UPEC
Connaissance - Action
UNIVERSITÉ
PARIS-EST CRÉTEIL
VAL DE MARNE

BIG DATA ANALYTICS

Kaggle Competition: H&M Personalized Fashion Recommendations

- Anu ENKHBOLD U32307570
- Benjamin MORELLE U32003506
- Pengyu XIONG U32318937
- Javzandulam BAYANMUNKH U32305952

Outline:

1. **Introduction**
   - Project overview
   - Why Segment-Based Recommendation
2. **Problem statement**
   - Introduction to dataset
   - Expected output
3. **Model Development**
   - Model description, how it works?
4. **Technical Details (Step-by-Step)**
5. **Results**
6. **Challenges encountered**

1. INTRODUCTION– PROJECT OVERVIEW

Challenge Background – Kaggle & H&M Group

H&M is a global fashion retailer with over 4,850 stores and 53 online markets. With such a huge product range, it's difficult for users to quickly find what they want.

Goal: Build a product recommendation system based on:
Customer purchase history
Customer & product metadata
Text & image data

Why this challenge matters?

- Improve customer experience.
- Reduce unnecessary returns → lower emissions → sustainability benefits.

1. INTRODUCTION– OUR METHOD

In this project, we apply a method proposed by Kaggle user Junji Takeshima, which focuses on customer segmentation for better recommendation accuracy.

Key idea: Since customer gender is not directly provided, we infer it from purchase behavior – for example, frequent purchases from the "Menswear" category suggest a male customer.

We segment customers using:

1.Age groups (e.g., 16–29 in 2-year bins, 30–59 in 5-year bins, 60+ as one group)

2.Sex attribute (inferred from dominant purchase category)

1. INTRODUCTION– WHY SEGMENT–BASED RECOMMENDATION WORKS

Goes beyond "you bought this, so buy that" logic
Focuses on group-level preferences:
“What people like in your group → Recommend based on that”

Example: A 25-year-old male frequently buys trendy items (e.g., Divided)
The system recommends:
Best-selling youth fashion (e.g., T-shirts, jeans)
Even if he hasn’t bought them before

Benefits
Stronger personalization
Improved customer satisfaction
Supports H&M’s sustainability goals by reducing returns

2. PROBLEM STATEMENT

Problematic: Personalized recommendations based on customer data.
Task: Develop a model to predict which articles each customer will purchase in the 7-day period immediately following the training data.
Input: Product details, Customer demographic data, purchase history of customers across time along with supporting metadata.

Files

Duplicate rows correspond to multiple purchases of the same item.

images/ - a folder of images corresponding to each article_id; images are placed in subfolders *(not all article_id values have a corresponding image)*

articles.csv - detailed metadata for each article_id available for purchase

customers.csv - metadata for each customer_id in dataset

sample_submission.csv - a sample submission file in the correct format

transactions_train.csv - the training data, consisting of the purchases each customer for each date, with additional information.

2. PROBLEM STATEMENT – DATA OVERVIEW

articles.csv

1. article_id
2. product_code
3. product_name
4. product_type_no
5. product_type_name
6. product_group_name
7. graphical_appearance
8. colour_group_code
9. colour_group_name
10. perceived_colour_value
11. perceived_colour_value_name
12. perceived_color_master_id
13. perceived_color_master_name
14. department_no
15. department_name
16. index_code
17. index_name
18. index_group_no
19. index_group_name
20. section_no
21. section_name
22. garment_group_no
23. garment_group_name
24. detail_desc

customers.csv

1. customer_id
2. #FN
3. #Active
4. club_membership_status
5. fashion_news_frequency
6. age
7. postal_code

transactions_train.csv

1. t_date
2. customer_id
3. article_id
4. price
5. sales_channel_id

How the Model Works

1. Age-Based Grouping
Age plays a significant role in shopping preferences. Customers are grouped into different age brackets based on their recorded age in the dataset.

The main strategy is:
Segmenting Customers into Age Groups:
0-18 (Teenagers)
19-30 (Young Adults)
31-50 (Middle-Aged Adults)
51+ (Seniors)
Each age group is associated with popular product categories preferred by customers within that demographic, derived from historical purchase data.
Personalized recommendations are generated by identifying trending fashion items purchased by other users in the same age segment.

How the Model Works

2. Sex/Attribute-Based Grouping

Unlike age, customer sex information is not directly available. Instead, we infer sex or attribute classification based on past purchases. The steps involved are:

Identifying Gendered Product Preferences:

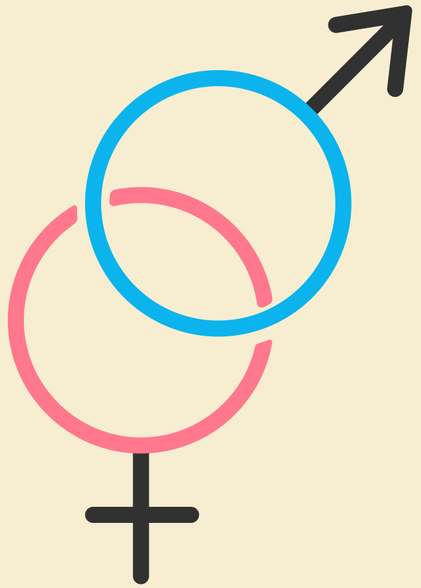
Assign each product a gender/age attribute based on metadata (e.g., "Women's Clothing," "Men's Shoes," "Kids' Apparel").

Inferring Customer Sex/Attribute:

Analyze each customer’s historical purchase data to estimate their most likely gender and attribute category.

Customers who predominantly purchase "Women's" items are classified under the female segment, and similarly for the male segment.

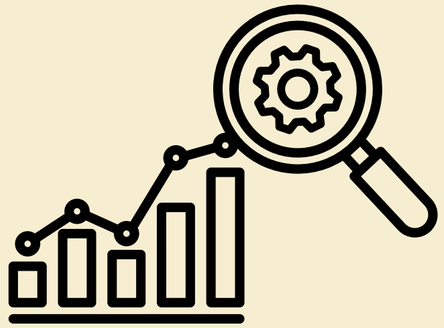
If a customer frequently buys kids' items, they are classified as having children, influencing their recommendations.



Performance Evaluation & Iterative Improvements

Initial Public Score: The initial model achieved a score of 0.0074.
Model Adjustment: Updating transaction history usage from "after 9/1" to "after 9/15" improved the score to 0.0078.
Future Enhancements: Incorporating additional features such as purchase frequency, product descriptions (NLP techniques), and image processing to refine recommendations further.

This approach leverages customer segmentation by age and inferred sex/attribute to improve the accuracy of product recommendations. By iterating and refining the model based on transaction data, H&M can provide more relevant recommendations, leading to higher engagement and better customer satisfaction.



4. TECHNICAL DETAILS

1. Dataset Overview & Implementation Process

The author executed a total of 42 code cells, covering both input and output operations

The transaction history spans 2 years from September 2018 to September 2020.
The customer dataset lacks a direct gender or attribute column, requiring inference.
Age is a critical factor in our approach

```
transactions_df["t_dat"] = pd.to_datetime(transactions_df['t_dat'])
transactions_df
```

	t_dat	customer_id	article_id	price	sales_channel_id
0	2018-09-20	000058a12d5b43e67d225668fa1f8d618c13dc232df0ca...	663713001	0.050831	2
1	2018-09-20	000058a12d5b43e67d225668fa1f8d618c13dc232df0ca...	541518023	0.030492	2
2	2018-09-20	00007d2de826758b65a93dd24ce629ed66842531df6699...	505221004	0.015237	2
3	2018-09-20	00007d2de826758b65a93dd24ce629ed66842531df6699...	685687003	0.016932	2
4	2018-09-20	00007d2de826758b65a93dd24ce629ed66842531df6699...	685687004	0.016932	2
...	...	...	...	...	...
31788319	2020-09-22	fff2282977442e327b45d8c89afde25617d00124d0f999...	929511001	0.059305	2
31788320	2020-09-22	fff2282977442e327b45d8c89afde25617d00124d0f999...	891322004	0.042356	2
31788321	2020-09-22	fff380805474b287b05cb2a7507b9a013482f7dd0bce0e...	918325001	0.043203	1
31788322	2020-09-22	fff4d3a8b1f3b60af93e78c30a7cb4cf75edaf2590d3e5...	833459002	0.006763	1
31788323	2020-09-22	fffef3b6b73545df065b521e19f64bf6fe93bfd450ab20...	898573003	0.033881	2

31788324 rows × 5 columns

2. Customer Segmentation Approach

A. Grouping Customers by Age

16–99 age will be grouped into age_id 0–13

```
age_group
```

	age	age_id
0	16	0
1	17	0
2	18	1
3	19	1
4	20	2
...	...	...
79	95	13
80	96	13
81	97	13
82	98	13
83	99	13

84 rows × 2 columns

B. Grouping Customers by Purchase History

Based on index-group_name in article list, purchase history is classified as women's item, men's item, kid's item ;
Based on purchase history, we classified the customers into 5 attributes guessing those who purchase more women's item are women and those who purchase more men's item are men

```
[15] print(articles_df["index_group_name"].unique())
     print(articles_df["index_group_no"].unique())

['Ladieswear' 'Baby/Children' 'Menswear' 'Sport' 'Divided']
[ 1  4  3 26  2]

[16] sex_category = articles_df[["index_group_no", "index_group_name"]].reset_index()
     display(sex_category["index_group_name"].value_counts())
```

index_group_name	count
Ladieswear	39737
Baby/Children	34711
Divided	15149
Menswear	12553
Sport	3392

dtype: int64

3. Assigning Attributes to Customers

- Customers are classified based on which category they purchase the most from:
- Each customer's purchase history is classified into five categories (Ladieswear, Menswear, Children, Divided and Sport.
- The category with the highest purchase count determines the customer's primary attribute.
- Handling Ties:
- If a customer buys equal amounts from multiple categories, they are assigned multiple attributes.
  Since Women's items dominate, customers with multiple attributes default to "Woman".

We identify the category with the highest purchase count for each customer. "The lambda function" finds the max value and assigns the corresponding category to a new column called "attribute".

```
[16] sex_category = articles_df[["index_group_no", "index_group_name"]].reset_index()
     display(sex_category["index_group_name"].value_counts())
```

index_group_name	count
Ladieswear	39737
Baby/Children	34711
Divided	15149
Menswear	12553
Sport	3392

dtype: int64

```
[18] import matplotlib.pyplot as plt

     sex_category_list = sex_category["index_group_name"].value_counts().index.to_list()
     plt.figure(figsize=(5, 5))
     plt.rcParams["font.size"] = 12
     plt.pie(sex_category["index_group_name"].value_counts().sort_values(ascending=False),
             labels = sex_category_list, startangle = 90, counterclock=False, autopct="%1.1f%%")
     plt.show() # Added parentheses to call the show function
```

```
[23] %%time
     cust_sex["attribute"] = cust_sex.apply(lambda x : list(x[x == x.max()].index), axis=1)
     cust_sex

CPU times: user 5min 9s, sys: 14 s, total: 5min 23s
Wall time: 5min 19s
```

4. Prediction Approach

The goal is to predict one-week transactions after the training period.
Transactions from September 2020 serve as the target data.
We calculate purchase counts per age group and attribute.

After prediction, we identify the most sold items in September 2020.
We sum up purchases by age_id and attribute, sorting by sales count.
This provides insights into which items are popular for each demographic.

```
[47] recommend_sex_age["article_id"] = recommend_sex_age["article_id"].str.strip()
     recommend_sex_age["article_id"] = recommend_sex_age["article_id"].str[:131]
     recommend_sex_age = recommend_sex_age.drop(["len"], axis =1)

[49] recommend_sex_age.loc[(recommend_sex_age["age_id"]==7), ]
```

	age_id	attribute	article_id
34	7	Have-kids	852748005 852748006 881133001 828812001 846181...
35	7	Man	685816001 763275002 598755001 598755003 598755...
36	7	Sports-person	866731001 791587001 805000001 902507001 852584...
37	7	Woman	924243001 909370001 906352001 923758001 865799...
38	7	Young	935541001 706016003 788575004 772902001 711053...

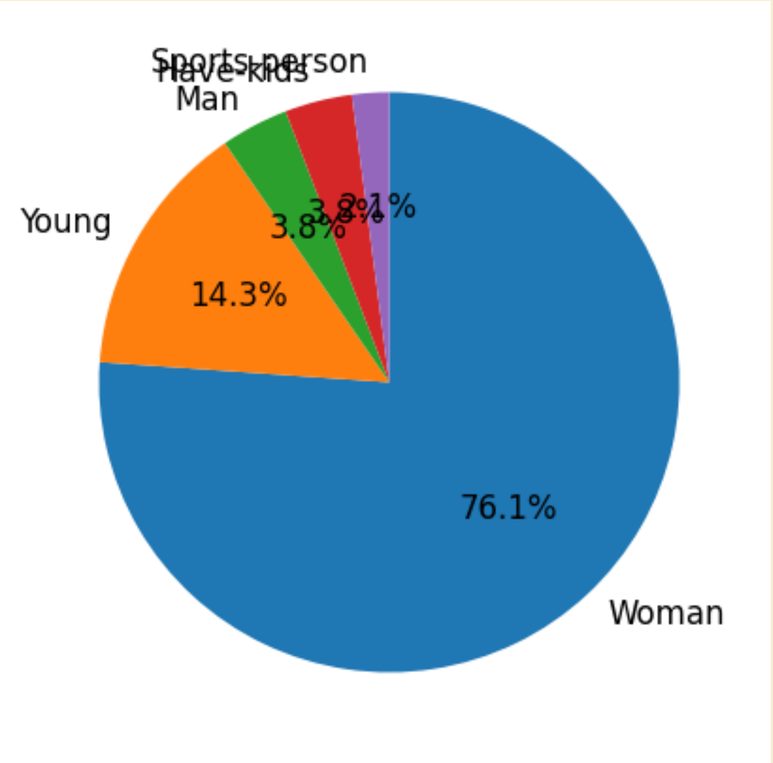
Recommendation for Early 30s Women
Based on analysis, the model suggests items frequently purchased by women aged 30–34.

```
In [41]: show_images(recommend_sex_age.iloc[38, 2].split(), 6)
```

5. Visualize Results
Customer Segmentation by Age and Gender (Pie Charts)

Show the distribution of customers by inferred gender (e.g., Women, Men, Young, Parents, Sports-person).

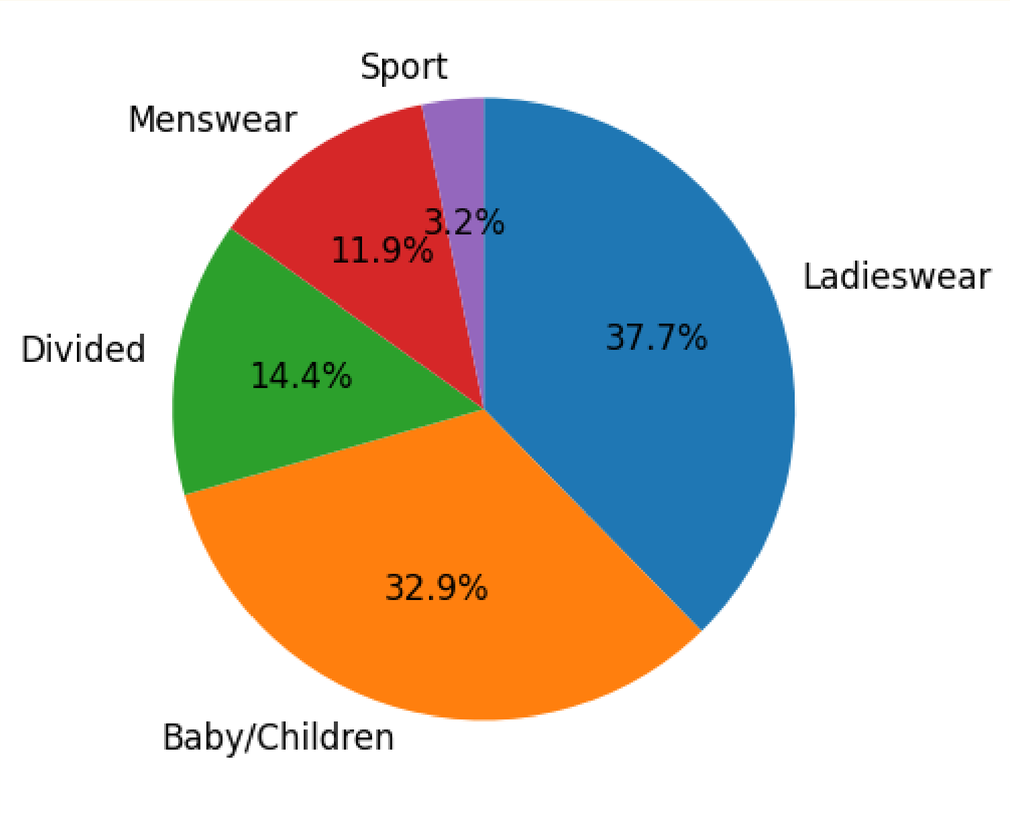
- **76.1% of customers are classified as "Woman", indicating a strong female customer base.**
- **14.3% are classified as "Young", showing that young customers make up a significant portion of the clientele.**
- **Other segments ("Man," "Have-kids," "Sports-person") are much less represented, which may indicate a data bias or a real market trend.**



Purchase Behavior by Customer Segment (Bar Chart)

Show the most purchased product categories by customers.

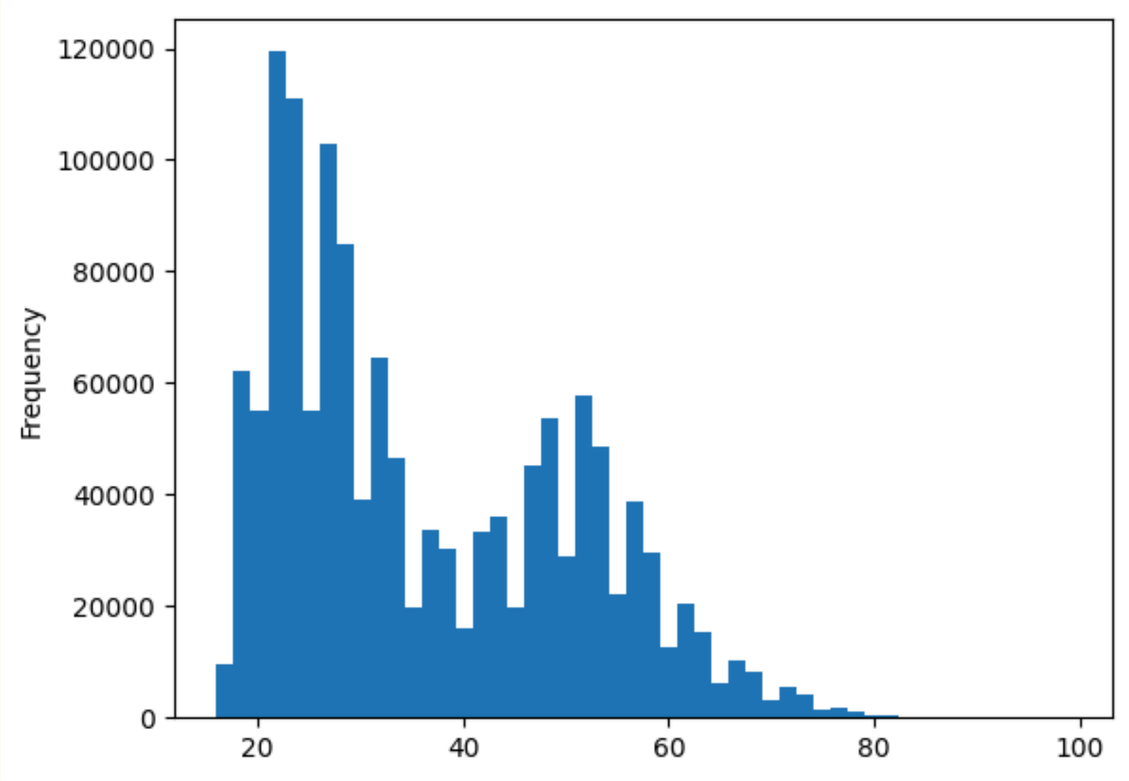
- **37.7% of products belong to the "Ladieswear" category, aligning with the predominance of female customers.**
- **32.9% of sales come from the "Baby/Children" category, indicating that many customers also purchase for children.**
- **The "Menswear" and "Sport" segments are smaller, suggesting that the male and sports-oriented customer base is less prominent.**



Age Distribution of Customers (Histogram)

Display the frequency of customers in different age groups to illustrate the most active demographics.

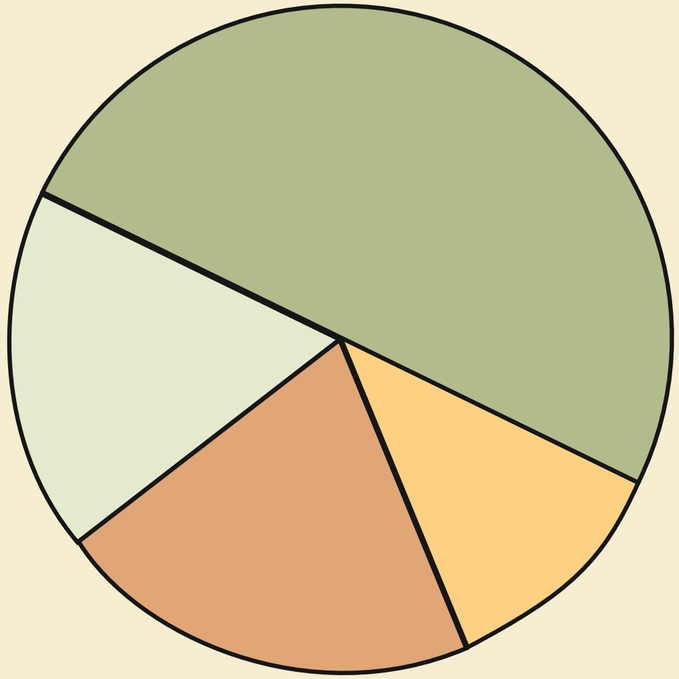
- **Most customers are between 20 and 40 years old, corresponding to young and middle-aged adults.**
- **The highest peaks are around ages 20 to 30, showing that this age group is the most active in shopping.**
- **A gradual decline in purchases is observed after age 50, suggesting lower engagement from older customers in online shopping.**



Impact of Segmentation on Recommendations

category by targeting sporty customers more effectively.

With these observations, we can improve recommendations:
For women (~76% of customers), recommendations should prioritize "Ladieswear" and "Baby/Children" items.
For young customers (~14%), recommendations can include more trendy or youth-oriented fashion items.
Customers purchasing "Baby/Children" products can be classified as parents ("Have-kids"), optimizing recommendations for this segment.
Adding data on sports preferences could help refine recommendations for the "Sports"



6. CHALLENGES

1. Deprecated append() Method in Pandas

Issue:

Original code used DataFrame.append(), which is deprecated and removed in pandas 2.0.

Effect:

Code fails to run on platforms like Google Colab using newer pandas versions.

Fix:

Replaced append() with pd.concat([...], ignore_index=True) to ensure compatibility.

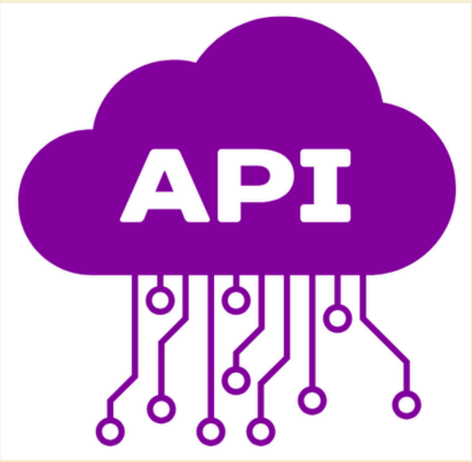
```
age_id = 0
age_group = pd.DataFrame(columns=["age", "age_id"])
age = 16

for i in range(53):
    if age < 30:
        temp_group = pd.DataFrame({"age": [age, age + 1], "age_id": [age_id, age_id]})
        # Use pd.concat instead of append
        age_group = pd.concat([age_group, temp_group], ignore_index=True)
        age += 2
        age_id += 1
    elif age < 60:
        temp_group = pd.DataFrame({"age": [age, age + 1, age + 2, age + 3, age + 4], "age_id": [age_id, age_id, age_id, age_id, age_id]})
        # Use pd.concat instead of append
        age_group = pd.concat([age_group, temp_group], ignore_index=True)
        age += 5
        age_id += 1
    else:
        temp_group = pd.DataFrame({"age": [age], "age_id": [age_id]})
        # Use pd.concat instead of append
        age_group = pd.concat([age_group, temp_group], ignore_index=True)
        age += 1
```

6. CHALLENGES

2. Kaggle API Download Failure (Challenge Closed)

Issue:

We attempted to use the Kaggle API to download the dataset and input the code below:



submission = pd.read_csv('../input/h-and-m-personalized-fashion-recommendations/sample_submission.csv')

Problem:

Since the Kaggle competition has ended, the API no longer supports downloading the full dataset. As a result, we encountered authentication errors and missing file issues.

Solution:

To resolve this, we manually downloaded the dataset from Kaggle and uploaded it to our Google Drive. We then accessed it via Google Colab using drive.mount() and specified the file paths accordingly.

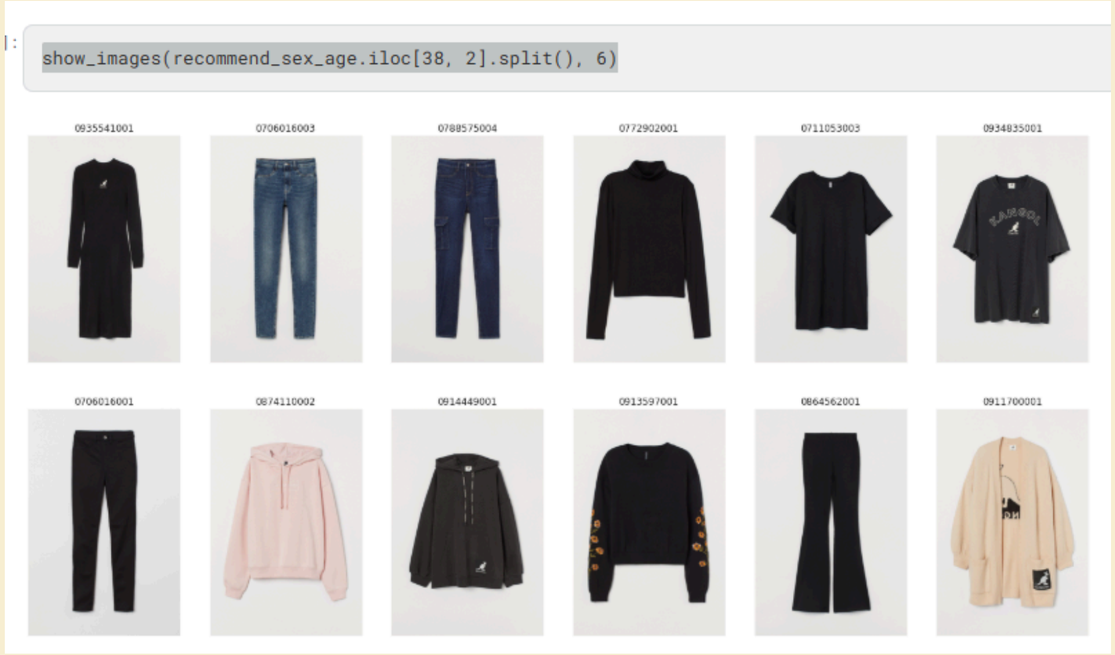
6. CHALLENGES

3. Image Display Limitations (30GB File Size)

Issue:
Dataset includes 30GB of image data, which exceeds Colab’s memory/storage limit.

Effect:
Functions like show_images() cause errors or runtime crashes.

Fix:
Skipped image-based output. Focused analysis on structured tabular data



THANK YOU