H&M Outfit

Recommendation System

Project By :-

Supraja R (23MSD7015)

Divya Bhardwaj (23MSD7020)

Pramod Choudhary (23MSD7030)

Aditya Antony Thomas (23MSD7063)

Supervised By

Dr .Manimaran A.

Department of Mathematics,

School of Advanced Sciences

VIT-AP UNIVERSITY

Abstract

The H&M Recommendation Model uses past transaction data to forecast and suggest future purchases, improving the shopping experience and increasing sales. The model examines prior activities and purchasing habits of H&M store patrons using sophisticated data mining techniques. The Frequent Pattern (FP) Growth algorithm is used in the core technique to find frequent purchase patterns and relationships in the transaction data. Through comprehension of these patterns, the model is able to provide tailored recommendations that correspond with specific client preferences, increasing customer happiness and decreasing return rates. This method helps H&M optimise its inventory control and marketing initiatives in addition to offering personalised purchasing recommendations.

Table of Contents

1. Introduction

Background

In this day and age, use of a product recommendation system is very important for companies to achieve various aspects of increasing their business such as Sales, Customer Satisfaction and Customer Loyalty to name a few. The fashion industry is undergoing a huge upheaval in today's fast-paced digital environment, which is being driven by technological breakthroughs and changing customer behaviours. Consumer expectations have escalated due to the growing trend of personalization across multiple industries. They now want experiences that are tailored to their own interests and inclinations. This transformation is most noticeable in the fashion sector, where e-commerce and artificial intelligence (AI) are combining to transform how consumers find and select clothing, which gives rise to Outfit recommendation systems, which provide personalised outfit recommendations based on user preferences, historical behaviour, and current fashion trends, are becoming a potent tool for improving the shopping experience.

Problem Statement

When implementing solutions like these it is important to note that a problem can be solved from multiple perspective and the correct solutions can be subjective to the requirement of the company or a user who wishes to implement such a system for their business growth,
The ways this can be achieved is summarised below.

Personalization Gap

Finding clothing that precisely fits their individual style, size, and occasion requirements can be difficult for shoppers. The enormous number of alternatives available can make shopping a time-consuming and frustrating affair. Using advanced recommendation algorithms that can examine each customer's unique interests, past purchases, and browsing habits to make highly relevant product recommendations is one way to close the customization gap. This improves the buying experience and raises the possibility of client loyalty and purchase happiness.

User Experience

Because so many e-commerce sites lack interactive and engaging components, it is challenging for customers to envision and decide what to buy. Adding features like augmented reality (AR) fitting rooms, virtual try-ons, and dynamic product displays can improve the user experience. By enabling buyers to see how items will appear on them or in their environment, these tools increase confidence in buying decisions and lower the possibility of returns.

Making sure that advice is current with the newest styles and is both individualised.

Retailer insights

In order to manage inventory and marketing tactics, retailers require sophisticated systems for tracking and analysing consumer behavior and preferences. Retailers can gain extensive insights into customer demographics, product performance, and purchase trends by implementing business intelligence and advanced analytics solutions. Making educated judgments on stock levels, promotions, and focused marketing efforts is dependent on this data, which will ultimately result in more productive operations and higher sales.

High Return Rates

In the absence of tailored suggestions, buyers frequently purchase goods that fall short of their expectations, leading to a high percentage of returns. Improving the precision of product suggestions and the readability of product information are two strategies that must be combined to lower return rates. Retailers may drastically lower the possibility of returns by making sure that the products recommended are extremely relevant to the customer's needs and by offering thorough, accurate product descriptions and reviews. Furthermore, adding user reviews to the recommendation engine can improve the precision and applicability of product recommendations even more.

Summary

Using Association Rule Algorithms and other cutting-edge methods to develop a strong product recommendation system takes care of several aspects of consumer satisfaction and operational effectiveness. These technologies help create a more engaging, effective, and profitable retail environment by bridging the customization gap, improving user experience, remaining up to date with fashion trends, and lowering high return rates.

Objectives

- To create a system that uses machine learning to select outfits based on user data, such as browsing history, past purchases, and style preferences.
- Provide a user-friendly interface that makes it simpler for people to find and buy suggested products, therefore improving their online shopping experience.
- Incorporate trend analysis functions to guarantee that suggestions are stylish and up to date.
- Give merchants useful information gleaned from user interaction data to enhance marketing and inventory control.
- Enhance the precision and pertinence of product suggestions to lower return rates.

Scope

This project's scope includes creating, putting into use, and assessing an AI-powered outfit recommendation system that aims to improve and customise online buying. To make sure the system satisfies the demands of both customers and retailers, the project will include a number of crucial stages and elements. The main areas of attention are listed below:

- Recommendation Engine Development
- Retailer Insights and Analytics

2. Literature Review

A key data mining technique is market basket analysis, which looks for relationships between items in big databases of customer transactions. Because association rule algorithms like FP-Growth and Apriori are effective at processing transactional data, they have historically been widely used in this field. However, the usage of deep learning models and other sophisticated algorithms, such as Gradient Boosted Trees (GBM), in a variety of predictive analytics jobs has expanded as a result of the recent explosion in computing power. In order to demonstrate the feasibility of Apriori and FP-Growth for market basket analysis in resource-constrained contexts, this paper evaluates the effectiveness, scalability, and applicability of association rule algorithms against these more resource-intensive models.

Algorithms for Association Rule

Apriori Algorithm

Agrawal et al. (1994) created the Apriori algorithm, a groundbreaking technique for mining frequent itemsets and determining association rules. It works on the tenet that every non-empty subset of an itemset that occurs frequently needs to likewise occur frequently. Candidate itemsets are created, their occurrences are counted, and those that don't reach a minimal support criterion are pruned. Apriori can be computationally expensive because it requires several scans of the transaction database, despite its simplicity and ease of use.

The FP-Growth Algorithm

The FP-Growth algorithm was suggested by Han et al. (2000) in order to overcome the drawbacks of Apriori. FP-Growth uses an FP-Tree to create a compressed representation of the transaction database. It then takes frequent itemsets straight out of the tree, eliminating the need to create candidate itemsets. This technique is more scalable for huge datasets since it drastically lowers the amount of database scans and processing overhead.

Comparative Study using GBM and Deep Learning Requirements for Resources

In a variety of predicting tasks, deep learning models—especially neural networks and GBM algorithms like LightGBM (Ke et al., 2017)—have shown improved performance. These models, however, are resource-intensive and frequently call for high amounts of memory and processing power. It may be necessary to use powerful GPUs and a lot of RAM—up to 300GB of GPU instances—when training deep learning models. Because of this, they are less practical in settings with constrained processing power.

As an illustration, association rule algorithms such as FP-Growth and Apriori are comparatively light-weight. They can function well with normal computing infrastructure and don't require specialised hardware. This makes them especially appropriate for applications or small- to medium-sized businesses where resource limitations are a major factor.

Application and Interpretability

Interpretability is a major benefit of association rule algorithms. Clear insights into the links between things are provided by the simple and easy-to-understand rules produced by Apriori and FP-Growth. A rule that says, for example, "If a customer buys bread and butter, they are likely to buy milk" is easy to understand and useful when making decisions for a firm.

On the other hand, models produced by GBM and neural networks are frequently referred to as "black boxes." Even while they might have great prediction accuracy, their inability to be easily interpreted can be problematic in situations when it's essential to comprehend the decision-making process.

Performance and Scalability

Because FP-Growth makes effective use of the FP-Tree structure, it has been demonstrated to perform better in terms of scalability than Apriori. Research like Borgelt (2005) has shown that FP-Growth is a better option for comprehensive market basket research since it can manage larger datasets more skillfully. However, although GBM and neural networks can grow with more data, their training times and the difficulty of fine-tuning hyperparameters can be prohibitive.

Applications in Practice for Market Basket Analysis: Because association rule algorithms are practical and efficient, they are frequently used in market basket analysis. They support companies in determining often co-purchased goods and in maximising inventory control, product placement, and marketing tactics. These algorithms offer a trade-off between computational viability and speed in resource-constrained contexts, guaranteeing that useful insights can be obtained without hefty hardware investments.

Although GBM and deep learning models perform well in predictive analytics, their resource needs may be prohibitive. Apriori and FP-Growth, two association rule algorithms, offer a good substitute for market basket analysis, particularly in situations when computational resources are scarce. Because of their effectiveness, interpretability, and scalability, they are perfect for gleaning insightful information from transactional data, which helps retail and other industries make smart decisions.

We noticed that a lot of the winning entries in the competition on Kaggle made extensive use of Deep Learning models, especially Neural Networks. Some of these solutions use up to 300GB of GPU instances, and they frequently call for strong GPUs and large amounts of RAM. Furthermore, machine learning models like LightGBM (LGBM) are often used.
We made the decision to
change the course of our project due to the limitations we are facing,
especially with regard to memory and processing power. We will concentrate on
Association Rule Algorithms, notably Apriori and FP-Growth, rather than
resource-intensive models.
When it comes to market basket analysis, association rule algorithms work effectively because the objective is to find product groupings that commonly appear together in

transactions. When compared to Gradient Boosted Trees and Neural Networks, these algorithms are substantially lighter.

Large datasets can be analysed by them effectively in order to find trends and connections between different items. Because of this, they are perfect for suggesting things that are regularly purchased in addition to those that are currently in the user's cart.

Our goal is to develop a strong recommendation system that can provide pertinent ideas without requiring a lot of processing power by utilising Apriori and FP-Growth. With this strategy, we can make better use of the resources at our disposal while still offering people insightful analysis and suggestions based on their purchasing habits.

3. Methodology

Research Design

We decided to use the FP Growth algorithm to understand what the customers tend to buy together and recommend the items which are generally bought together and recommend them to the customer when they are shopping/searching for products.

Data Sources

We procured the dataset from a competition which was held in Kaggle.

Tools and Techniques

We took advantage of python libraries such as matplotlib, seaborn, mlxtend.preprocessing, mlxtend.frequent_patterns, apriori, association_rules, fp growth, sklearn.neighbors, NearestNeighbors.

4. Data Collection and Preparation

The Data was obtained from the Kaggle Competition which was titled “H&M Personalized Fashion Recommendations”.

Data Cleaning

The data provided to the contestants in the competition was clean and required very minimal preparation to feed into the model, other than converting the features into their correct data types.

Data Exploration

A number of significant insights regarding the interests and purchase patterns of our customers have been obtained from our thorough examination of the three-year dataset. The year 2019 had the highest volume of purchases, with a noticeable concentration of transactions in the two months preceding summer and in the summer months of June and July. This suggests a seasonal uptick in consumer activity, probably brought on by the nature of summer clothing and seasonal advertising.

According to the statistics, the most popular goods are dresses, sweaters, and pants, which highlights their significance in customer wardrobes. In line with this, the product categories associated with clothing, namely Garment Upper Body, Garment Lower Body, and Garment Full Body, were the most favoured by consumers. These categories are broad, indicating the variety of clothing items that fall within them.

Customers' preferred colours were also determined; the most popular shades were pink, white, dark blue, and black. On the other hand, the least popular hues were orange, silver, and green, indicating that these could not suit consumer preferences or current fashion trends. Melange designs and solid, all-over patterns were the most common patterns.

Furthermore, the most popular tones were light, dusty light, and dark, suggesting a propensity for modest and adaptable colour schemes.

The most popular departments in the store, according to an analysis of department performance, are the ones that sell jerseys, knitwear, and trousers, while the least popular departments are small accessories and kid girl dresses. This discrepancy could help marketing and inventory initiatives better suit the needs of the target audience.

Gender-specific buying trends revealed a preference for ladieswear, which was followed by menswear and the "Divided" category, which includes fashionable clothing that is popular among young people. This demonstrates how much demand there is for clothing for women and young people.

The majority of clients are members of the store's club, according to membership and subscription data, which might be used for loyalty programs and targeted advertising. Nonetheless, nearly two-thirds of clients do not receive the fashion news stream, suggesting a possible avenue to boost interaction and correspondence.

A youthful client base is shown by the fact that over half of the clients are under 40. October 2019 saw the largest transaction volume in the dataset, with major sales peaks occurring in

April, May, October, and December. These patterns point to possible times for focused advertising and stock management during these busyA number of significant insights regarding the interests and purchase patterns of our customers have been obtained from our thorough examination of the three-year dataset. The year 2019 had the highest volume of purchases, with a noticeable concentration of transactions in the two months preceding summer and in the summer months of June and July. This suggests a seasonal uptick in consumer activity, probably brought on by the nature of summer clothing and seasonal advertising.

According to the statistics, the most popular goods are dresses, sweaters, and pants, which highlights their significance in customer wardrobes. In line with this, the product categories associated with clothing, namely Garment Upper Body, Garment Lower Body, and Garment Full Body, were the most favoured by consumers. These categories are broad, indicating the variety of clothing items that fall within them.

Customers' preferred colours were also determined; the most popular shades were pink, white, dark blue, and black. On the other hand, the least popular hues were orange, silver, and green, indicating that these could not suit consumer preferences or current fashion trends. Melange designs and solid, all-over patterns were the most common patterns.

Additionally, the preferred shades tended to be dark, dusty light, and light, indicating a tendency towards versatile and subtle colour palettes.

The most popular departments in the store, according to an analysis of department performance, are the ones that sell jerseys, knitwear, and trousers, while the least popular departments are small accessories and kid girl dresses. This discrepancy could help marketing and inventory initiatives better suit the needs of the target audience.

Gender-specific buying trends revealed a preference for ladieswear, which was followed by menswear and the "Divided" category, which includes fashionable clothing that is popular among young people. This demonstrates how much demand there is for clothing for women and young people.

The majority of clients are members of the store's club, according to membership and subscription data, which might be used for loyalty programs and targeted advertising. However, about two-thirds of customers are not subscribed to the fashion news feed, indicating a potential area for increasing engagement and communication.

A youthful client base is shown by the fact that over half of the clients are under 40. October 2019 saw the largest transaction volume in the dataset, with major sales peaks occurring in April, May, October, and December. These patterns point to possible times during these spikes in activity for focused promotions and inventory management.

Ultimately, the investigation verified that the most popular sales channel is an online store, which is consistent with the expanding e-commerce trend. These results provide strong support for the choice to use Association Rule Algorithms in market basket analysis. With the goal of increasing customer happiness and increasing sales, this strategy will make use of our data resources to improve the product recommendation system by offering customised product recommendations based on recognized purchase trends.

5.Analysis and Results

Analysis

The H&M Recommendation Model uses sophisticated algorithms and efficient preprocessing of transaction data to provide individualised item recommendations and important insights into consumer purchasing behavior. The model predicts products that consumers are likely to purchase based on the contents of their current cart and detects frequently occurring itemsets by using the FP-Growth algorithm for pattern recognition and association rule mining. By providing tailored and pertinent advice, this strategy not only increases customer satisfaction but also boosts revenues and improves inventory control. The implementation's success shows that there is room for further development, including honing the recommendation system, assessing its effectiveness, and taking user input into account. greater sophisticated models like Decision Trees and Neural Networks could be investigated to further improve the performance with extra resources like greater RAM and GPU capacity.

```
def recommend_items_based_on_basket(basket, frequent_itemsets):
    filtered_itemsets = frequent_itemsets[frequent_itemsets['itemsets'].apply(lambda x: all(item in x for item in basket))]
    recommendations = [item for itemset in filtered_itemsets['itemsets'] for item in itemset if item not in basket]

    return recommendations[:2]

# Example basket
basket =['721270016', '831467001']
total=[]
for i in basket:
  l=[i]
  recommendations = recommend_items_based_on_basket(l, frequent_itemsets)
  for i in recommendations:
    if i not in total:
      total.append(i)
print("Top Recommendations based on Basket:", total)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transfo
  and should_run_async(code)
Top Recommendations based on Basket: ['845790002', '832361002', '841185002', '736530007']
```

We have provided a basket which indicate the item present in the customers cart

Which then recommends the items which the user may want to buy.

These items were then selected to recommend to the user.

Results Summary

Data Preprocessing: Transformed and filtered transaction data effectively.

Grouping and Filtering: Focused on customers with richer transaction histories.

Association Rule Mining: Prepared data and identified meaningful frequent itemsets.

Recommendations Generation: Generated relevant recommendations.

Visualisation: Provided a visual comparison of basket and recommended articles.

Interpretation

We can see that the user has a Grey Coat in their basket. Consequently our model suggested 2 items which are related to the first item in the basket.

This can help Users find products much more quickly, saving them time and at the same time allows the seller to get more sales.

6.Discussion

Insights

The project's goal is to create a machine learning-based system that makes personalised outfit recommendations for each user based on their browsing history, previous purchases, and preferred styles. The system will have an easy-to-use interface that makes finding and buying products simple, improving the online shopping experience. Through collaborative filtering, content-based filtering, and hybrid techniques, the system can produce customised outfit suggestions by monitoring and evaluating user behaviour, including products browsed and past purchases. Furthermore, trend analysis features will be integrated to guarantee fashionable and current recommendations by using information from fashion blogs, social media, and sales trends.

Featuring filters for size, colour, and occasion in addition to a virtual try-on option to increase user engagement, the interface will be visually appealing and easy to use. Merchants can gain useful insights to optimise their marketing efforts and manage inventory more effectively by analysing user interaction data, such as clicks and time spent on recommendations. This will help ensure that popular items are available and reduce overstock of less popular products. using the goal of lowering return rates by increasing the accuracy and relevancy of recommended products, the system will continuously enhance recommendation accuracy by fine-tuning machine learning models using fresh data and feedback. Strong security protocols will be put in place to safeguard user information and adhere to privacy laws, guaranteeing a safe and reliable online purchasing experience.

The project's ultimate goal is to raise conversion rates and enhance customer happiness by providing tailored, fashionable, and pertinent attire recommendations. Merchant profitability can be increased by accurate recommendations, which can result in more sales and a decrease in return rates.

Comparison

Because association rule algorithms, like FP-Growth and Apriori, can find associations between products in big datasets and discover frequently occurring itemsets, they are very useful for market basket analysis. In order to create association rules based on frequently occurring itemsets that satisfy a confidence level, the Apriori algorithm first generates candidate itemsets and then prunes those that do not achieve a minimal support criterion. Even though Apriori's pruning algorithms necessitate several dataset scans, they aid in reducing computational complexity. However, by building an FP-tree and compressing the dataset into a tree structure, FP-Growth increases efficiency. This makes the method faster and more scalable than Apriori by enabling it to mine frequently occurring patterns without requiring several scans.

For finding product groups in transactional data, Apriori and FP-Growth stand out for their ease of use and effectiveness when compared to more resource-intensive models like Gradient Boosted Trees and Neural Networks. Gradient Boosted Trees and Neural Networks are less appropriate for jobs where the goal is to find simple, regular co-occurrences of objects rather than complicated patterns because they demand a significant amount of computing resources and prolonged training. These models can be overkill for simple association rule mining, but they perform exceptionally well in situations where high-dimensional data handling and forecast accuracy are critical.

When it comes to market basket analysis, FP-Growth is superior to Convolutional Neural Networks (CNNs). CNNs are made to handle grid-like input, such photographs, and use convolutions to record spatial hierarchy. CNNs are not naturally suited for frequent itemset mining and association rule discovery, despite the fact that they can be modified for sequence data and excel in challenging pattern recognition tasks. Converting transactional data into an appropriate format and training the network are necessary steps in the implementation of CNNs for market basket analysis. These processes are computationally demanding and could not produce appreciable gains over simpler models.

However, FP-Growth is designed especially for frequent pattern mining. Because it builds an FP-tree, the dataset may be represented compactly, facilitating the efficient mining of recurring patterns without requiring repeated dataset scans. This improves FP-Growth's speed and scalability while also improving its interpretability for the particular purpose of market basket analysis. Market basket analysis tries to provide precise and practical insights into customer purchasing behaviour; FP-Growth achieves this by focusing on the linkages and co-occurrences of items.

Implications

The results of applying the FP-Growth and Apriori algorithms in data science demonstrate their high practical utility and efficiency in a wide range of applications. These algorithms are particularly good at finding common itemsets and product correlations in huge transactional datasets. In order to reduce computational complexity, Apriori generates and prunes candidate itemsets. FP-Growth further improves efficiency by compressing data using an FP-tree structure, which enables fast and scalable pattern mining without the need for multiple scans. Because of these features, they are perfect for market basket research in retail settings, where they can aid with inventory optimisation, strategic planning for promotions, and improving the shopping experience for customers by offering tailored product recommendations.

These algorithms' adaptability extends to other domains, such as text mining for often co-occurring phrases, web mining for patterns in user behaviour, and bioinformatics, where they can detect gene correlations. Apriori and FP-Growth are more interpretable and computationally economical than resource-intensive models like Gradient Boosted Trees and Neural Networks, which makes them useful for situations requiring simple association rule mining. In corporate settings where decision-making requires precise insights, interpretability is essential. Furthermore, by enhancing the accuracy and applicability of recommendations, these algorithms aid in the reduction of return rates. They help diagnose and cure patients by finding patterns in their data; in finance, they look for patterns in fraudulent transactions.

7.Conclusion

Summary

Deep Learning models, especially Neural Networks, have dominated the Kaggle competition scene. These models require large amounts of RAM and powerful GPU instances, such as 300GB. Apriori and FP-Growth, two Association Rule Algorithms that are well-known for their effectiveness and lower processing requirements when compared to more sophisticated models like Gradient Boosted Trees and Neural Networks, have been the focus of the project's pivot as a result of these restrictions.

Association Regulation When it comes to market basket analysis, algorithms are the best at figuring out which product groups customers are likely to buy together. Their efficiency in analysing enormous datasets and identifying patterns and relationships among many elements is what makes them successful. Because of this, they can recommend frequently bought items in addition to those that are currently in a user's cart, improving the shopping experience without requiring a lot of processing resources.

The main goal of the project is to create a strong recommendation system that, based on customer purchase patterns, offers intelligent and pertinent choices. The project attempts to maximise resource use while providing users with insightful information and recommendations by utilising FP-Growth and Apriori. This approach fits with the project's objective of providing users with a worthwhile service while making the most use of the resources at hand.

The project's strategic move towards more effective and useful solutions is reflected in its choice to concentrate on Association Rule Algorithms rather than resource-intensive models. The project's goal is to create a recommendation system that provides insightful information without using a lot of processing power by utilising Apriori and FP-Growth. This will increase the project's overall efficacy and efficiency.

Contributions

Novelty

The project's work focuses on using resource-efficient Association Rule Algorithms—specifically, FP-Growth and Apriori—for market basket analysis. This method is in opposition to the prevalent tendency in Kaggle competitions, which strongly favours Deep Learning models that demand high processing power. This work is interesting since it deliberately shifted its focus to lightweight algorithms while taking RAM and processing power limits into account.

The study emphasises the value of taking into account alternate, less resource-intensive methods in data analysis by highlighting how well Apriori and FP-Growth detect product categories and patterns in big datasets. This change in emphasis brings attention to the usefulness and effectiveness of Association Rule Algorithms for market basket analysis, making a significant addition to the area by shedding light on how to create practical and efficient recommendation systems.

Additionally, the project's objective of creating a robust recommendation system based on consumer purchasing trends shows how these algorithms might be used in practice to enhance consumers' shopping experiences. The project advances our understanding and practical use of Association Rule Algorithms by demonstrating the potential of Apriori and FP-Growth in creating effective recommendation systems.

The project's overall contribution is that it shows how Apriori and FP-Growth algorithms work well in market basket analysis, providing a useful substitute for models that require a lot of resources and emphasising their potential for use in real-world recommendation system development.

Individual Contribution

Supraja R (23MSD7015) performed the EDA for the Articles, Customer,Transactions and the images and provided the important features which may be useful for the model building. Also attempted to build KNN models.

Divya Bhardwaj (23MSD7020) did the preprocessing and attempted to build a Apriori Model which was computationally heavy

Pramod Choudhary (23MSD7030) Created the FPGrowth,which was our final model and mined the model for the recommended items for the customers current basket.

Aditya Antony Thomas (23MSD7063) researched about similar topics and attempted to build a LGBM and CAT boost model

Future Work

A number of topics should be looked into for further studies or additional study to improve and expand the project's recommendation system. Algorithm Optimisation: Look on ways to improve the FP-Growth and Apriori algorithms even further to handle enormous datasets even more quickly. This can entail looking into algorithmic advancements or parallel processing methods. Hybrid Models: To develop a hybrid recommendation system, take into account combining Association Rule Algorithms with other lightweight machine learning models, like logistic regression or decision trees. This might enhance the diversity and accuracy of recommendations. User Behaviour Analysis: To improve the recommendation system, examine user purchasing habits in greater detail. This can entail integrating contextual data, user preferences, and temporal patterns into the recommendation system. Dynamic Pricing Strategies: Examine how to incorporate user behaviour and market basket analysis into dynamic pricing strategies. This could aid in the optimisation of price plans and marketing initiatives for increased revenue production. Recommendation in real time: Provide users with immediate recommendations based on their current buying context by developing real-time recommendation capabilities. Efficient processing and analysis of data streams would be necessary for this. Personalisation: Add user comments, demographics, and previous purchase history to the recommendation system to improve its personalisation capabilities. This can result in recommendations that are more pertinent and customised. assessment measures: To gauge the effectiveness of the recommendation system, investigate and put into practice additional assessment measures. This could assist improve the system's algorithms and offer more insights into how effective it is. Interdomain Suggestions: Expand the recommendation system's functionality to include cross-domain suggestions, which make product recommendations from several categories based on user behaviour and preferences. Security and Privacy: Handle security and privacy issues pertaining to user data in the recommendation system. Employ privacy-preserving strategies to safeguard user data while maintaining customised recommendations. Analysis of Market Baskets by Industry: To find insightful trends and insights, apply the market basket analysis approach to industries other than retail, such as e-commerce, healthcare, or finance.

8.References

[1] https://medium.com/@bhavya789shah/fashion-recommender-system-3f56a16f81de

[2] Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. ACM SIGMOD Record, 22(2), 207-216.

[3] Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. ACM SIGMOD Record, 29(2), 1-12.

[4] Zaki, M. J. (2000). Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering, 12(3), 372-390.

[5] https://www.kaggle.com/datasets/odins0n/handm-dataset-128x128?select=articles.csv

[6] https://www.kaggle.com/code/mahluo/2023-kaggle-h-m-eda-first-look

9.Appendices

Appendix A: Additional figures, tables and charts.

Articles.csv

```
articles.head()
```

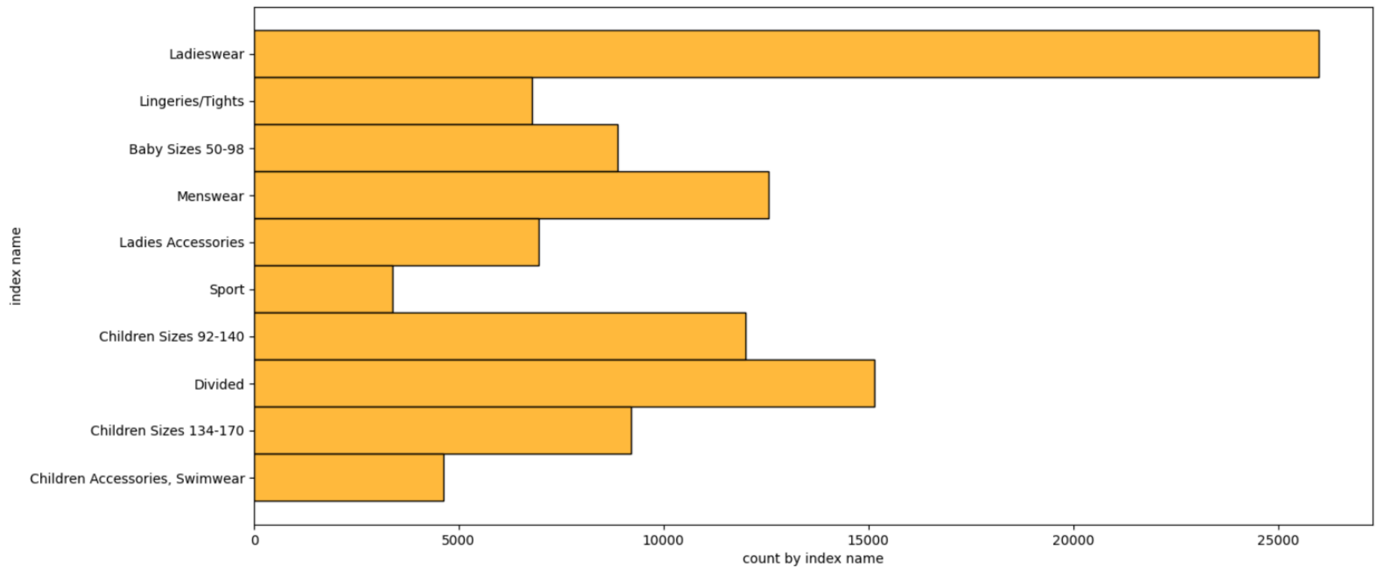
	article_id	product_code	prod_name	product_type_no	product_type_name	product_group_name	graphical_appearance_no	graphical_appearance_name	colour_group_code	colour_group_name	perceived_colour_value_id
0	108775015	108775	Strap top	253	Vest top	Garment Upper body	1010016	Solid	9	Black	4
1	108775044	108775	Strap top	253	Vest top	Garment Upper body	1010016	Solid	10	White	3
2	108775051	108775	Strap top (1)	253	Vest top	Garment Upper body	1010017	Stripe	11	Off White	1
3	110065001	110065	OP T-shirt (Idro)	306	Bra	Underwear	1010016	Solid	9	Black	4
4	110065002	110065	OP T-shirt (Idro)	306	Bra	Underwear	1010016	Solid	10	White	3

Customers.csv

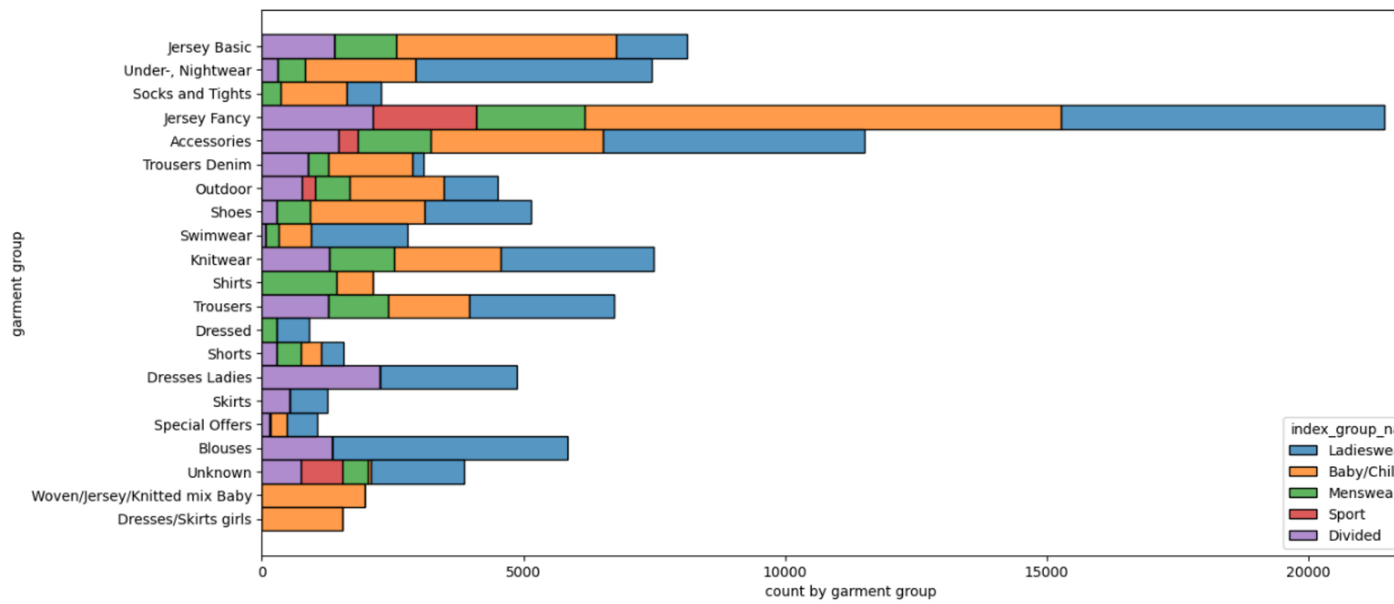
```
pd.options.display.max_rows = 50
customers.head()
```

	customer_id	FN	Active	club_member_status	fashion_news_frequency	age	postal_code
0	00000dbacae5abe5e23885899a1fa44253a17956c6d1c3...	NaN	NaN	ACTIVE	NONE	49.0	52043ee2162cf5aa7ee79974281641c6f11a68d276429a...
1	0000423b00ade91418cceaf3b26c6af3dd342b51fd051e...	NaN	NaN	ACTIVE	NONE	25.0	2973abc54daa8a5f8ccfe9362140c63247c5eee03f1d93...
2	000058a12d5b43e67d225668fa1f8d618c13dc232df0ca...	NaN	NaN	ACTIVE	NONE	24.0	64f17e6a330a85798e4998f62d0930d14db8db1c054af6...
3	00005ca1c9ed5f5146b52ac8639a40ca9d57aeff4d1bd2...	NaN	NaN	ACTIVE	NONE	54.0	5d36574f52495e81f019b680c843c443bd343d5ca5b1c2...
4	00006413d8573cd20ed7128e53b7b13819fe5cfc2d801f...	1.0	1.0	ACTIVE	Regularly	52.0	25fa5ddee9aac01b35208d01736e57942317d756b32ddd...

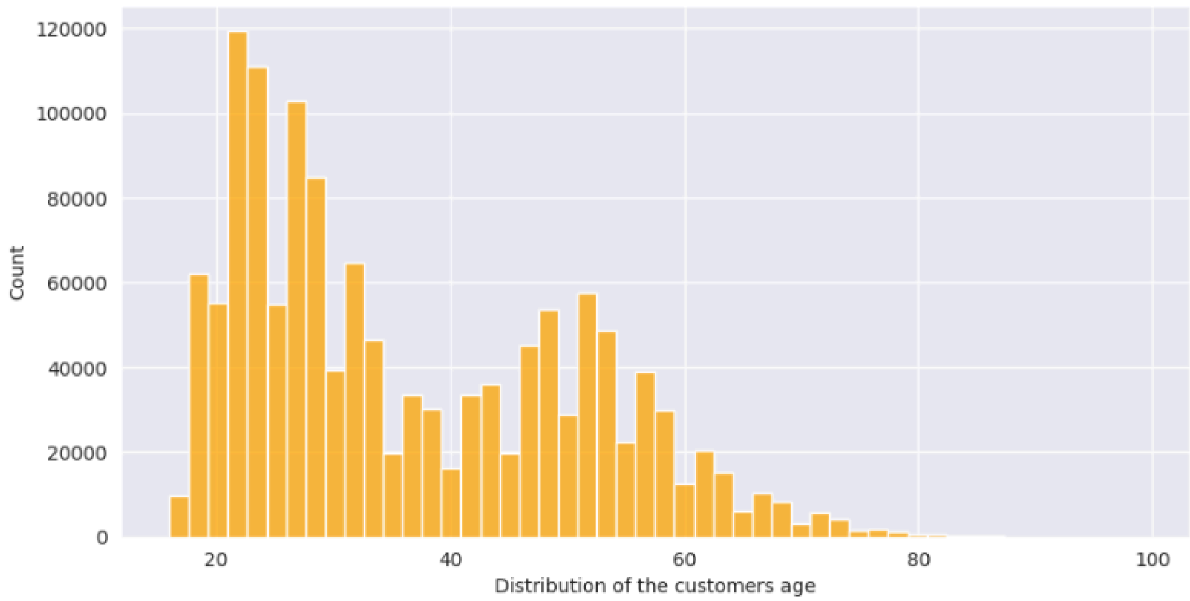
Count of articles according to Groups



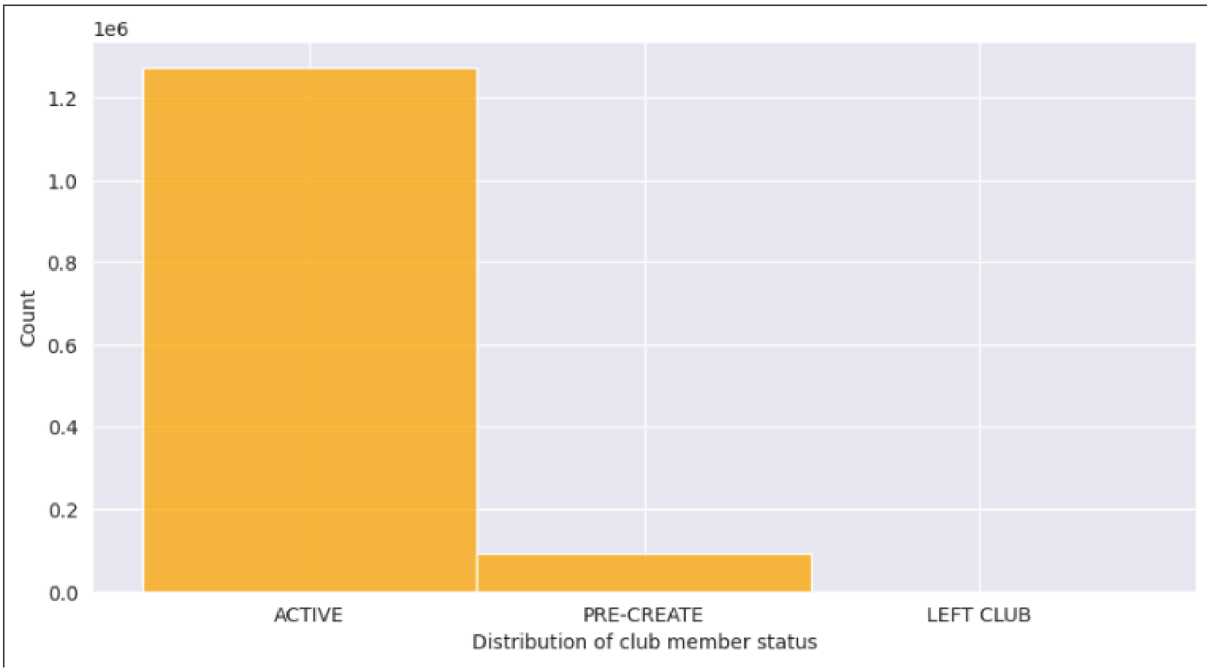
Stacked Bar Chart of Articles Types with Departments



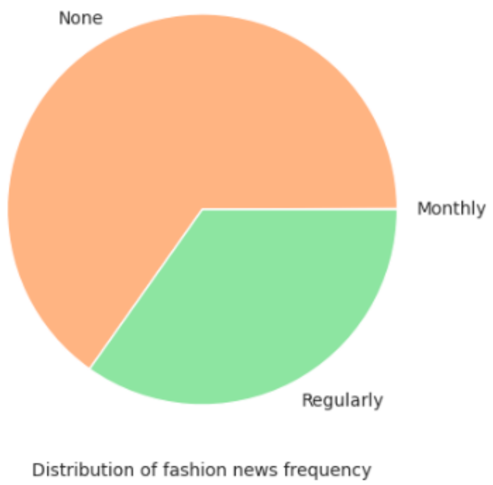
Age of Customers



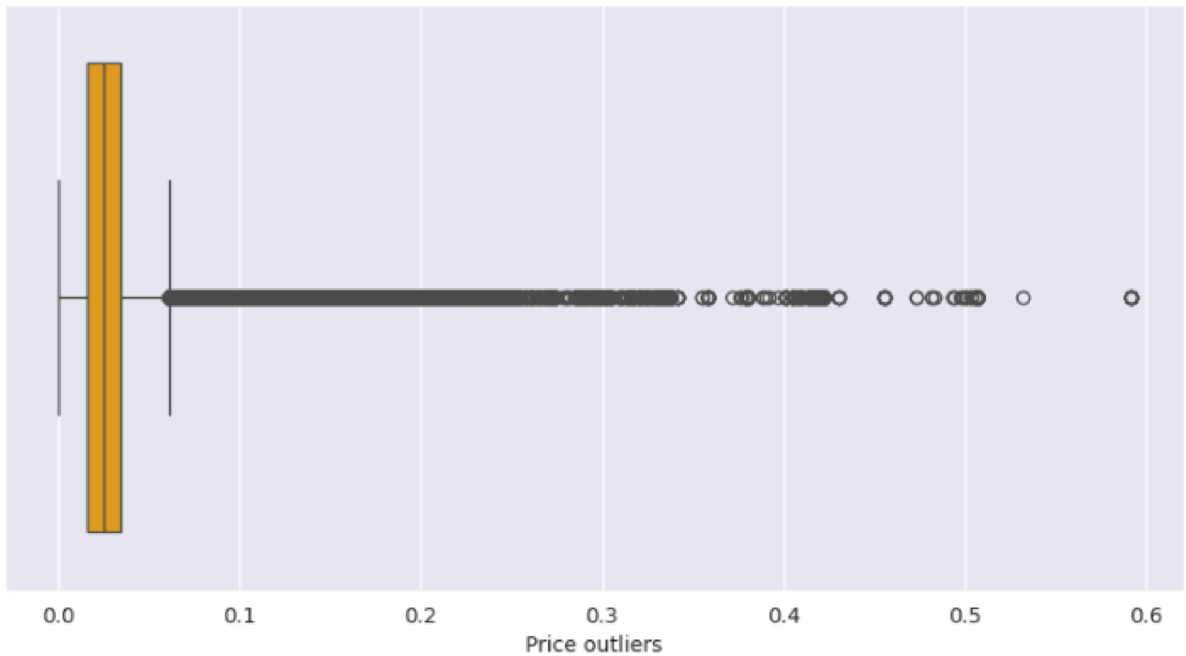
Active Club Subscription of Customers



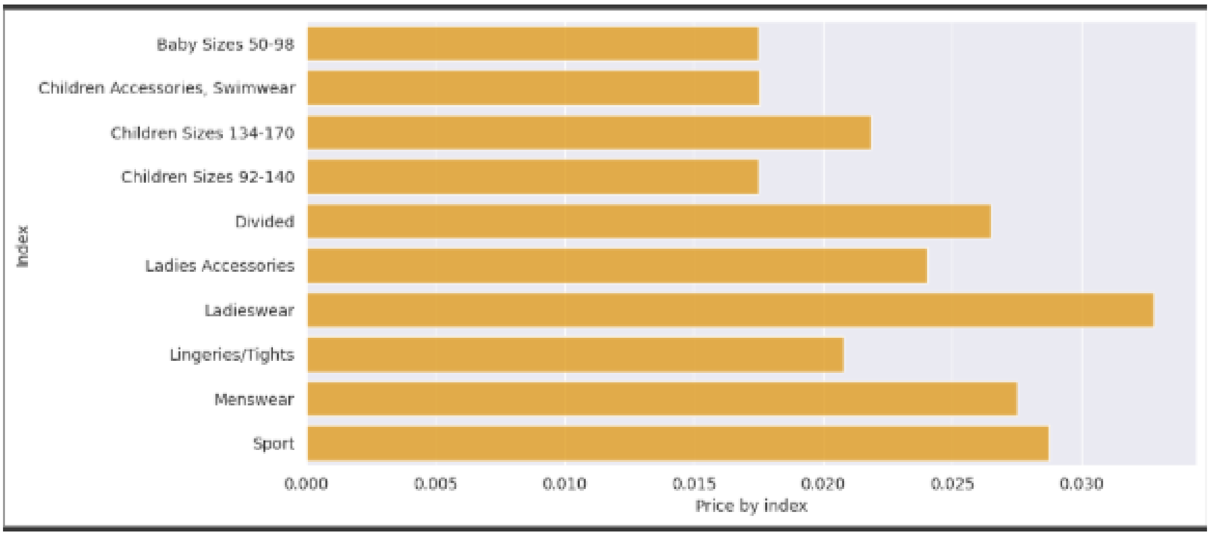
News Subscription For Fashion



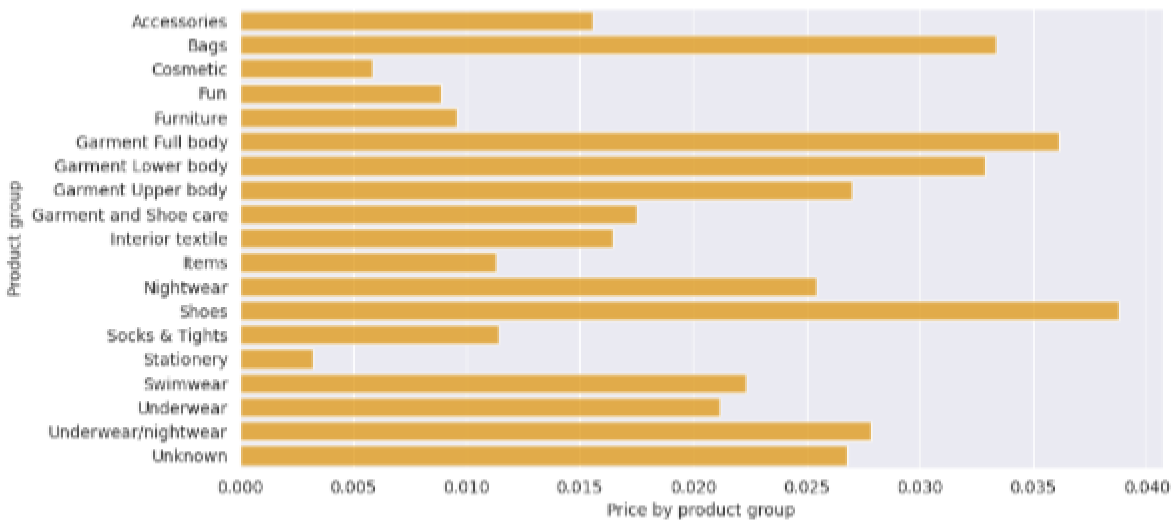
Price Distribution of Articles



Prices By Departments



Prices by Garment Type



Appendix B: Code snippets

FOR EDA

To create histogram of Department and Prices

```
f, ax = plt.subplots(figsize=(15, 7))
ax = sns.histplot(data=articles, y='index_name', color='orange')
ax.set_xlabel('count by index name')
ax.set_ylabel('index name')
plt.show()
```

To create histogram of Item Types and Prices

```
import seaborn as sns
from matplotlib import pyplot as plt
sns.set_style("darkgrid")
f, ax = plt.subplots(figsize=(10,5))
ax = sns.histplot(data=customers, x='age', bins=50, color='orange')
ax.set_xlabel('Distribution of the customers age')
plt.show()
```

To create the histogram for Customers who are club members

```
sns.set_style("darkgrid")
f, ax = plt.subplots(figsize=(10,5))
ax = sns.histplot(data=customers, x='club_member_status', color='orange'
ax.set_xlabel('Distribution of club member status')
plt.show()
```

To create a pie chart which shows News Frequency

```
sns.set_style("darkgrid")
f, ax = plt.subplots(figsize=(10,5))
# ax = sns.histplot(data=customers, x='fashion_news_frequency', color='orange')
# ax = sns.pie(data=customers, x='fashion_news_frequency', color='orange')
colors = sns.color_palette('pastel')
ax.pie(pie_data.customer_id, labels=pie_data.index, colors = colors)
ax.set_facecolor('lightgrey')
ax.set_xlabel('Distribution of fashion news frequency')
plt.show()
#Customers prefer not to get any messages about the current news.
```

To create Box plot to figure out outliers

```
sns.set_style("darkgrid")
f, ax = plt.subplots(figsize=(10,5))
ax = sns.boxplot(data=transactions, x='price', color='orange')
ax.set_xlabel('Price outliers')
plt.show()
```

To create histogram which displays price outliers and count

```
#Top 10 customers by num of transactions.
sns.set_style("darkgrid")
f, ax = plt.subplots(figsize=(10,5))
ax = sns.histplot(data=transactions, x='price',bins=100, kde=False,color='orange')
ax.set_xlabel('Price outliers')
plt.show()
```

To create barplot for prices vs Categories

```
articles_index = articles_for_merge[['index_name', 'price']].groupby('index_name').mean()
sns.set_style("darkgrid")
f, ax = plt.subplots(figsize=(10,5))
ax = sns.barplot(x=articles_index.price, y=articles_index.index, color='orange', alpha=0.8)
ax.set_xlabel('Price by index')
ax.set_ylabel('Index')
plt.show()
#The index with the highest mean price is Ladieswear. With the lowest - children
```

To create barplot for Garment type and prices

```
articles_index = articles_for_merge[['product_group_name', 'price']].groupby('product_group_name').mean()
sns.set_style("darkgrid")
f, ax = plt.subplots(figsize=(10,5))
ax = sns.barplot(x=articles_index.price, y=articles_index.index, color='orange', alpha=0.8)
ax.set_xlabel('Price by product group')
ax.set_ylabel('Product group')
plt.show()
#Stationery has the lowest mean price, the highest - shoes.
```

FOR MODEL

```
transaction_data = pd.read_csv('/content/drive/MyDrive/fashion/Dataset/transactions_train.csv')

# Convert the last 16 characters of 'customer_id' to hex, then to int64
transaction_data['customer_id'] = transaction_data['customer_id'].str[-16:].apply(lambda x: int(x, 16)).astype('int64')

# Convert 'article_id' to int32
transaction_data['article_id'] = transaction_data['article_id'].astype('int32')

# Convert 't_dat' to datetime
transaction_data['t_dat'] = pd.to_datetime(transaction_data['t_dat'])

# Select specific columns and rearrange the DataFrame
transaction_data = transaction_data[['t_dat', 'customer_id', 'article_id', 'price']]
```

```
## Data preprocess
transaction_data.loc[:,'t_dat'] = pd.to_datetime(transaction_data['t_dat'])
transaction_data.loc[:, 'year'] = transaction_data['t_dat'].dt.year
transaction_data.loc[:, 'month'] = transaction_data['t_dat'].dt.month
transaction_data.loc[:, 'day'] = transaction_data['t_dat'].dt.day
```

```
customers_data = pd.read_csv("/content/drive/MyDrive/fashion/Dataset/customers.csv")
customers_data['customer_id'] = customers_data['customer_id'].str[-16:].apply(lambda x: int(x, 16)).astype('int64')

# Read articles data
articles_data = pd.read_csv("/content/drive/MyDrive/fashion/Dataset/articles.csv")

# Filter transaction data for the year 2020, month > 6, and day > 20
transactions_data_2020 = transaction_data[(transaction_data['t_dat'].dt.year == 2020) &
                                          (transaction_data['t_dat'].dt.day <= 7)]

# Group transactions in 2020 by customer_id and retrieve unique article_ids
transactions_customers_all_articles = transactions_data_2020.groupby('customer_id')['article_id'].unique().reset_index()
```

```
from itertools import chain

# Extract unique itemsets as tuples
unique_itemsets_tuples = frequent_itemsets['itemsets'].unique()

# Flatten the tuples to get individual items
unique_items = list(set(chain.from_iterable(unique_itemsets_tuples)))
```

```python
def recommend_items_based_on_basket(basket, frequent_itemsets):
    filtered_itemsets = frequent_itemsets[frequent_itemsets['itemsets'].apply(lambda x: all(item in x for item in basket))]
    recommendations = [item for itemset in filtered_itemsets['itemsets'] for item in itemset if item not in basket]

    return recommendations[:2]

# Example basket
basket =['721270016', '831467001']
total=[]
for i in basket:
  l=[i]
  recommendations = recommend_items_based_on_basket(l, frequent_itemsets)
  for i in recommendations:
    if i not in total:
      total.append(i)
print("Top Recommendations based on Basket:", total)
```

```python
path="/content/drive/MyDrive/fashion/Dataset/images_128_128"
f, ax1 = plt.subplots(1, len(basket), figsize=(5, 5))
for i, article in enumerate(basket):
    file_name = "0" + str(article) + ".jpg"
    dir_name = "0" + str(article)[:2]
    image = mpimg.imread(path + "/" + dir_name + "/" + file_name)
    ax = ax1 if len(basket) == 1 else ax1[i]  # Adjust for single or multiple axes
    ax.imshow(image)
    ax.set_xticks([])
    ax.set_yticks([])
    ax.grid(False)

plt.show()
```

```python
f, ax1 = plt.subplots(1, len(total), figsize=(10, 10))
if len(total) == 1:
    ax1 = [ax1]  # Convert single axes object to list

for i, article in enumerate(total):
    file_name = "0" + str(article) + ".jpg"
    dir_name = "0" + str(article)[:2]
    image = mpimg.imread(path + "/" + dir_name + "/" + file_name)
    ax = ax1[i]  # Access individual axes object
    ax.imshow(image)
    ax.set_xticks([])
    ax.set_yticks([])
    ax.grid(False)

plt.show()
```