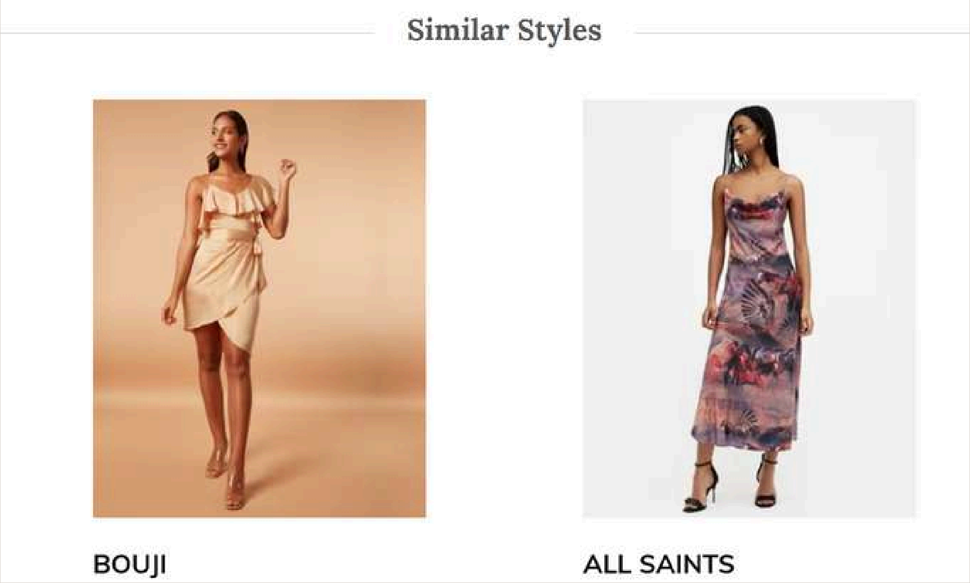
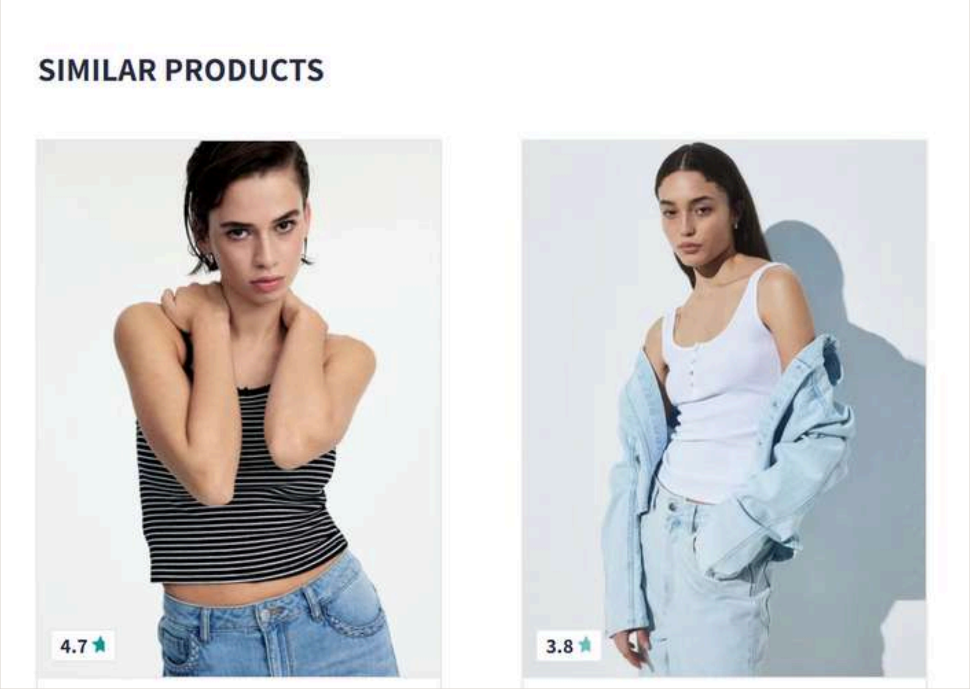
FASHION MARKETING RECOMMENDER SYSTEM

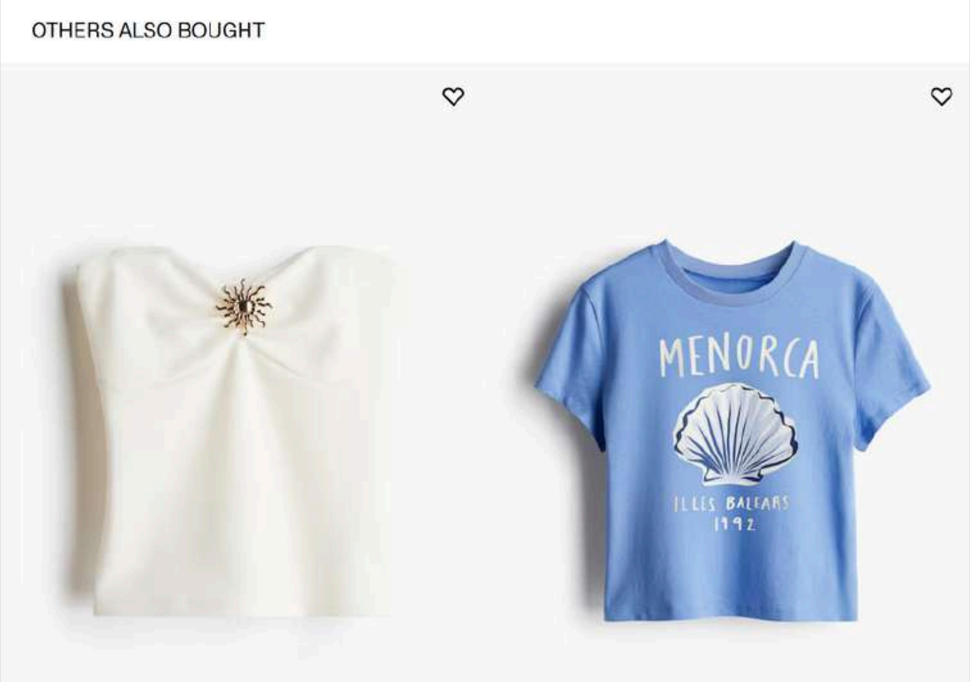
BY: SARA GOYAL, MUSKAAN SAHNI



Ajio Luxe



Myntra



H&M

Current fashion recommendation systems focus on showing **similar products** or **items purchased by others**. Also, recommendations often disregard financial history, creating a **disconnect** rather than truly personalizing recommendations and **reducing conversion rates**. This contributes to the **30-40% return rates.**

This **gap** presents an opportunity to create a more integrated, personalized fashion recommendation system that:

1. That **balances** style aspirations with budget constraints
2. Creates a **seamless, personalized shopping** journey

PROBLEM STATEMENNT

We **aim** to develop a recommendation system that **integrates** transaction history, style preferences, and comprehensive product metadata -to create truly **personalized fashion suggestions** that evolve with each customer's unique style journey while respecting their budget constraints.

By bridging this critical gap in fashion e-commerce personalization, our system aims to reduce return rates, increase conversion, and foster deeper customer loyalty through recommendations that genuinely resonate with each user's individual fashion identity and budget constraint.

LITERATURE SURVEY

Traditional recommender systems like collaborative filtering and content-based models have been widely used in retail to personalize product suggestions. However, in the fashion domain, these often overlook important context such as visual preferences and budget sensitivity.

Recent approaches (He & McAuley, 2016) incorporate image-based features to improve fashion relevance, while others explore outfit compatibility and seasonal trends. Yet, very few models integrate user budget directly into the recommendation logic.

Our work bridges this gap by combining:
Recency-weighted popularity,
Visual similarity via product images, and
Budget-aware filtering using mock prices and age-based budget estimates.
This multi-factor ranking system outputs personalized, affordable, and visually aligned outfit recommendations — validated through a 7-day holdout set and user feedback via survey.

DATASET OVERVIEW

The H&M Personalized Fashion Recommendations dataset is made available by the **H&M Group** through a **Kaggle competition**. This dataset offers unique insights into fashion retail purchasing patterns and customer preferences, making it valuable for developing personalized recommendation systems. It encompasses: **Customer metadata, Product metadata, Detailed transaction records**

Key Dataset Features

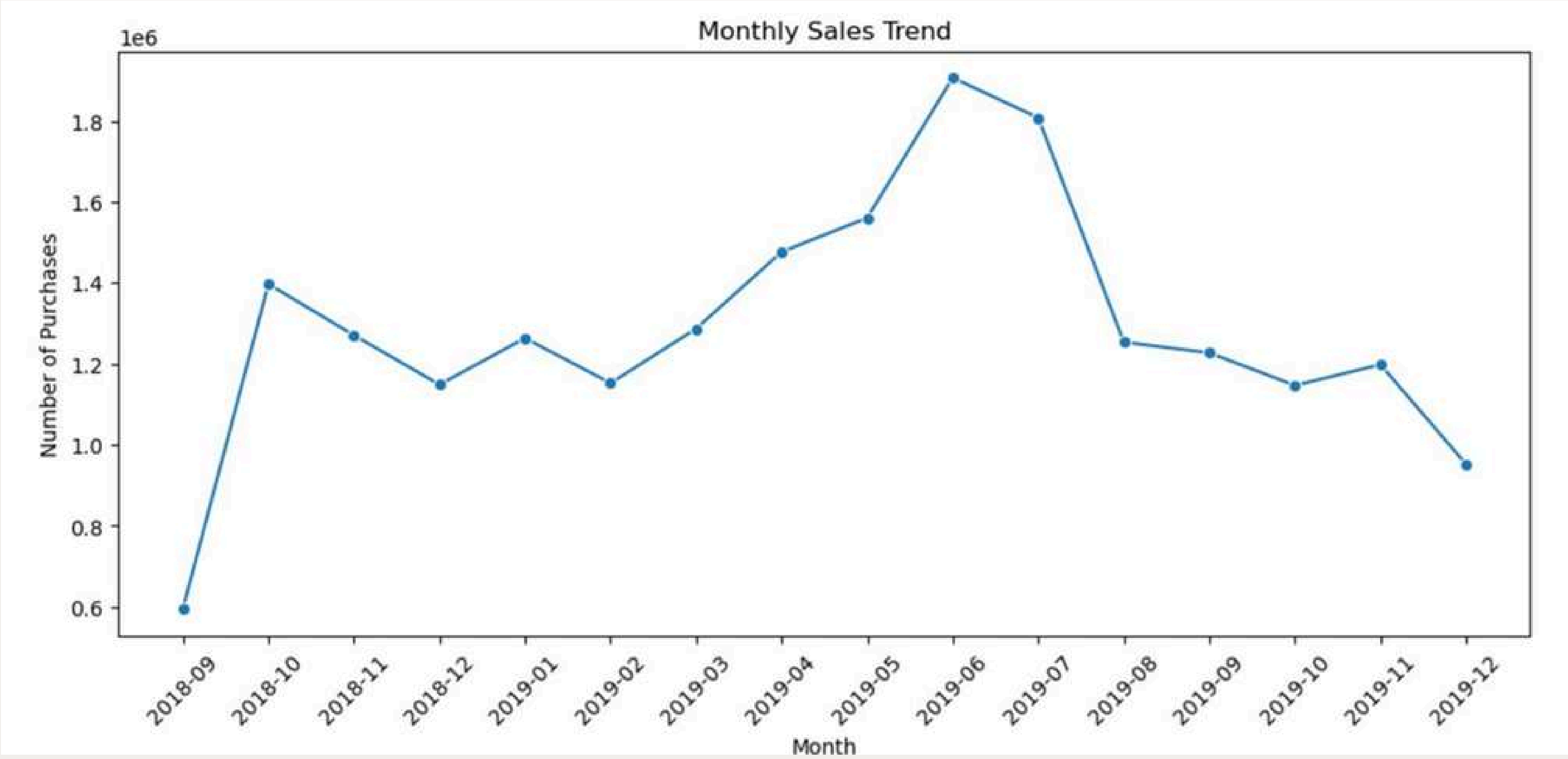
1. **Customer Features: Contains information on 1,371,980 unique customers**
- Demographic information (age, postal code)
- Membership status
- Fashion news frequency
- Purchase history
2. **Product Features: Contains detailed metadata for 105,542 unique fashion articles**
- Detailed product metadata
- Product type
- Color groups
- Graphical appearance
- Department information
- Garment group details
3. **Transaction Features:** Contains 31,788,324 transactions spanning from September 20, 2018, to September 22, 2020.
- Date of purchase
- Article ID
- Price
- Sales channel

DATASET AND FEATURES PREPROCESSING

we tried addressing the **null values** or data entries in our data to make it more efficient and avoid redundancy

```
In [2]: print(articles.head())
        print(customers.head())
        print(transactions.head())

   article_id  product_code          prod_name  product_type_no  \
0   108775015        108775          Strap top              253
1   108775044        108775          Strap top              253
2   108775051        108775      Strap top (1)              253
3   110065001        110065  OP T-shirt (Idro)              306
4   110065002        110065  OP T-shirt (Idro)              306

  product_type_name  product_group_name  graphical_appearance_no  \
0          Vest top  Garment Upper body                  1010016
1          Vest top  Garment Upper body                  1010016
2          Vest top  Garment Upper body                  1010017
3               Bra           Underwear                  1010016
4               Bra           Underwear                  1010016

  graphical_appearance_name  colour_group_code colour_group_name  ...  \
0                     Solid                  9             Black  ...
1                     Solid                 10             White  ...
2                    Stripe                 11         Off White  ...
3                     Solid                  9             Black  ...
4                     Solid                 10             White  ...

   department_name index_code        index_name index_group_no  \
0    Jersey Basic           A        Ladieswear              1
1    Jersey Basic           A        Ladieswear              1
2    Jersey Basic           A        Ladieswear              1
3  Clean Lingerie           B  Lingeries/Tights              1
4  Clean Lingerie           B  Lingeries/Tights              1

   index_group_name section_no            section_name garment_group_no  \
0        Ladieswear         16  Womens Everyday Basics             1002
1        Ladieswear         16  Womens Everyday Basics             1002
2        Ladieswear         16  Womens Everyday Basics             1002
3        Ladieswear         61         Womens Lingerie             1017
4        Ladieswear         61         Womens Lingerie             1017
```

```
print(customers.isnull().sum())

article_id                        0
product_code                      0
prod_name                         0
product_type_no                   0
product_type_name                 0
product_group_name                0
graphical_appearance_no           0
graphical_appearance_name         0
colour_group_code                 0
colour_group_name                 0
perceived_colour_value_id         0
perceived_colour_value_name       0
perceived_colour_master_id        0
perceived_colour_master_name      0
department_no                     0
department_name                   0
index_code                        0
index_name                        0
index_group_no                    0
index_group_name                  0
section_no                        0
section_name                      0
garment_group_no                  0
garment_group_name                0
detail_desc                     416
dtype: int64
customer_id                     0
FN                         895050
Active                     907576
club_member_status           6062
fashion_news_frequency      16009
age                         15861
postal_code                     0
```



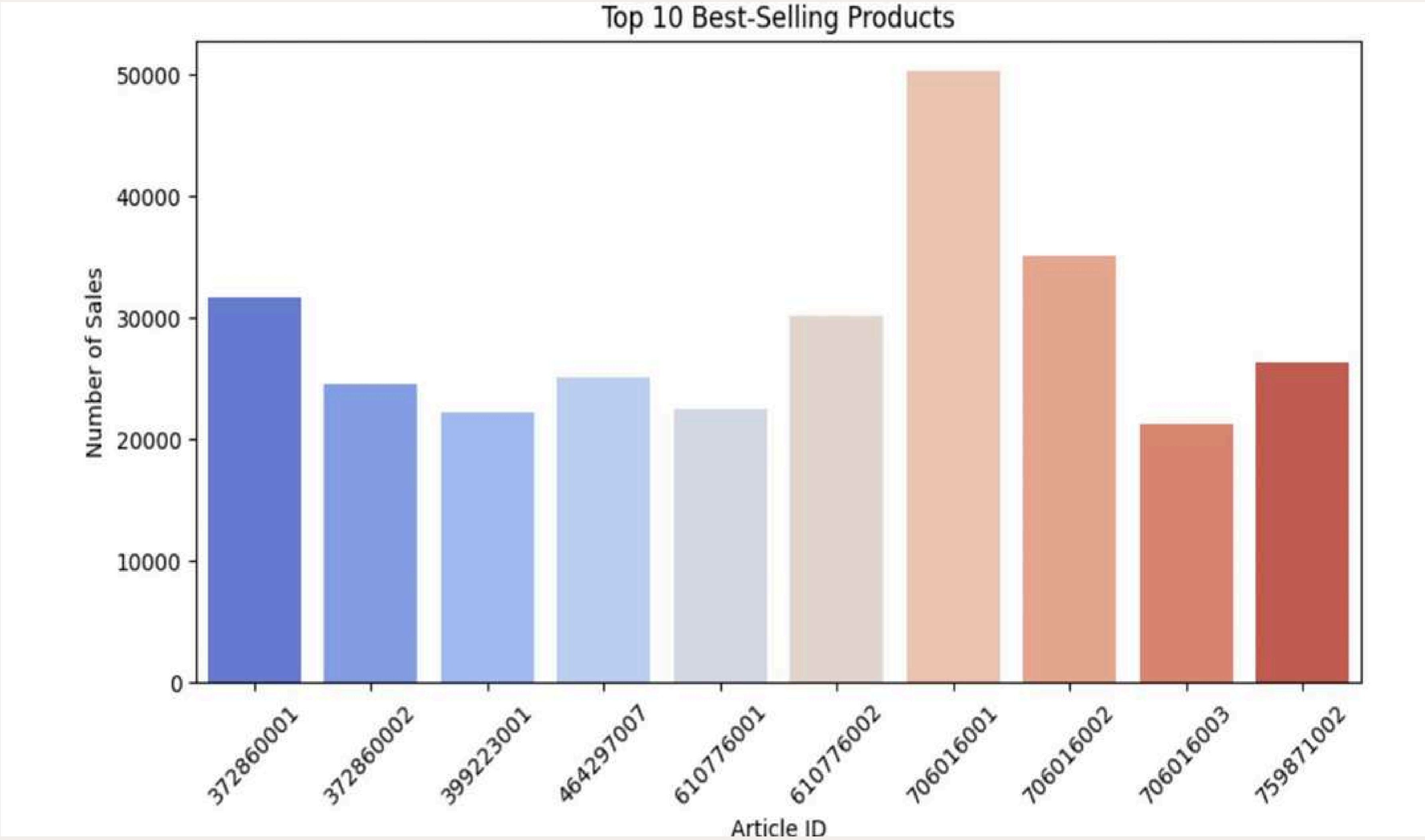
Highest sales peak around **June-July 2019**
Another peak in **October 2018**
Decline from **July 2019 to December 2019**.

Causes for the Peaks
October-November 2018: Possibly a holiday or Black Friday effect.
June-July 2019: Could be related to summer sales or fashion seasonality.

Overall Sales Growth and Drop From September 2018 to mid-2019, sales increased, indicating growing customer engagement. However, the sharp drop after mid-2019 suggests either less demand, fewer promotions, or data limitations

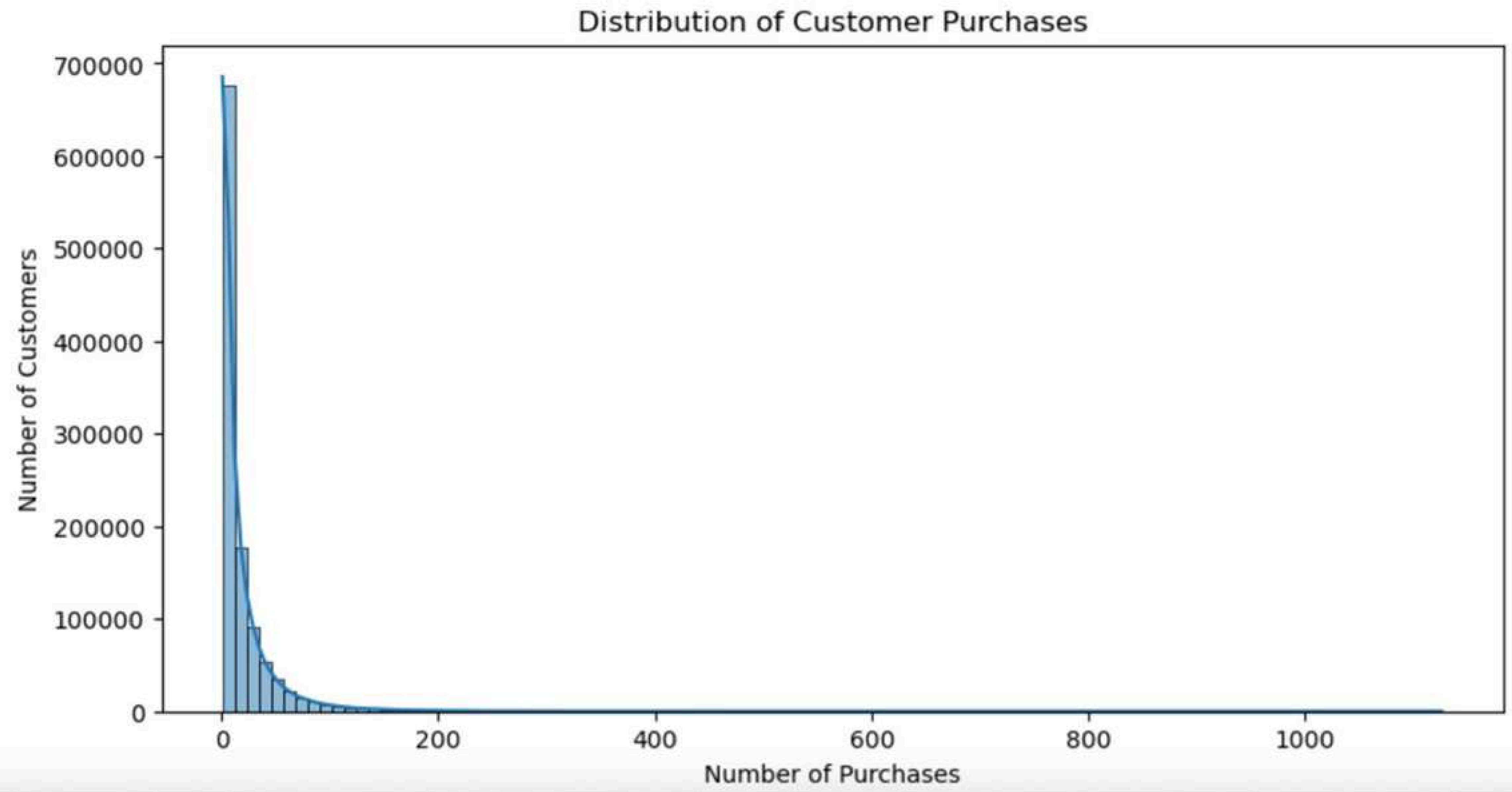
-selling product (**Article ID 706016001**) has over 50,000 sales, significantly higher than the rest.
-Other products have a moderate distribution, with sales ranging from 20,000 to 40,000.

The **best-selling products might indicate general customer trends.** If products in the top 10 are out of stock, alternative recommendations should be suggested.

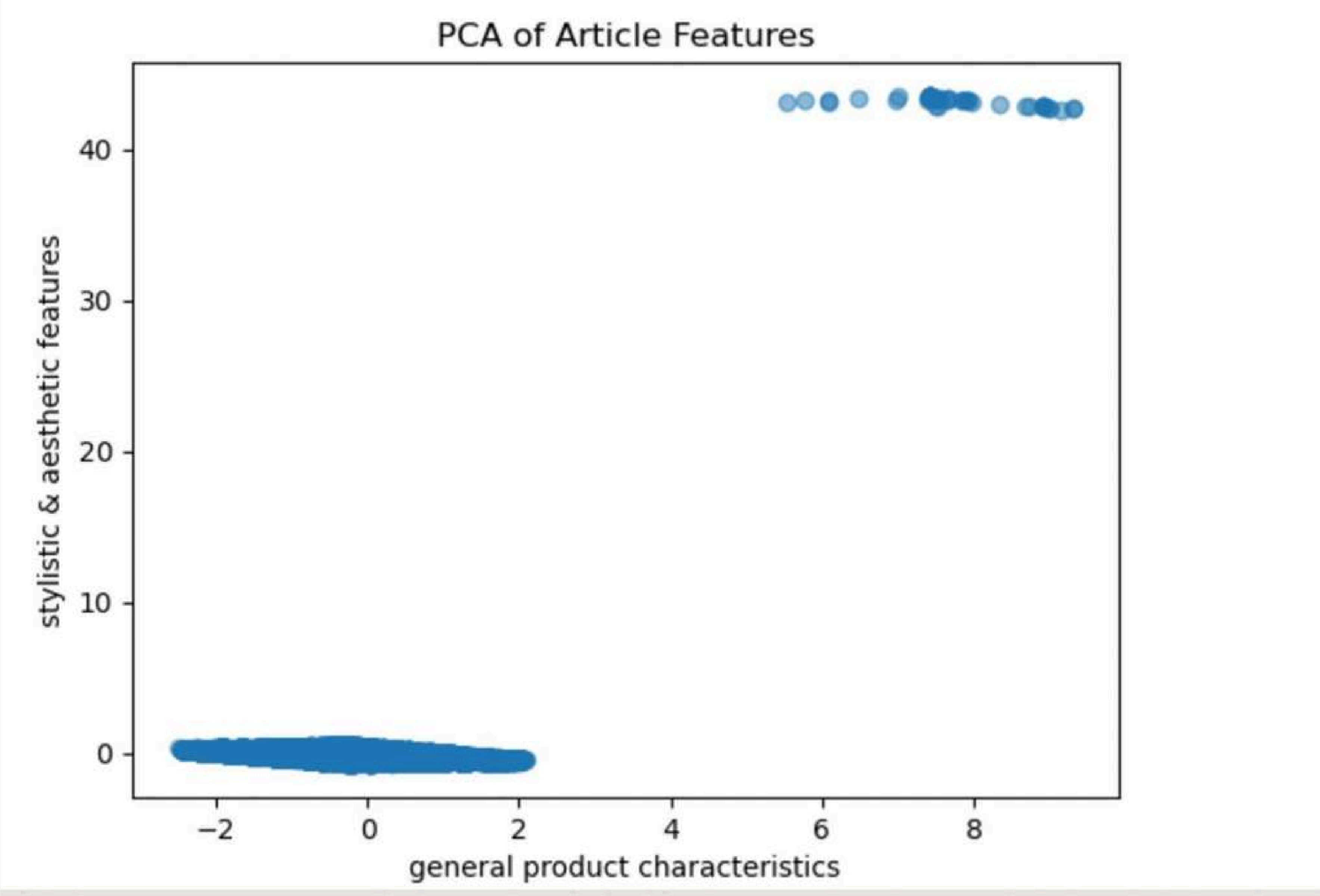


This histogram visualizes the distribution of customer purchases in the H&M dataset, showing **how frequently customers make purchases**.The majority of customers have very few purchases. There is a long tail, meaning a small subset of customers makes a significantly higher number of purchases. Most customers have between **1 to 10** purchases, while **very few exceed 100 purchases**.The highest peak (mode) is at 1 or 2 purchases, suggesting that most customers buy only once or twice.

```
import matplotlib.pyplot as plt
import seaborn as sns

customer_purchases = transactions.groupby("customer_id")["article_id"].count()

plt.figure(figsize=(10, 5))
sns.histplot(customer_purchases, bins=100, kde=True)
plt.xlabel("Number of Purchases")
plt.ylabel("Number of Customers")
plt.title("Distribution of Customer Purchases")
plt.show()
```



```
plt.scatter(principal_components[:, 0], principal_components[:, 1], alpha=0.5)
plt.title('PCA of Article Features')
plt.xlabel('general product characteristics')
plt.ylabel('stylistic & aesthetic features')
plt.show()
```

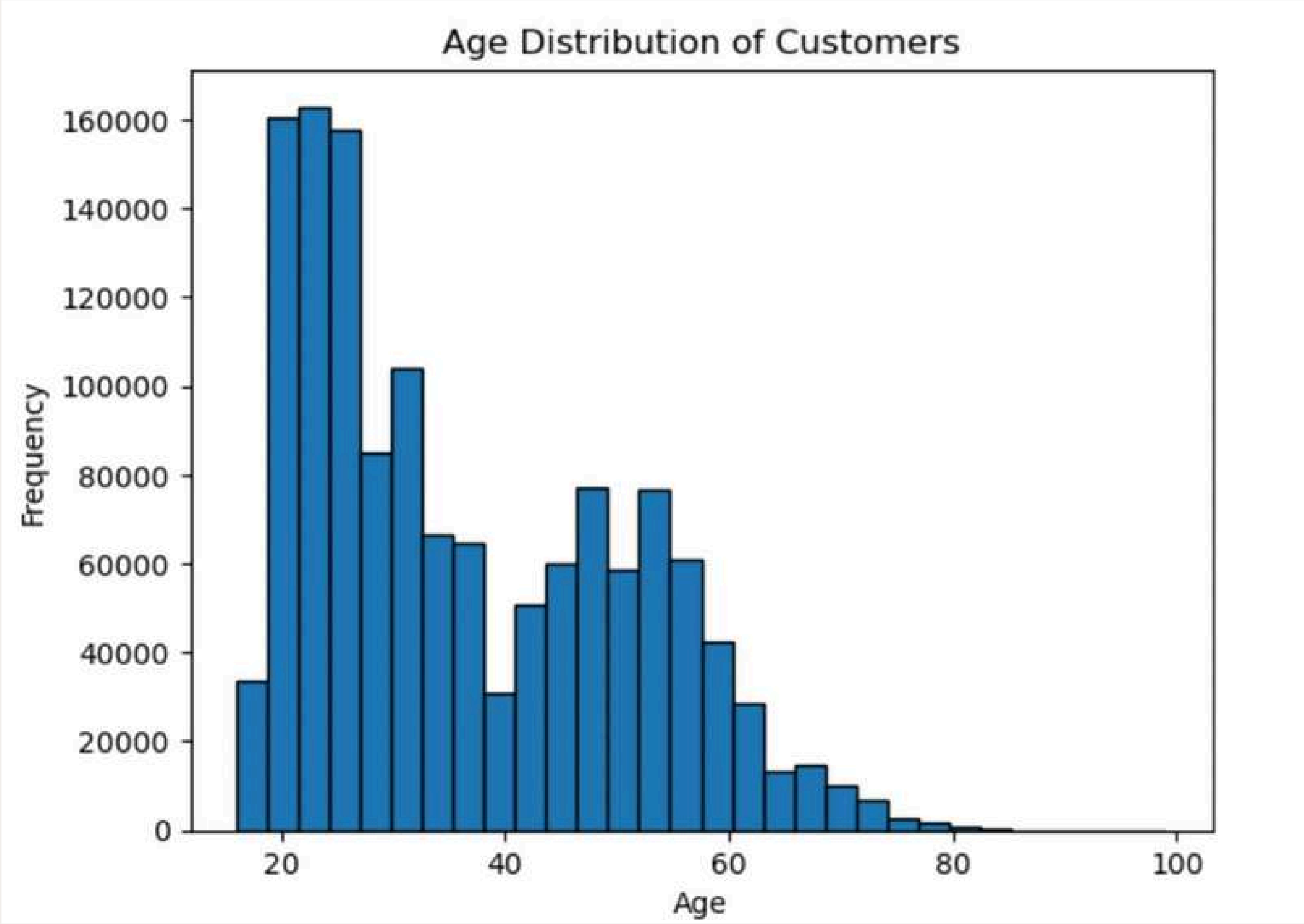


PCA has reduced the high-dimensional fashion dataset into **two** principal components, summarizing key variations in the dataset.

-The data appears to cluster into **two distinct groups**:

- A large cluster near the bottom → These could be basic clothing items with minimal stylistic variation.
- A smaller, separate cluster at the top → These likely represent fashion articles with strong aesthetic/style differentiation

```
import matplotlib.pyplot as plt

plt.hist(customers['age'].dropna(), bins=30, edgecolor='k')
plt.title('Age Distribution of Customers')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



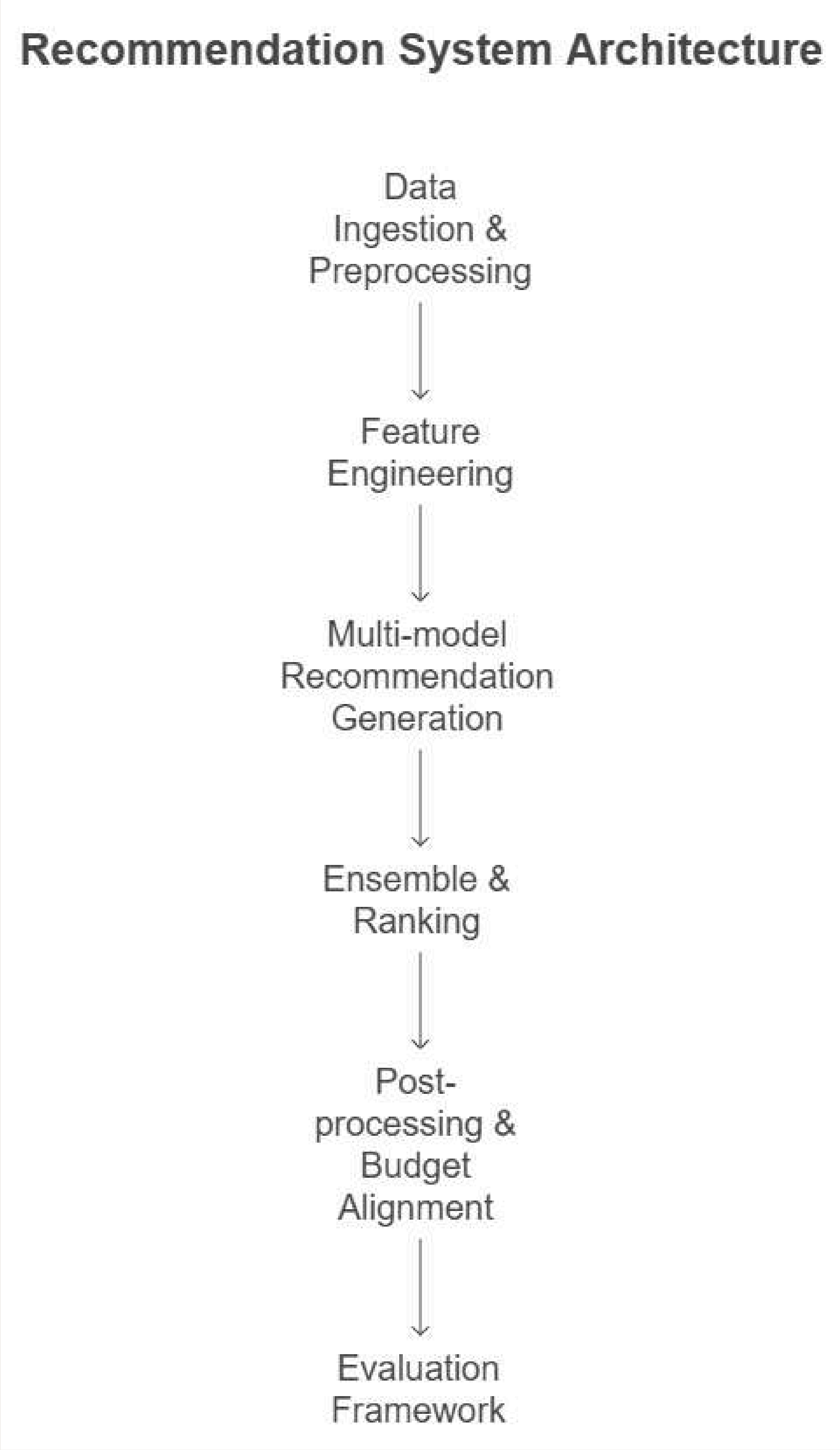
- There is a peak around **20-25 years**, indicating that most customers belong to this age group.-There are also minor peaks
- around 40 and 55 years, suggesting other potential customer segments.
- The distribution is **right-skewed,** meaning older customers are fewer. If there are outliers (e.g., customers above 80).

METHODOLOGY

The **preprocessing** creates several derived features:

- Time-based features (days of week, recency metrics)
- Purchase frequency counts
- Weekly sales aggregations
- Product co-occurrence matrices
- Age group segmentation
- Product popularity metrics

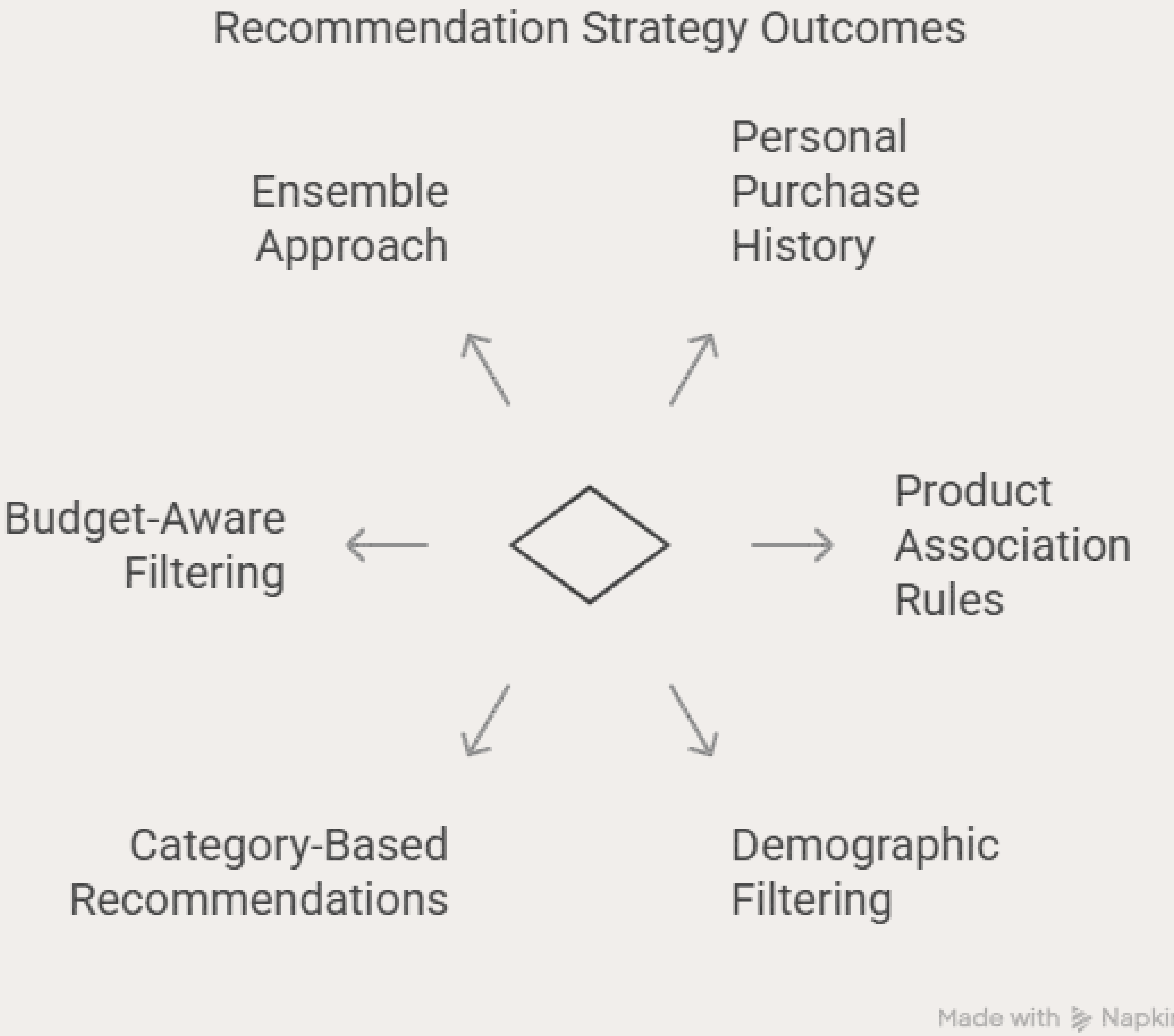
- Date conversion: `transactions_train['t_dat'] = pd.to_datetime(transactions_train['t_dat'])`
- Feature renaming: `articles.rename(columns={'detail_desc':'detail_feature_no', 'index_group_no':'population_no'}, inplace=True)`
- Category encoding: `articles['detail_feature_no'] = pd.factorize(articles['detail_feature_no'])[0]`
- Age grouping: `customers['age_group'] = pd.cut(customers['age'], bins, labels=names, right=False)`



Feature Engineering

The system creates a set of features that capture customer behavior and product characteristics:

1. **Temporal Features:** These features capture timing patterns in customer purchases.
2. **Popularity and Trend Features:** This helps the system prioritize products that are gaining momentum in the market. Represents a product's current popularity relative to its historical baseline, effectively identifying rising or falling trends.
3. **Purchase Frequency Features:** The system captures repeat purchase behavior.
4. **Co-Purchase Features:** This creates powerful association rules that capture patterns like "customers who bought X also bought Y".



Core Recommendation Strategy

1. **Personal Purchase History Analysis**: Identifies items a customer has purchased recently and frequently, with a **time-decay function** that weighs recent purchases more heavily.
2. **Product Association Rules**: Analyzes which products are c**ommonly purchased together,** similar to market basket analysis, to recommend complementary items.
3. **Demographic Filtering**: Creates **age-specific recommendation** pools based on what's popular among similar demographic groups.
4. **Category-Based Recommendations**: Recommends products from categories the customer has shown interest in previously.
5. **Ranking Mechanism:**

- Value-Based Ranking The primary ranking mechanism uses the calculated 'value' score.
- Age-Specific Popularity Ranking, Within each age segment, products are ranked by popularity.
- Purchase Frequency and Recency Ranking.

6. **Cold-Start Handling:** For customers with no purchase history, the system falls back to demographic recommendations
7. **Budget-Aware Filtering**: In the second code segment, there's an additional layer that estimates customer budgets based on age and filters recommendations to match these budgets.
8. **Price-Distance Calculation and Re-ranking**: The system then calculates how well each recommended item aligns with the customer's budget
9. **Ensemble Approach**: The final recommendations combine outputs from these different strategies.

MACHINE LEARNING METHODS USED

1. Item-to-Item Collaborative Filtering

2. Time-Weighted Recommendations

3. Demographic-Based Filtering

4. Content-Based Filtering

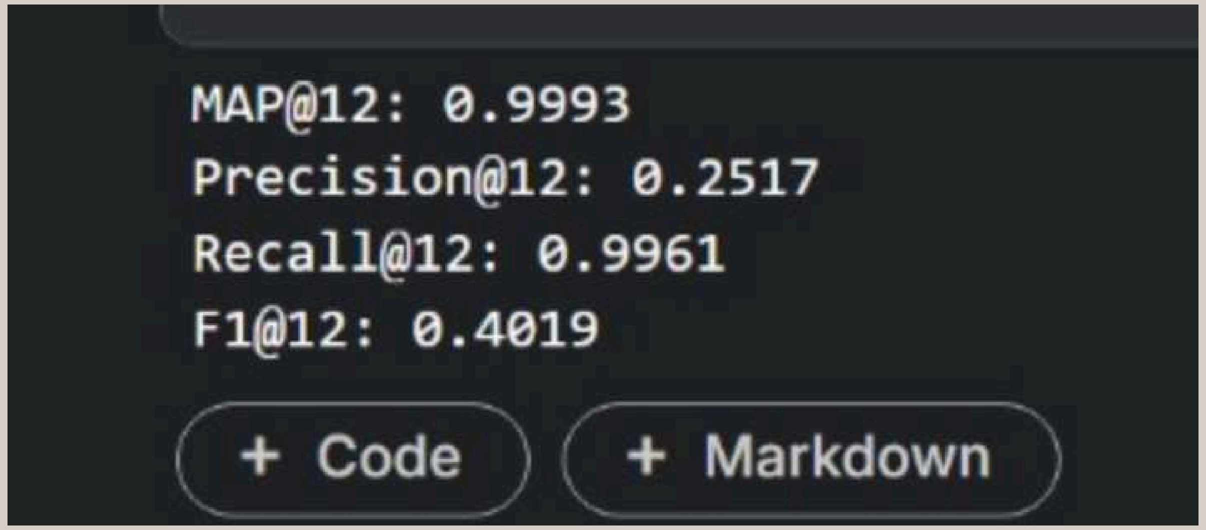
5. Ensemble Method

PERFORMANCE METRICS

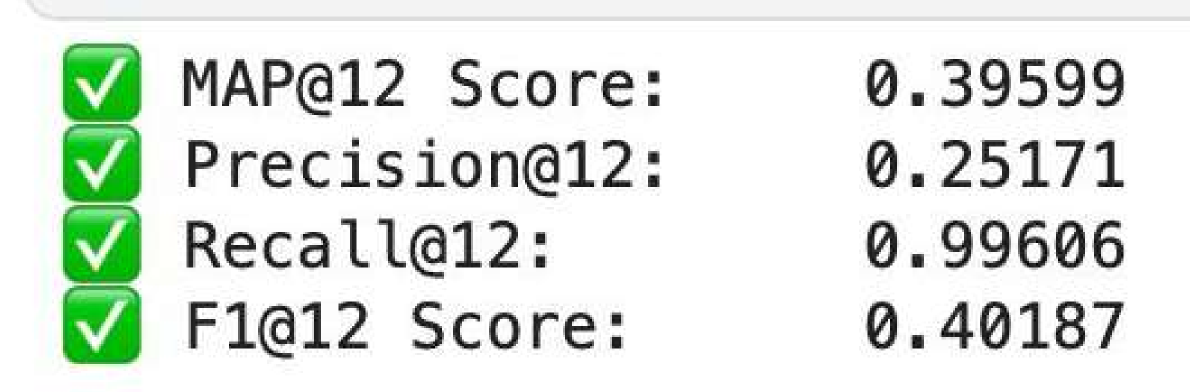
1. MAP@12 (Mean Average Precision at 12)
The code calculates MAP@12, which is a ranking metric that measures how well the algorithm ranks relevant items among its top 12 recommendations.
2. Precision@12
Precision@12 shows what percentage of recommended items were actually purchased by users, indicating how accurate the system is in predicting user preferences.
3. Recall@12
Recall@12 indicates what percentage of items a user actually purchased were captured in the recommendations, showing how well the system covers the range of a user's preferences.
4. F1@12
F1@12 combines precision and recall, providing a single metric that balances between recommending items users will like (precision) and covering all their potential interests (recall).

PERFORMANCE METRICS

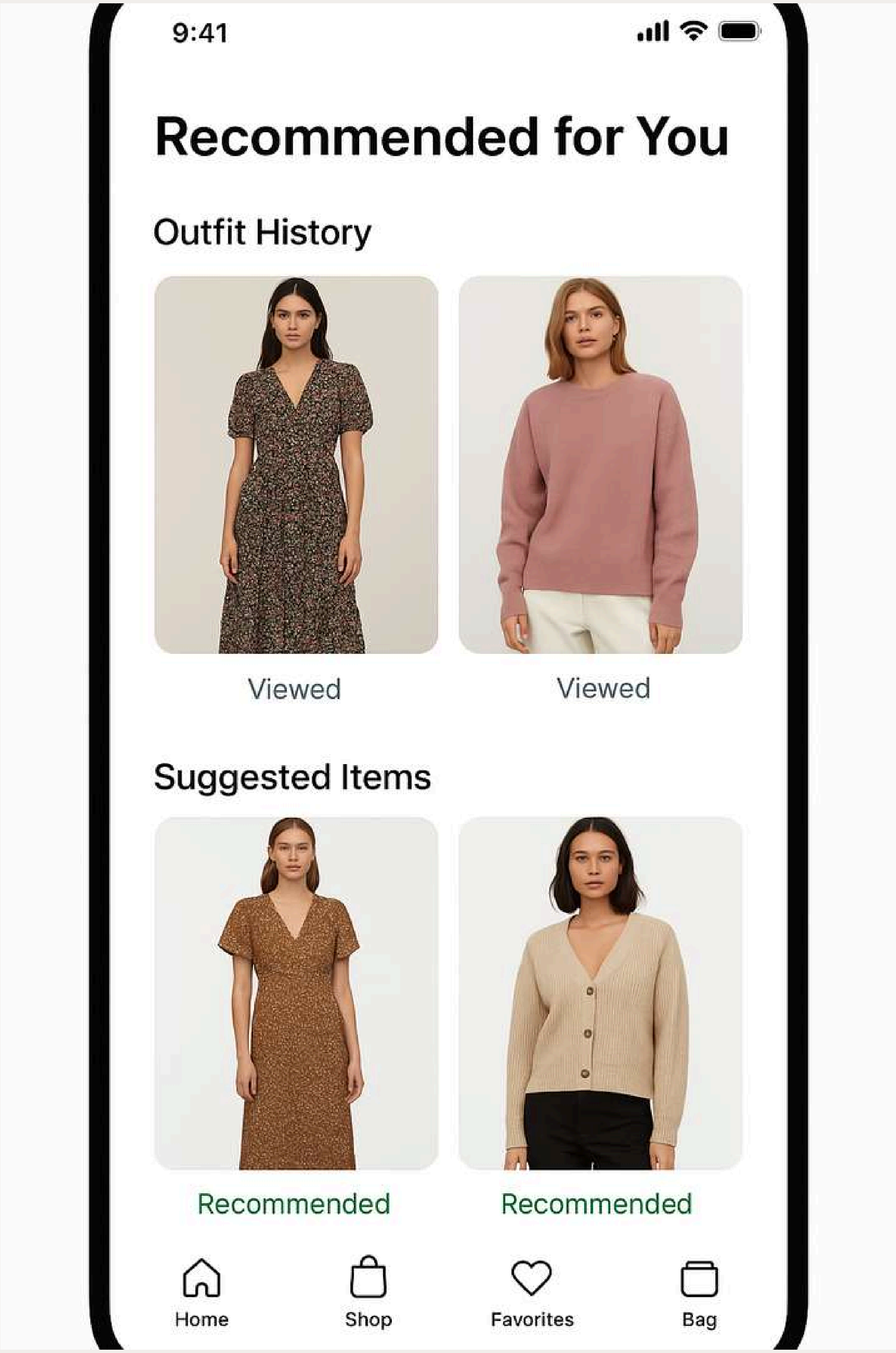
Before Budget Aware filtering



After Budget Aware filtering

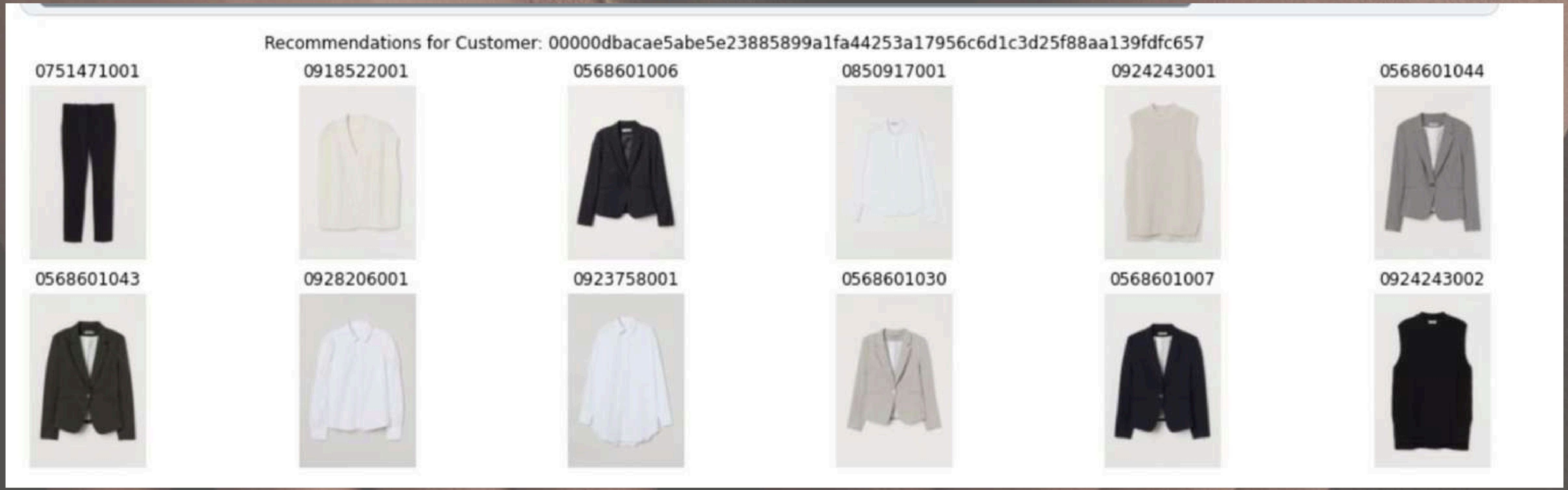


The **validation approach using the last 7 days of transaction data** is particularly strong, where we tried that it mimics real-world conditions where the system would be used to predict future purchases based on past behavior.

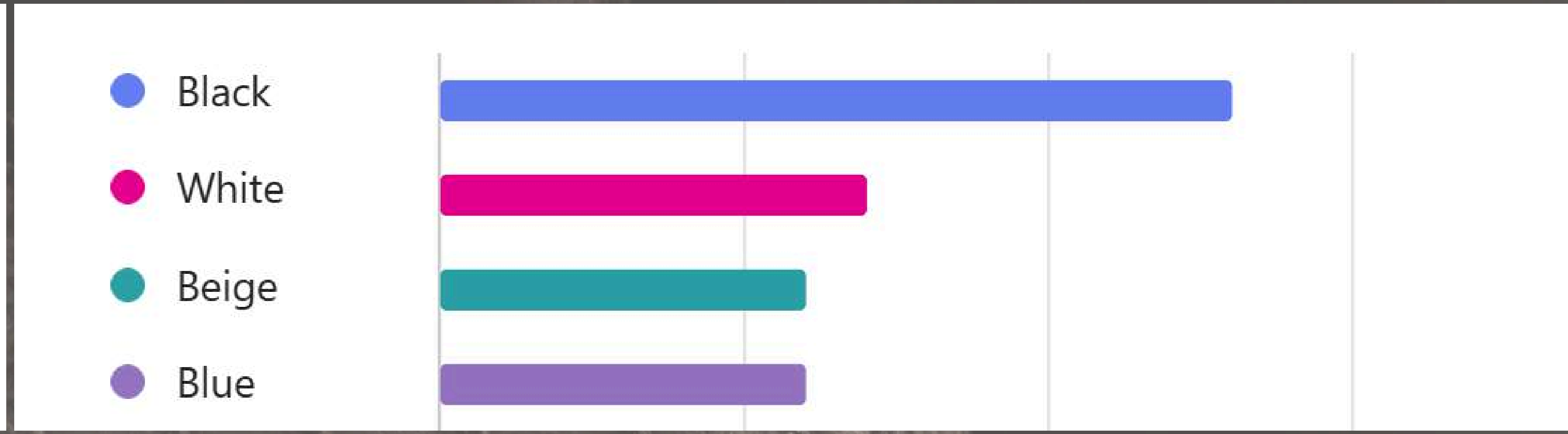


Example of how we validated through our survey

Recommendations they received as an output



AROUND 30% ANSWERED MIN. 1500 AS THEIR BUDGET
STYLE: FORMAL
TOP COLOUR PREFERENCES: BLACK, WHITE ,BEIGE
THEN THEY WERE SHOWED THE ABOVE OUTPUT AS THE RECOMMENDATION OUT OF WHICH APPROX 72% ALMOST FOUND THE RIGHT MATCH OR SOMEWHAT NEAR TO IT



CHALLENGES AND SOLUTIONS

1. Data Scale Challenges
Challenge: Processing 32 million transactions and 105,000 products efficiently.

2. Cold Start Problem:
Challenge: Making recommendations for new customers without purchase history.

3. Evaluation Challenges
Challenge: Measuring recommendation quality without explicit feedback.

CAN THE SOLUTION BE DEPLOYED AT PLAKSHA TO SOLVE THE PROBLEM YOU HAVE CHOSEN? IF SO, HOW?

It can't be implemented directly in Plaksha but there are some ways we can in future.

The system could be adapted to recommend academic resources (papers, books, videos) based on students' past engagement patterns.

This solution could feasibly be deployed at Plaksha for fashion recommendations if:

1. Plaksha has a fashion retail operation or partnership
2. There's sufficient transaction history data
3. Customer demographic information is available

OUR TEAM

Thank You!

Sara Goyal

Muskaan Sahni