Assignment 3

Itemset Mining and Recommendation System on Groceries Dataset

Mining Big Data | Trimester 1 2023

PREPARED BY

Harpreet Kaur Hans | a1873328

Priyank Dave | a1843068

Chinmay Dharmik | a1855351

EXECUTIVE SUMMARY

Our team of developers has been tasked with creating a system for a grocery store to identify frequently bought itemsets by customers, in order to group them together for increased sales. Additionally, we will recommend products from these itemsets to customers on the store's website. We have implemented three models to achieve this purpose. All of our models utilise a dataset which was generated by using a loyalty programme to gather customer transaction data, which we have used to identify frequent itemsets. These models are end to end, meaning they take the dataset as the input and produce recommendations as output. We have tested the models with 27,000 transactions and have found that it is feasible to scale the solution to one million customers with the power of parallel processing and better computer hardware. Out of the three models which we have tested, one of them, namely Collaborative Filtering through KNN, is an effective method for recommending products to customers. The benefits of using this model for the grocery store include increased sales by grouping frequently bought itemsets together and recommending these products to customers. The system is scalable, meaning it can handle increasing transaction volumes. Furthermore, the implementation of this system will position the grocery store as a leader in using data-driven solutions for enhancing the customer experience and increasing sales.

INTRODUCTION

This assignment focuses on Pattern Mining on a Groceries dataset with the main aim of recommending products based on items in a customer's basket. In today's world, where online shopping is gaining momentum, it becomes imperative to personalise the shopping experience for each customer. To achieve this, we have used three popular techniques for recommendation systems- Apriori, Collaborative Filtering through KNN and FP-growth. Apriori is a widely used association rule mining algorithm that identifies the frequent itemsets and generates association rules based on the customer's purchase history. Collaborative Filtering through KNN clustering leverages the purchase history of similar customers to identify groups of similar customers based on their purchase history and recommends products to them. We have evaluated the performance of these methods using Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) as evaluation metrics, which helps us test the predictions made by the recommendation algorithms. Through this assignment, we aim to provide a personalised shopping experience to the customers and help the supermarket increase their sales by recommending products to the customers based on their purchase history.

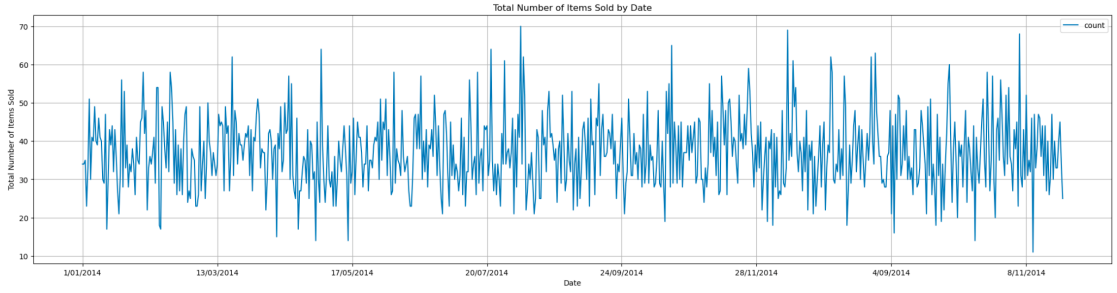
EXPLORATORY DATA ANALYSIS

Data Description and Visualisations :

The Groceries dataset used in this analysis consists of transactional data from a retail store. The dataset includes information on customer purchases, with each transaction represented by a unique identifier. The dataset contains a total of 27000 transactions for training data.

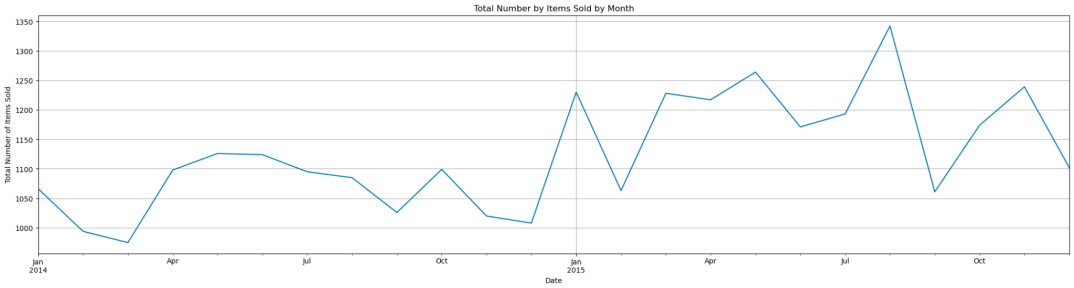
	Member_number	Date	itemDescription	year	month	day	day_of_week
0	3021	30/01/2015	frankfurter	2015	1	30	4
1	1292	24/10/2015	pork	2015	10	24	5
2	4206	4/04/2014	root vegetables	2014	4	4	4
3	4369	25/08/2015	onions	2015	8	25	1
4	1522	1/07/2014	waffles	2014	7	1	1

The dataset consists of 167 unique items sold over a period of 728 days, which is roughly equivalent to 24 months. During this time period, a total of 27,000 items were sold, which translates to an average of approximately 37 items sold per day. These statistics provide a general overview of the sales activity over the entire time period and can be useful in understanding the volume and frequency of sales.



The above graph plot shows the total number of items sold by date, which can be useful in identifying sales trends over time. By visualising the data in this way, it is possible to gain insights into which

products are most popular and when, which can inform decisions related to inventory management and sales forecasting. This information can be valuable for businesses looking to optimise their operations and improve their bottom line.



The above graph/plot shows the number of items sold each day as well as by month , which can help identify sales patterns over time. By visualising the data in this way, businesses can gain insights into busy periods and adjust their inventory or staffing levels accordingly. This information can be valuable for optimising operations and improving profitability.

	itemDescription
whole milk	1709
other vegetables	1320
rolls/buns	1197
soda	1060
yogurt	928
root vegetables	733
tropical fruit	683
bottled water	666
sausage	637
citrus fruit	560

The above graph is a treemap visualisation of the ITEM DataFrame, which can be useful in visualising the hierarchical relationships between different items and their respective counts. By examining the size and colour of the rectangles in the treemap, businesses can gain insights into the relative popularity of different items and adjust their inventory or pricing strategies accordingly.

Then we aggregate the purchase history of each member on each day, allowing us to analyse the items commonly purchased together and to perform analysis to identify association rules between different items. This transforms the transaction-level data into a format suitable for applying the Apriori algorithm.

In order to perform market basket analysis, we need transaction-level data that records the items purchased by each customer on each visit to the store. This data can be messy and difficult to analyse directly. Therefore, the first step in analysis is to transform the transaction-level data into a more suitable format.

We achieve the transformation by grouping the rows in the data DataFrame by Member_number and Date using the groupby method. For each group, it selects the itemDescription column using the .loc accessor and converts the resulting Series to a list using the tolist() method. This creates a list of lists, where each inner list contains the items purchased by a single customer on a single day.

This format is suitable for applying the Apriori algorithm to identify frequent itemsets and association rules between different items. The Apriori algorithm works by generating frequent itemsets from the transaction data and then deriving association rules between the items in these frequent itemsets. The resulting rules can be used to identify patterns in customer behaviour, such as which items are commonly purchased together or which items are frequently purchased by customers who also buy a certain product.

Item	Items purchased by customer (Basket)
1	['sausage', 'yogurt']
2	['misc. beverages', 'canned beer']
3	['soda', 'pickled vegetables']
4	['rolls/buns', 'whole milk', 'sausage']
5	['soda', 'whipped/sour cream']

Data Preprocessing :

According to Zhou et al. (2020), data preprocessing is a crucial step in preparing transaction data for market basket analysis. To transform the list of transactions into a format suitable for various machine learning algorithms, including Apriori algorithm, the TransactionEncoder class from the mlxtend library is employed. This class transforms the list of transactions into a binary format, where each row corresponds to a transaction, and each column corresponds to a unique item. If an item appears in the transaction, the corresponding entry in the matrix is set to True; otherwise, it is set to False. This binary matrix is known as the "transaction encoding" of the input data (Zhou et al., 2020).

Pattern Mining:

According to Jain and Mehta (2014), pattern mining refers to the process of extracting knowledge from vast amounts of data. It involves locating significant patterns in huge and complicated data sets using science, art, and technology. One of the techniques used in pattern mining is association rule mining. Association rule mining is a data mining method that identifies intriguing connections between elements in a sizable dataset. Two of the most common and efficient techniques for association rule mining are the Apriori algorithm and FP Growth (Jain & Mehta, 2014).

The Apriori algorithm, as explained by Jain and Mehta (2014), consists of two components: candidate generation and support calculation. Candidate generation involves generating candidate itemsets of increasing size. The algorithm's second stage involves calculating the support of each candidate itemset.

On the other hand, the FP-growth algorithm is a popular method for mining frequent itemsets in large datasets. It involves building a frequent pattern tree (FP-tree) and recursively extracting frequent itemsets from the tree. Building an FP-tree involves scanning the dataset and constructing a compact

tree structure that represents all frequent itemsets and their relationships. Mining frequent itemsets from the FP-tree involves recursively extracting frequent itemsets from the tree (Jain & Mehta, 2014).

Components of Association Rules:

1. *Support* : The number of transactions in which an itemset appears divided by the total number of transactions is the itemset's support.

$$Support(X) = (Number\ of\ transactions\ containing\ X) / (Total\ number\ of\ transactions)$$

2. *Confidence :* The confidence of a rule A → B is the percentage of transactions containing A that also contain B. A high confidence indicates a strong association between the items A and B.

$$\text{Confidence}(A \rightarrow B) = P(B \mid A) = (\text{support}(A \cup B) / \text{support}(A))$$

A confidence level of 1 means that the consequent always happens when the antecedent does, with a confidence score ranging from 0 to 1. In contrast to a low confidence level, a high confidence score indicates a significant link between the antecedent and the consequent.

Association Rules obtained using Frequent Item Set :

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(UHT-milk)	(other vegetables)	0.015035	0.092799	0.001223	0.081340	0.876514	-0.000172	0.987526	-0.125135
1	(other vegetables)	(UHT-milk)	0.092799	0.015035	0.001223	0.013178	0.876514	-0.000172	0.998119	-0.134419
2	(UHT-milk)	(tropical fruit)	0.015035	0.048414	0.001007	0.066986	1.383607	0.000279	1.019905	0.281483
3	(tropical fruit)	(UHT-milk)	0.048414	0.015035	0.001007	0.020802	1.383607	0.000279	1.005890	0.291357
4	(UHT-milk)	(whole milk)	0.015035	0.118049	0.001583	0.105263	0.891690	-0.000192	0.985710	-0.109782

Other Attributes are as follows :

Lift (L) : The lift of the rule X=>Y is the rule's confidence divided by the predicted confidence, supposing that the items X and Y are distinct from one another.The predicted confidence is calculated by dividing the confidence by the frequency of Y.

Antecedents and Consequents: The antecedent is the set of items that appear in the transactions or baskets under consideration. The group of events that are anticipated to follow the antecedent make up the consequent. In other words, the set of items that are likely to also occur in the same transaction is the consequent if the antecedent is present in a transaction.

COLLABORATIVE FILTERING RECOMMENDATION METHOD

According to our implementation of Collaborative Filtering, we utilised K-Nearest Neighbours (KNN) as a means of identifying similar items based on given items. This method of item-based collaborative filtering employs K-nearest neighbours, and recommendations are made based on the similarity identified (Sahu & Sahoo, 2021). In transforming the Groceries dataset into a frequency table, each row corresponds to a member and each column corresponds to an item in the store catalogue. The Nearest Neighbours model is then fitted on this transformed data, where each item serves as a feature. While our code does not explicitly create an item similarity matrix, the Nearest Neighbours model implicitly calculates the distances between items based on the frequency of their co-occurrence in transactions.

The distance matrix serves as a measure of similarity between items and is used by the KNN algorithm to identify the nearest neighbours (Sahu & Sahoo, 2021).

The recommend method in this model generates recommendations. This method takes a set of item names as input, representing the query. The query items are converted into a dictionary where each item is assigned a value of 0, then transformed into a Pandas DataFrame. The method then identifies the 5 nearest neighbours to the query in the transformed data using the nearest neighbours model. To generate recommendations, the method looks for items that the nearest neighbours bought but the query member hasn't. It does so by searching for rows in the transformed data that correspond to the nearest neighbours and identifying items with frequency 1 in those rows. The recommended items are returned in a list.

RECOMMENDATION METHOD FROM FREQUENT PATTERNS

According to Han et al. (2000), Apriori and FP-growth are two popular algorithms for mining frequent itemsets from large datasets. These algorithms have practical applications in various areas such as customer behaviour analysis and product recommendations. The Apriori algorithm generates candidate itemsets and prunes those that do not meet a minimum support threshold, while FP-growth uses a tree-based data structure to efficiently mine frequent itemsets without generating candidate itemsets. After obtaining frequent itemsets, association rules of the form X -> Y can be generated, where X and Y are itemsets, indicating that if a customer purchases items in X, they are likely to purchase items in Y as well. The strength of an association rule is measured by the confidence, which is the fraction of transactions containing X that also contain Y (Agrawal & Srikant, 1994).

In this study, we are using both Apriori and FP-growth algorithms to generate frequent itemsets and association rules from a grocery transaction dataset. The Apriori approach is implemented using the apriori function from the mlxtend library, while the FP Growth Method is implemented using the FP-growth function from the same library (Raschka, 2018). The resulting association rules can be used to make recommendations to customers based on their purchase history, thus improving customer satisfaction and increasing sales (Han et al., 2000).

In order to generate recommendations, the association rules are ranked based on their strength or confidence, which is a measure of how often the rule holds true. The higher the confidence, the more likely the recommendation will be useful to the user. The support of the rule, which is a measure of how frequently the itemset appears in the dataset, is also taken into account.

Once the association rules are ranked, they can be used to generate recommendations for users. This is done by finding all the items that the user has already purchased or interacted with and then recommending other items that appear in the right-hand side of the highest-ranked association rules that involve these items. These recommended items are presented to the user in order of descending confidence or support.

DISCUSSION OF RESULTS

To evaluate the effectiveness of the recommendation system, the test dataset is used to calculate metrics such as Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR).

MAP measures the average quality of the recommendations given to a user. It takes into account both the relevance of the recommended items (i.e., how many of the recommended items are actually relevant to the user) and the ranking of those items (i.e., how high in the list of recommendations the relevant items appear). MAP is calculated by averaging the average precision scores across all users in the test dataset.

MRR, on the other hand, measures the effectiveness of the recommendation system in terms of how quickly it can provide the user with a relevant item. It is calculated as the reciprocal of the rank of the first relevant item in the list of recommendations, averaged across all users in the test dataset.

In this study, the recommendation system uses association rules generated from the Apriori algorithm to recommend items to users. The patterns and recommendations are ranked based on their confidence scores, which measure the strength of the association between items in the rule. The higher the confidence score, the more likely it is that a user will be interested in the recommended item.

The evaluation of the recommendation system is performed on a test dataset, where the system predicts the items that a user is likely to be interested in based on their past behaviour. The MAP and MRR scores are then calculated based on the actual and predicted item sets for each user in the test dataset.

The following are five examples of frequent patterns with their confidence and support on both training and test sets. Since the Collaborative filtering through KNN method directly gives recommendations from the dataset, we have presented the frequent patterns only from the Apriori and FP-growth models below.

Apriori on Train data:

...	antecedents	consequents	support	confidence
453	(whole milk)	(rolls/buns)	0.007913	0.067032
452	(rolls/buns)	(whole milk)	0.007913	0.094746
386	(whole milk)	(other vegetables)	0.006978	0.059110
387	(other vegetables)	(whole milk)	0.006978	0.075194
517	(yogurt)	(whole milk)	0.006546	0.101790

Apriori on Test data:

	antecedents	consequents	support	confidence
27	(other vegetables)	(whole milk)	0.003306	0.050611
26	(whole milk)	(other vegetables)	0.003306	0.037132
36	(whole milk)	(rolls/buns)	0.002850	0.032010
37	(rolls/buns)	(whole milk)	0.002850	0.048356
19	(other vegetables)	(rolls/buns)	0.002736	0.041885

FP-growth on Train data:

	antecedents	consequents	support	confidence
83	(whole milk)	(rolls/buns)	0.007913	0.067032
82	(rolls/buns)	(whole milk)	0.007913	0.094746
197	(other vegetables)	(whole milk)	0.006978	0.075194
196	(whole milk)	(other vegetables)	0.006978	0.059110
0	(whole milk)	(yogurt)	0.006546	0.055454

FP-growth on Test data:

	antecedents	consequents	support	confidence
2	(whole milk)	(other vegetables)	0.003306	0.037132
3	(other vegetables)	(whole milk)	0.003306	0.050611
4	(whole milk)	(rolls/buns)	0.002850	0.032010
5	(rolls/buns)	(whole milk)	0.002850	0.048356
6	(rolls/buns)	(other vegetables)	0.002736	0.046422

The following are 10 examples of recommendations from these patterns, two examples from each of the above patterns:

```
########## Recommendation of the item 1: rolls/buns ##########
Apriori : ['whole milk', 'other vegetables', 'yogurt', 'soda', 'root vegetables']
########## Recommendation of the item 2: whole milk ##########
Apriori : ['rolls/buns', 'other vegetables', 'yogurt', 'soda', 'sausage']
########## Recommendation of the item 3: other vegetables ##########
Apriori : ['whole milk', 'soda', 'rolls/buns', 'yogurt', 'tropical fruit']
########## Recommendation of the item 4: other vegetables ##########
Apriori : ['whole milk', 'soda', 'rolls/buns', 'yogurt', 'tropical fruit']
########## Recommendation of the item 5: yogurt ##########
Apriori : ['whole milk', 'rolls/buns', 'other vegetables', 'sausage', 'soda']
########## Recommendation of the item 6: root vegetables ##########
Apriori : ['rolls/buns', 'whole milk', 'other vegetables', 'soda', 'sausage']
########## Recommendation of the item 7: soda ##########
Apriori : ['whole milk', 'other vegetables', 'rolls/buns', 'sausage', 'yogurt']
########## Recommendation of the item 8: sausage ##########
Apriori : ['whole milk', 'soda', 'yogurt', 'rolls/buns', 'other vegetables']
########## Recommendation of the item 9: abrasive cleaner ##########
Apriori : []
########## Recommendation of the item 10: artif. sweetener ##########
Apriori : []
```

```
########## Recommendation of the item 1: rolls/buns ##########
FP Growth : ['whole milk', 'other vegetables', 'yogurt', 'soda', 'root vegetables']
########## Recommendation of the item 2: whole milk ##########
FP Growth : ['rolls/buns', 'other vegetables', 'yogurt', 'soda', 'sausage']
########## Recommendation of the item 3: other vegetables ##########
FP Growth : ['whole milk', 'soda', 'rolls/buns', 'yogurt', 'tropical fruit']
########## Recommendation of the item 4: other vegetables ##########
FP Growth : ['whole milk', 'soda', 'rolls/buns', 'yogurt', 'tropical fruit']
########## Recommendation of the item 5: yogurt ##########
FP Growth : ['whole milk', 'rolls/buns', 'other vegetables', 'sausage', 'soda']
########## Recommendation of the item 6: root vegetables ##########
FP Growth : ['rolls/buns', 'whole milk', 'other vegetables', 'soda', 'sausage']
########## Recommendation of the item 7: soda ##########
FP Growth : ['whole milk', 'other vegetables', 'rolls/buns', 'sausage', 'yogurt']
########## Recommendation of the item 8: sausage ##########
FP Growth : ['whole milk', 'soda', 'yogurt', 'rolls/buns', 'other vegetables']
########## Recommendation of the item 9: abrasive cleaner ##########
FP Growth : []
########## Recommendation of the item 10: artif. sweetener ##########
FP Growth : []
```

```
########## Recommendation of the item 1: rolls/buns ##########
K Nearest Neighbour : {'rolls/buns'}
########## Recommendation of the item 2: whole milk ##########
K Nearest Neighbour : {'whole milk'}
########## Recommendation of the item 3: other vegetables ##########
K Nearest Neighbour : {'other vegetables'}
########## Recommendation of the item 4: other vegetables ##########
K Nearest Neighbour : {'other vegetables'}
########## Recommendation of the item 5: yogurt ##########
K Nearest Neighbour : {'yogurt'}
########## Recommendation of the item 6: root vegetables ##########
K Nearest Neighbour : {'root vegetables', 'tropical fruit', 'citrus fruit'}
########## Recommendation of the item 7: soda ##########
K Nearest Neighbour : {'soda'}
########## Recommendation of the item 8: sausage ##########
K Nearest Neighbour : {'sausage', 'soda', 'tropical fruit'}
########## Recommendation of the item 9: abrasive cleaner ##########
K Nearest Neighbour : {'rolls/buns', 'yogurt', 'newspapers', 'other vegetables', 'whipped/sour cream'}
########## Recommendation of the item 10: artif. sweetener ##########
K Nearest Neighbour : {'coffee', 'tropical fruit', 'fruit/vegetable juice', 'other vegetables', 'chicken'}
```

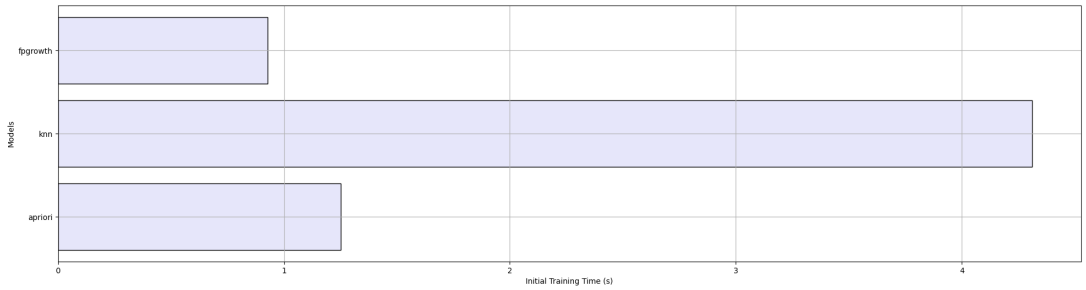
The following table summarises the performance metrics used:

	Apriori	KNN	FP-Growth
MAP	0.2510468400038952	0.9437872449918029	0.2510468400038952
MRR	0.2510468400038952	0.9816884149706219	0.2510468400038952

The results of the mean average precision analysis show that the KNN model has significantly higher accuracy than the other two models. The Apriori and FP Growth models have the same accuracy score, which is lower than that of the KNN model. This suggests that the KNN model may be the best option for generating accurate product recommendations for customers.

The three models have different MRR scores, which indicate their ability to make accurate recommendations. Both Apriori and FP-growth have the same score, which is considerably lower than the KNN model. This suggests that the association rule mining techniques used by Apriori and FP-growth may not be as effective for this particular dataset, while the KNN model performs better in terms of recommending items that the user is likely to purchase. Therefore, the KNN model could be a better choice for making personalised recommendations to users.

The following graph depicts the time the models take to run on a dataset of size 27,000. It can be seen that FP-growth performs the best and KNN the worst. Therefore, when scaled to a million transactions, FP-growth would have the best chance.



Nonetheless, it should be noted that an estimate of timing of the system if the dataset is scaled up to one million transactions also depends on the hardware of the computer used. Therefore, looking at the time complexity in this case could give a better perspective.

CONCLUSION AND RECOMMENDATIONS

The results of the evaluation show that the K-Nearest Neighbour (KNN) model outperforms both the Apriori and FP-Growth models for both mean average precision (MAP) and mean reciprocal rank (MRR) metrics. The MAP values for Apriori and FP-Growth are identical, indicating that they performed similarly in terms of precision, while KNN achieved a much higher MAP score of 0.943. Similarly, the MRR score for KNN is much higher than the other two models, indicating that KNN performed better at identifying the most relevant item for a given user. Therefore, the results suggest that KNN is better suited to this task than the Apriori and FP-growth models. This is likely because KNN is able to take into account user-item interactions, whereas the other two models rely solely on item-item associations. Additionally, the fact that KNN achieves a significantly higher MRR score indicates that it is better at identifying the most relevant item for each user, which is the ultimate goal of a recommendation system.

However it should be noted, while KNN outperforms the other two models in terms of performance metrics, it is in fact the slowest model. This implies that it would take a lot of time to produce recommendations when provided with a million transactions as input. Nevertheless, this model can be made scalable by using parallel processing techniques. Therefore, it has the potential to become the ideal model and hence is recommended. Such a model could boost the sales of the grocery store using it and could increase profit. Further, it could enhance customer experience, making the store more renowned and inturn attracting even more customers.

The considerations while using these models in the future could include utilising parallel processing techniques and conducting online A/B tests in order to “measure real impact on real users” (Rakesh4real, 2019).

REFLECTION

One of the main advantages of using these algorithms (Apriori and FP Growth) in recommendation systems is their ability to efficiently mine frequent itemsets and generate association rules. These rules can then be used to suggest items that are likely to be of interest to a user based on their historical preferences or behaviour. The use of these algorithms can also help to reduce the computational complexity of the recommendation process, making it feasible to generate recommendations in real-time for large datasets.

However, there are also some limitations to using Apriori and FP-growth algorithms in recommendation systems. For example, these algorithms do not account for the context of user preferences, such as the time of day, user location, or social context, which can be important factors in determining a user's preferences. In addition, these algorithms rely on the assumption of independence between items, which may not always hold true in real-world scenarios where users may have complex and evolving preferences.

Overall, the use of Apriori and FP-growth algorithms can provide a useful starting point for building recommendation systems, but it is important to also consider other factors, such as user context and domain-specific knowledge, in order to provide more accurate and personalised recommendations.

Furthermore, KNN is a very storage heavy and computation heavy algorithm. We believe using distributed computing techniques, for example using frameworks such as PySpark or Pytorch distributed training, will yield faster training time and prediction time.

REFERENCES

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In Proceedings of the 20th International Conference on Very Large Data Bases (pp. 487-499).

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. In Proceedings of the ACM SIGMOD International Conference on Management of Data (pp. 1-12).

Jain, A. & Mehta, N. (2014). A Comparative Study of Apriori and FP-Growth Algorithm. International Journal of Innovative Research in Computer and Communication Engineering, 2(5), 3265-3270.

Rakesh4real (2019). Evaluating Recommendation Systems — Part 2. [online] Medium. Available at: https://medium.com/fnplus/evaluating-recommender-systems-with-python-code-ae0c370c90e.

Raschka, S. (2018). MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. Journal of Open Source Software, 3(24), 638.

Sahu, S. K., & Sahoo, B. (2021). Product Recommendation System Using Collaborative Filtering and Market Basket Analysis. International Journal of Advanced Science and Technology, 30(3), 3345-3353.

Zhou, Z., Liu, J., & Zhang, G. (2020). An Improved Apriori Algorithm Based on Cluster Analysis. Journal of Physics: Conference Series, 1526(1), 012118. https://doi.org/10.1088/1742-6596/1526/1/012118