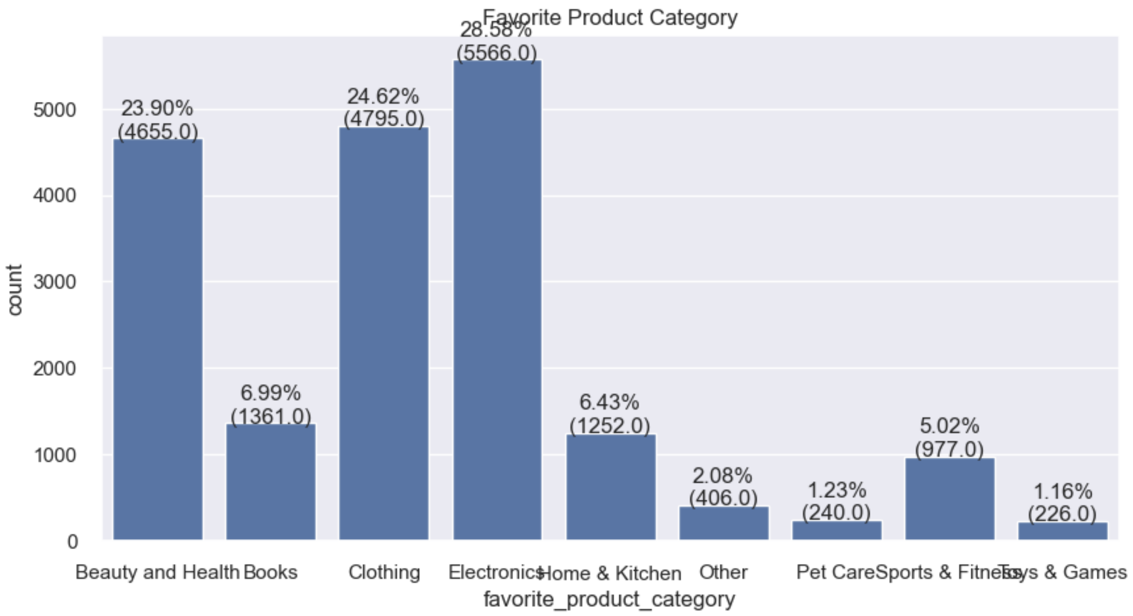


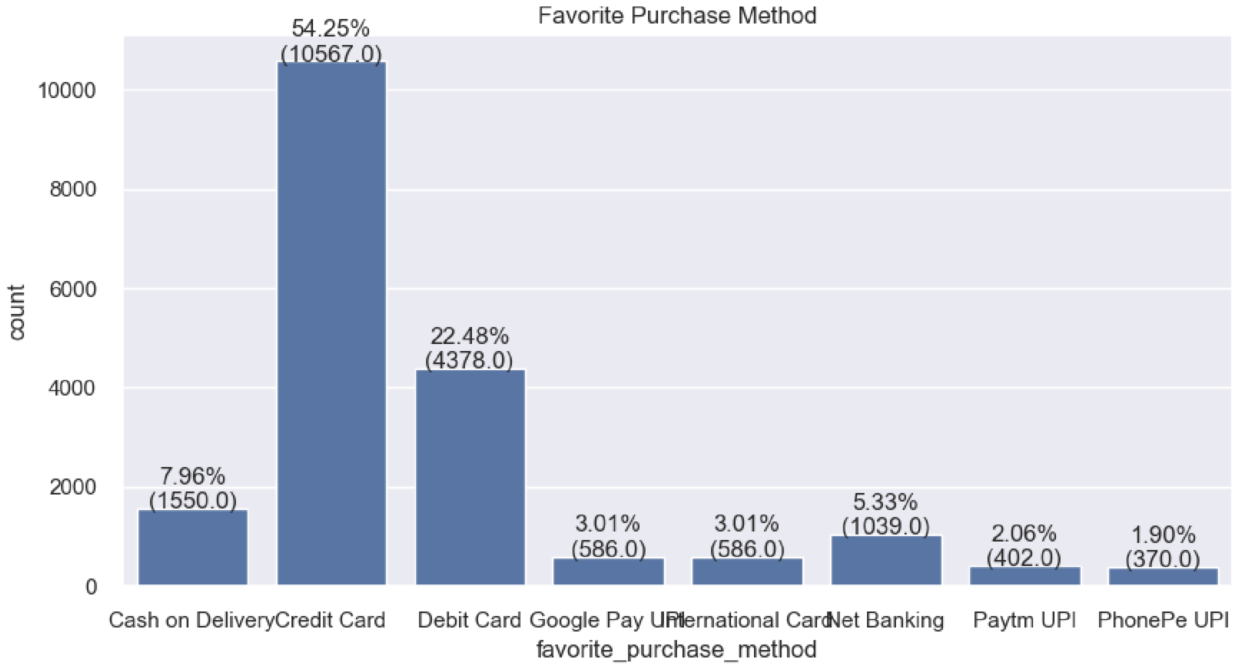




- Most of the customers are female

- ~ 80% of customers in ages between 18-45
- Very few customers in ages above 60 and under 18

# EXPLORATORY DATA ANALYSIS
# CATEGORICAL

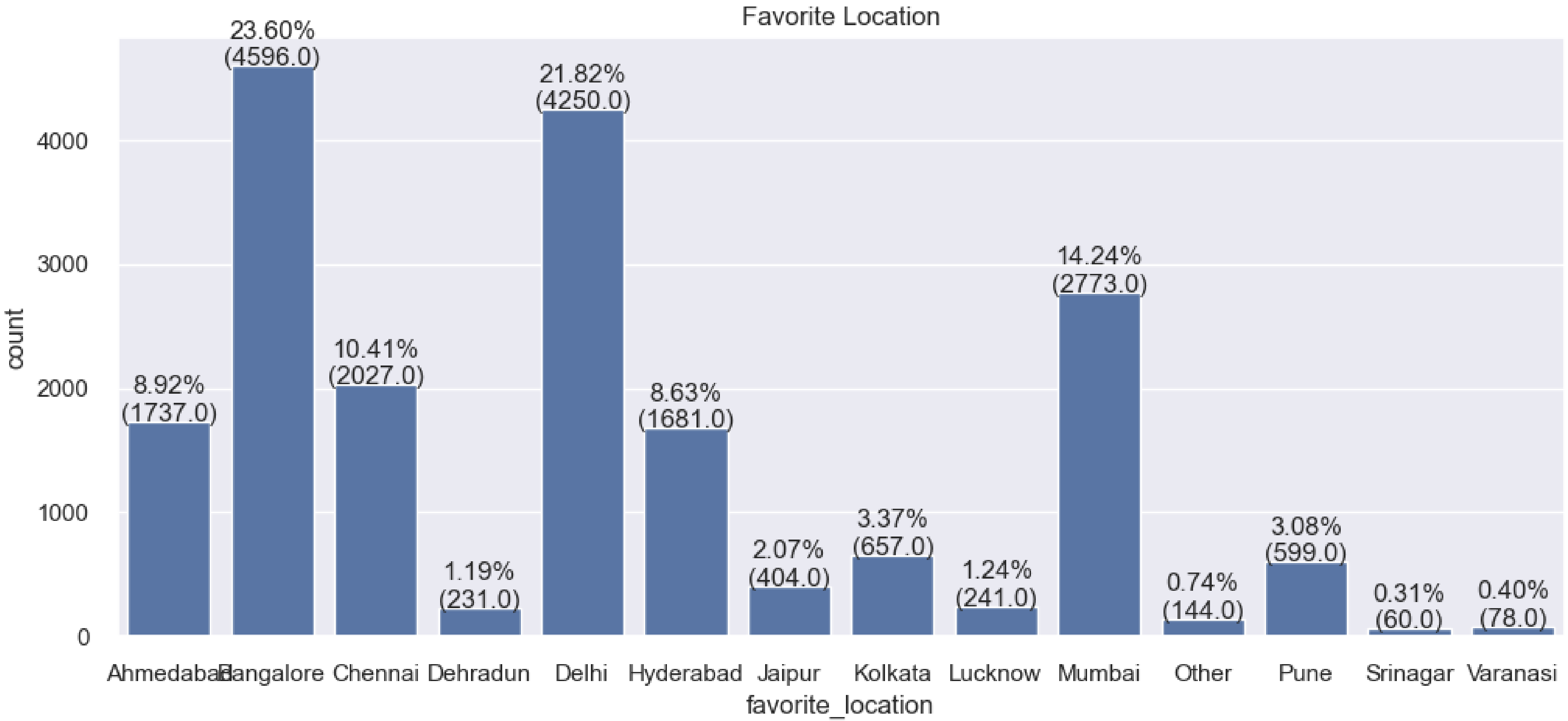



- The most popular categories are Electronics, Clothing, and Beauty and Health (~ 75%)
- Pet care and toys & Games have a very few amount

- Most of the payments are made using credit cards (~74%)
- The second most popular payment method is debit card

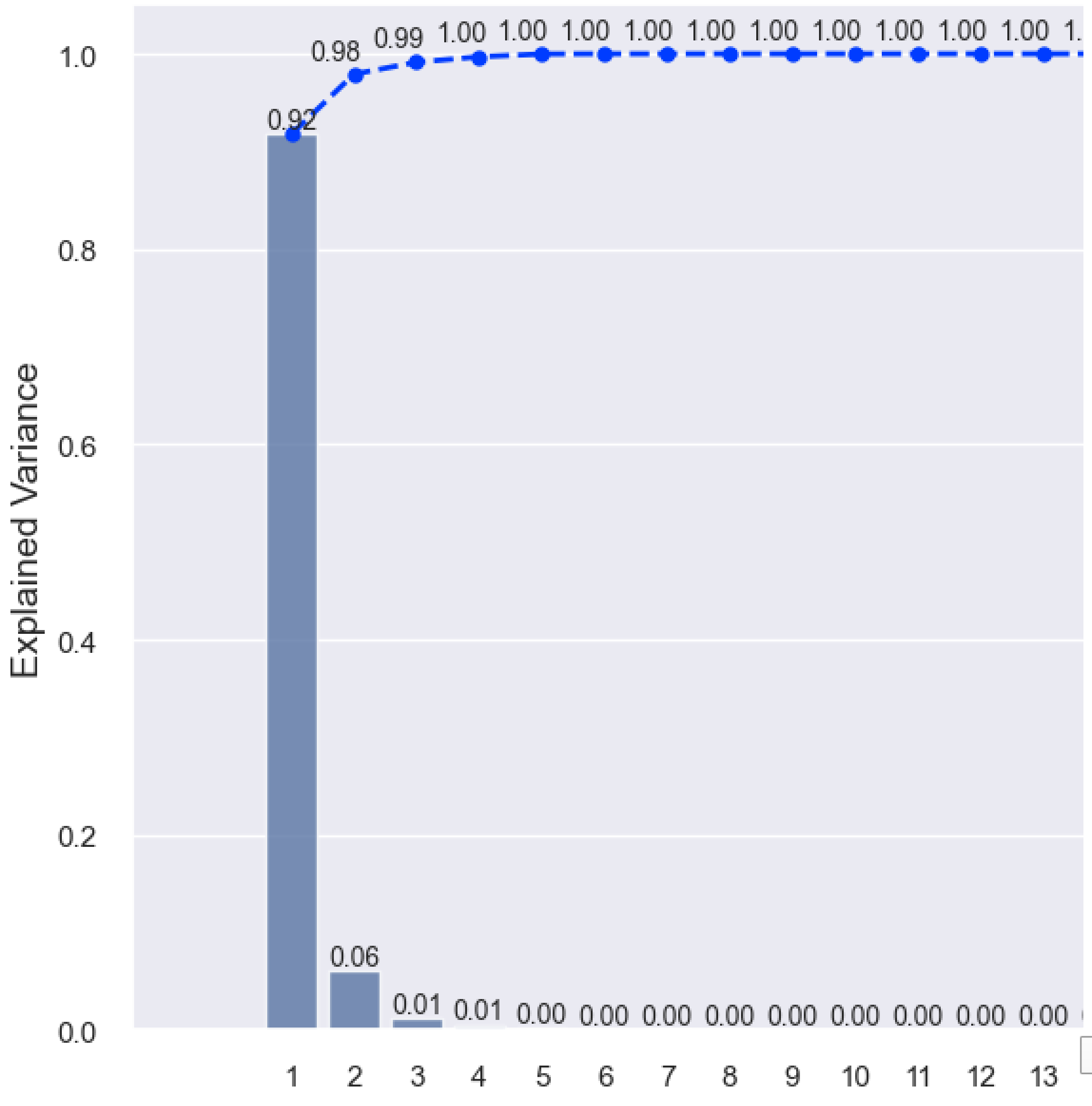
# EXPLORATORY DATA ANALYSIS
# CATEGORICAL




- Of the 29071 customers, ~ 57% are from Bangalore, Delhi, and Mumbai
- Dehradun, Srinagar, Varanasi, and Other have under 1% of customers each

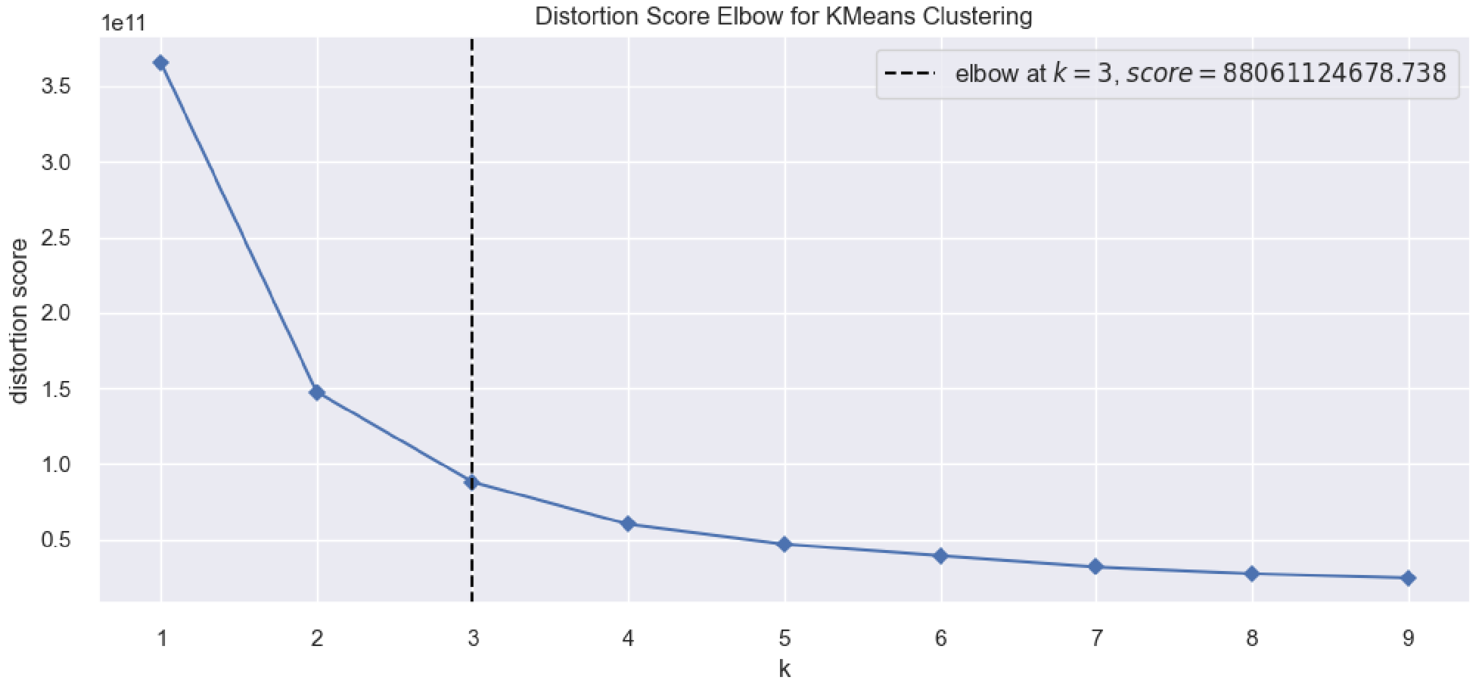
# DATA PREPARATION

- One Hot encode
  - 43 features
- PCA
  - We chose 3 as the number of components, as this explains .99 of the variance



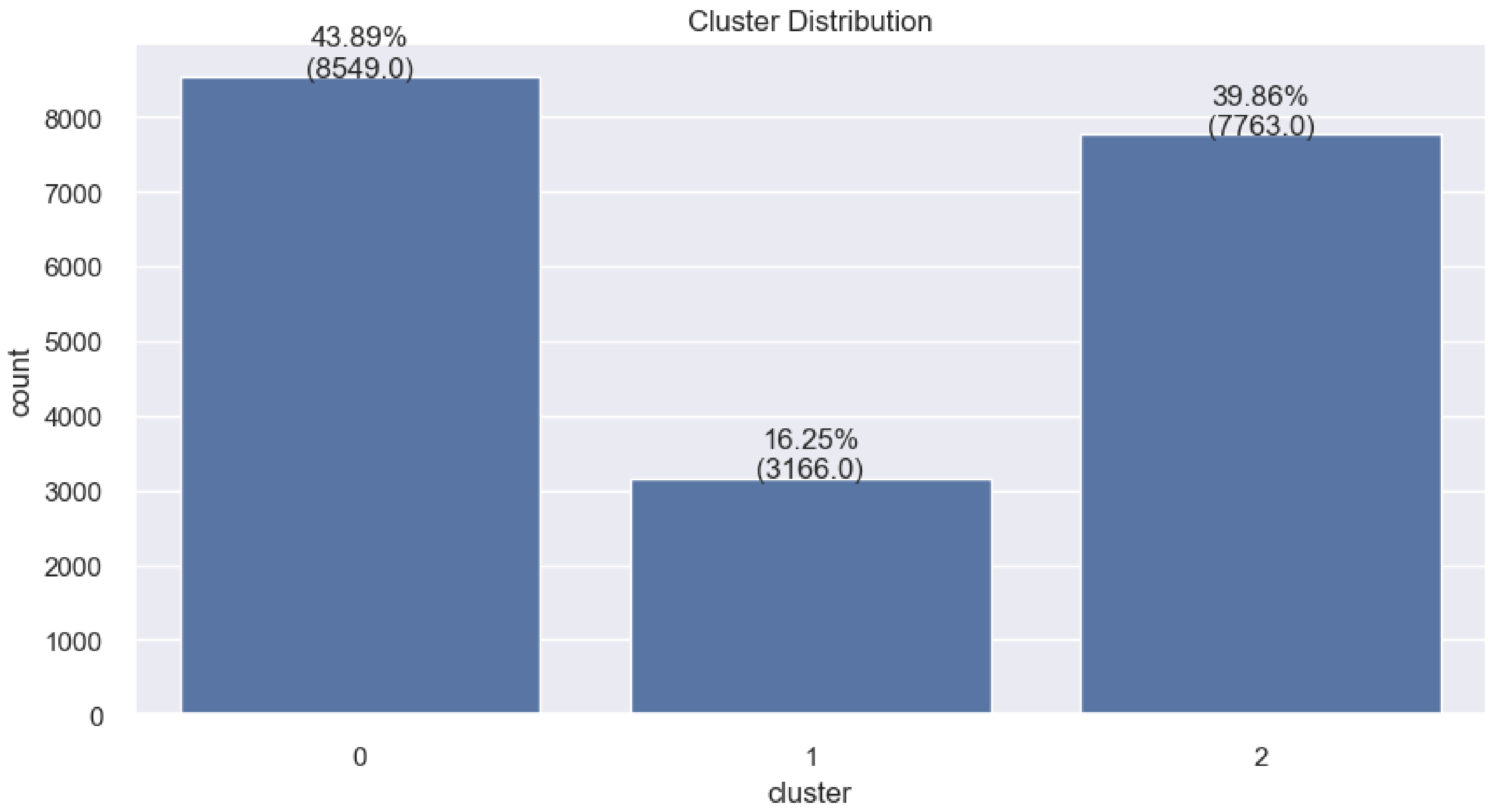
# K-MEANS MODELING
# DETERMINE NUMBER OF CLUSTERS

- Using elbow method
  - Suggest an optimal number of clusters of 3

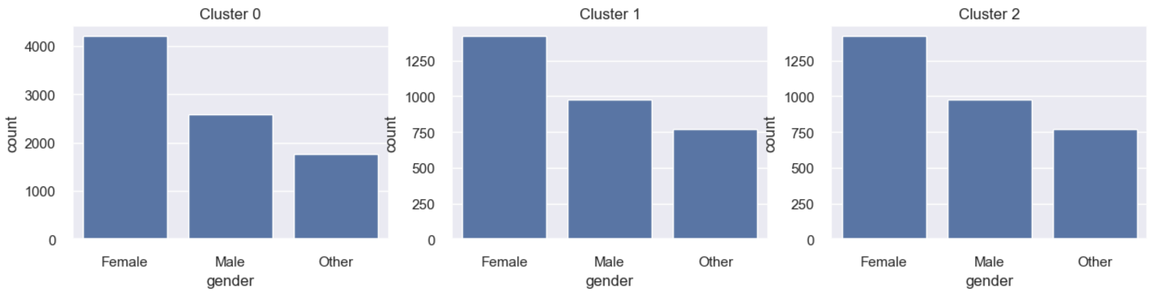


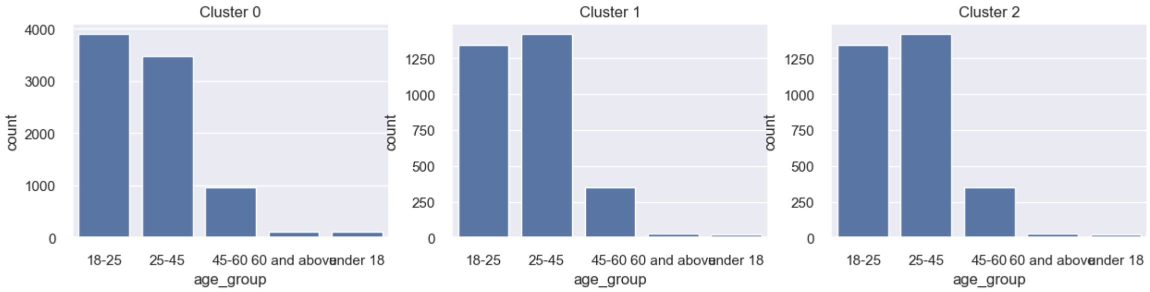
# K-MEANS MODELING CLUSTERING MODEL

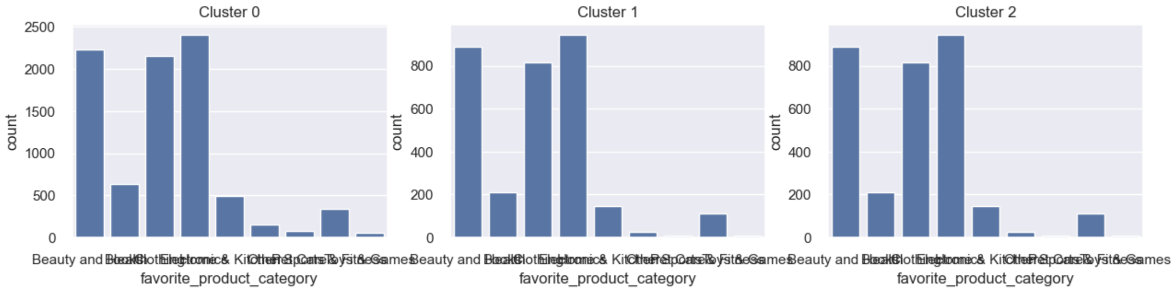
- Clusters 0 and 2 have a balanced amount of customers
- Cluster 1 has the least customers (4638)



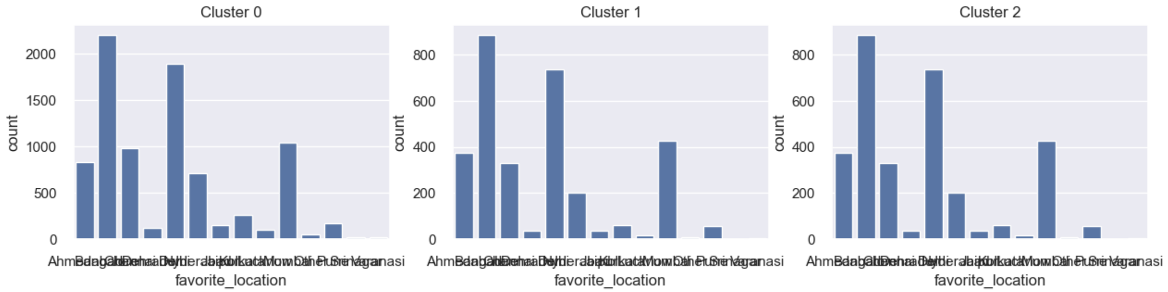
# K-MEANS MODELING CLUSTERING MODEL

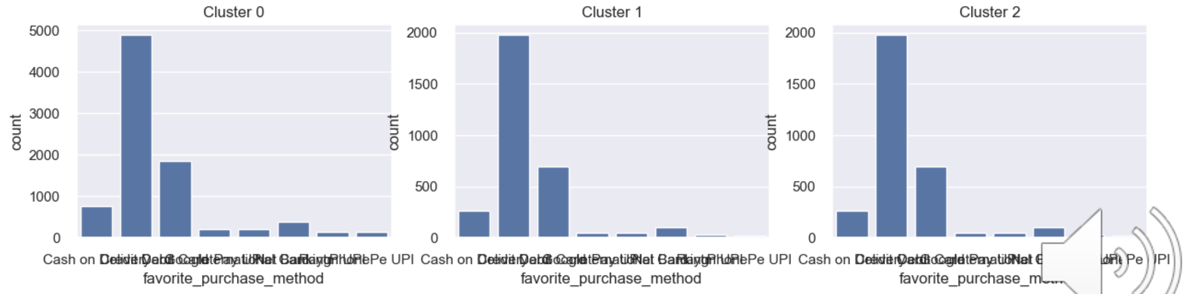



- The 3 clusters have similar distribution for gender, age group, favorite product category, favorite purchase method, and purchase location.

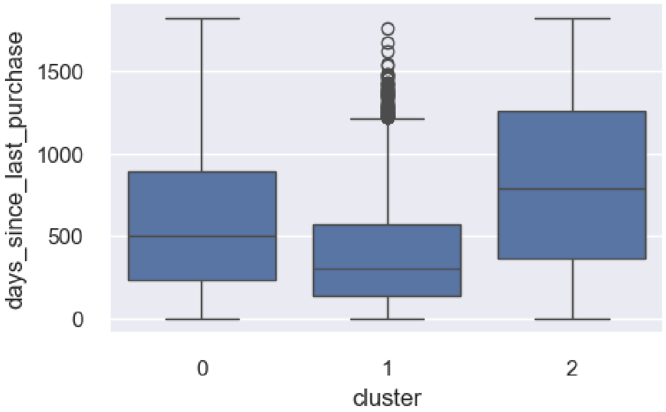


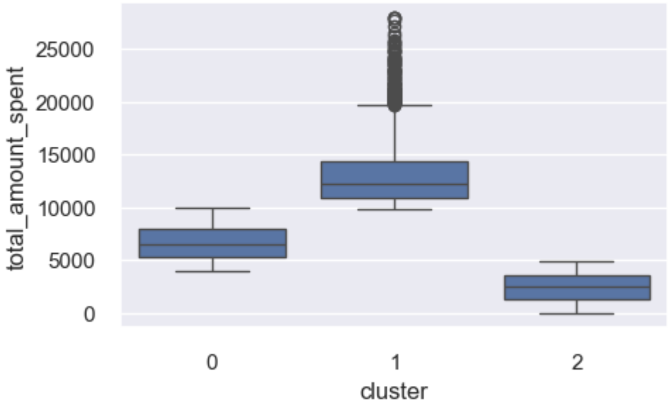



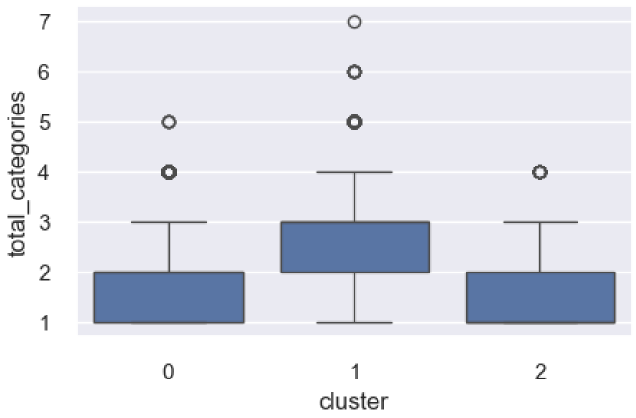
# K-MEANS MODELING CLUSTER DESCRIPTIONS

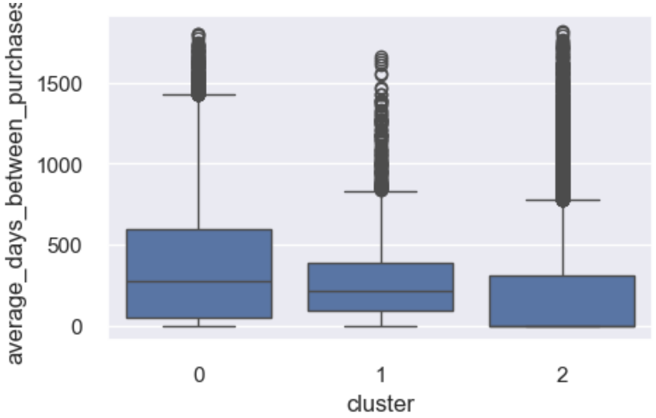
- Cluster 0 - Casual buyers:
- Customers that buys more than once
- All age ranges
- Total spend between 4000 and 9000
- They spend between 2500 to 5000 per purchase
- days between purchases is under 1000 days

- Cluster 1 - Recurrent buyers:
- Customers with many purchases
- Age range from 18 to 60 years old
- They have a total spend of over 10000
- Most of the purchases are with a discount
- They spend between 2500 to 5000 per purchase
- The time between purchases is under 500 days

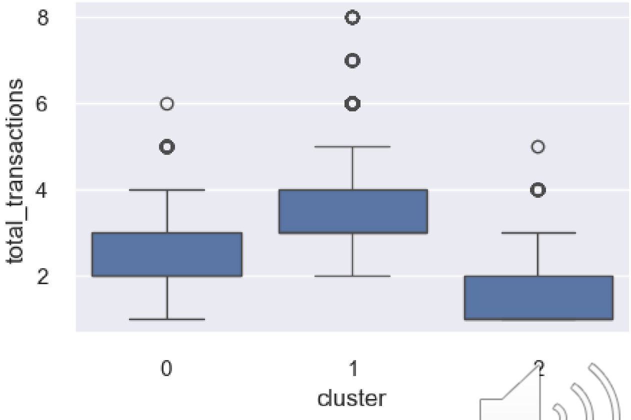



- Cluster 2 - One-time buyers:
- Customers that are one or two-time buyers.
- All age ranges
- buy items from all of the categories
- Spend under 4000 in total
- Spend under 3500 per purchase
- Buys with the highest discount amount
- Buys from one or two product categories



# RECOMMENDER SYSTEM

- Used matrix factorization
- Matrix preparation:
- Given that the amount spent was the most differentiated in the clustering, we will create the matrix in the following order
  - p_cat_1_0 -> count of purchases was more than 0 but less than 2500
  - p_cat_1_2500 -> count of purchases was more than 2500 but less than 3200
  - p_cat_1_3200 -> count of purchases was more than 3200
- In this way, our recommender system can recommend a category and a price range from that category

| Product Category<br>CID | Beauty and Health_0 | Beauty and Health_2500 | Beauty and Health_3200 | Books_0 | Books_2500 | Books_3200 | Clothing_0 | Clothing_2500 | Clothing_3200 | Electronics_0 | Electronics_2500 | Electronics_3200 | Home & Kitchen_0 | Ki |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 100009 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | |
| 100037 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | |
| 100063 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 100089 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | |
| 100096 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

# RECOMMENDER SYSTEM

- Started with 10 components
- Root mean square error : 0.1079

```
from sklearn.decomposition import NMF

# Create an NMF model
nmf = NMF(n_components=10, init="random", random_state=0)
W = nmf.fit_transform(matrix)
H = nmf.components_
V = W @ H
```

```
from sklearn.metrics import root_mean_squared_error

rmse = root_mean_squared_error(matrix, V)
print("RMSE:", rmse)
```

```
RMSE: 0.10788207134618165
```

# RECOMMENDER SYSTEM

- Finding the best parameters
- 34 is the optimal for n_components
- Giving a RMSE of 0.0000503

```python
# lets try different parameters
best_model = None
best_rmse = 999.0

for i in range(1, 40):
    try:
        nmf = NMF(n_components=i, init="random", random_state=0)
        W = nmf.fit_transform(matrix)
        H = nmf.components_
        V = W @ H
        RMSE = root_mean_squared_error(matrix, V)
        if RMSE < best_rmse:
            best_rmse = RMSE
            best_model = W
            print("Best RSME so far: ", RMSE, "n_components:", i)
    except Exception as e:
        continue
```

# RECOMMENDER SYSTEM - TESTING

- We can appreciate the top 5 recommendations for the first 10 users.
- the _0, _2500, and _3200 represent the price range that the user is likely to purchase
- When a user logs in to the e-commerce platform we can check this matrix and recommend to the user products in these categories and in the price range.

```
# Print the top 3 recommendations for the first 5 users
for i in range(10):
    print(f"Recommendations for user {V_optimal.index[i]}: {V_optimal.iloc[i].sort_values(ascending=False).head(5).index.tolist()}")
```

```
Recommendations for user 100009: ['Electronics_3200', 'Clothing_3200', 'Pet Care_2500', 'Toys & Games_0', 'Sports & Fitness_2500']
Recommendations for user 100037: ['Electronics_2500', 'Home & Kitchen_3200', 'Sports & Fitness_2500', 'Beauty and Health_0', 'Home & Kitchen_0']
Recommendations for user 100063: ['Books_0', 'Toys & Games_2500', 'Beauty and Health_0', 'Home & Kitchen_3200', 'Toys & Games_0']
Recommendations for user 100089: ['Home & Kitchen_3200', 'Beauty and Health_3200', 'Electronics_0', 'Other_0', 'Toys & Games_2500']
Recommendations for user 100096: ['Pet Care_3200', 'Beauty and Health_3200', 'Electronics_3200', 'Books_3200', 'Sports & Fitness_2500']
Recommendations for user 100097: ['Clothing_3200', 'Home & Kitchen_3200', 'Beauty and Health_3200', 'Sports & Fitness_2500', 'Books_3200']
Recommendations for user 100139: ['Electronics_3200', 'Clothing_3200', 'Pet Care_2500', 'Toys & Games_0', 'Sports & Fitness_2500']
Recommendations for user 100177: ['Other_3200', 'Home & Kitchen_3200', 'Toys & Games_0', 'Pet Care_2500', 'Home & Kitchen_0']
Recommendations for user 100193: ['Pet Care_2500', 'Home & Kitchen_0', 'Beauty and Health_3200', 'Pet Care_3200', 'Beauty and Health_0']
Recommendations for user 100205: ['Beauty and Health_0', 'Electronics_3200', 'Pet Care_0', 'Clothing_0', 'Electronics_0']
```

# CONCLUSIONS

- We analyzed the data of an e-commerce platform with 55000 entries (purchases)
- We treated missing values, wrong data types, negative values, and adjusted discounts for proper data analysis and modeling
- We had to transform the data before implementing any model to get meaningful insight into each segment
- We implemented unsupervised models for an e-commerce platform
- We implemented k-means for clustering and identified 3 different segments of users with different preferences and behaviors
  - one-time buyers
  - casual buyers
  - recurrent buyers
- We used Non-Negative Matrix Factorization to implement a recommender system that is able to suggest the top categories for each user and the price range that they are likely to purchase.