**CCT College Dublin**

**Assessment Cover Page**

*To be provided separately as a word doc for students to include with every submission*

|  |  |
| --- | --- |
| **Module Title:** | Data Visualization Techniques, Machine Learning for Business |
| **Assessment Title:** | CA2 |
| **Lecturer Name:** | David McQuaid, Muhammad Iqbal |
| **Student Full Name:** | Caroline Floss Bresolin |
| **Student Number:** | 2023190 |
| **Assessment Due Date:** | 26/05/2024 |
| **Date of Submission:** | 26/05/2024 |

**Declaration**

|  |
| --- |
| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

Table of Contents

[INTRODUCTION 3](#_Toc167653115)

[DATA UNDERSTANDING 3](#_Toc167653116)

[Data Preparation 4](#_Toc167653117)

[Exploratory Data Analysis (EDA) 5](#_Toc167653118)

[Customer 5](#_Toc167653119)

[When the purchases were made? 6](#_Toc167653120)

[Monthly Transactions per Year 6](#_Toc167653121)

[Number of Transactions per Day 6](#_Toc167653122)

[Transaction Counts by Year and Day of the Week 6](#_Toc167653123)

[Days Between Purchases 7](#_Toc167653124)

[Correlation between sales in 2014 and 2015 8](#_Toc167653125)

[Top purchased items 8](#_Toc167653126)

[Top 10 least selling items 9](#_Toc167653127)

[Top 10 selling items 9](#_Toc167653128)

[Top 10 selling items at weekends 10](#_Toc167653129)

[Frequency of the Purchased Items 10](#_Toc167653130)

[Iterative Dashboard Groceries 12](#_Toc167653131)

[Market Basket Analyses 12](#_Toc167653132)

[APRIORI: 13](#_Toc167653133)

[FP-growth algorithm / Market Basket Analysis 16](#_Toc167653134)

[Compare Apriori vs Fp\_growth 17](#_Toc167653135)

[Content and Collaborative filtering / Grocery Store 19](#_Toc167653136)

[Create User-Based Collaborative Filtering 19](#_Toc167653137)

[Content-Based 20](#_Toc167653138)

[Movies Recommender 22](#_Toc167653139)

[CONTENT BASED RECOMMENDER 23](#_Toc167653140)

[Item-Item 23](#_Toc167653141)

[COLLABORATIVE FILTERING 25](#_Toc167653142)

[Summary 27](#_Toc167653143)

[References 29](#_Toc167653144)

# INTRODUCTION

Implementing machine learning techniques, particularly recommendation systems, has become essential for enhancing customer experience and driving sales. A recommendation system predicts and suggests products to users based on various data-driven insights, helping companies personalize customer experience and increase user engagement. This project aims to explore the development and application of a recommendation system using both Content-Based and Collaborative Filtering methods, utilizing a movie dataset from MovieLens (Harper and Konstan, 2015).

Additionally, to further enhance customer personalization and engagement, I will perform Market Basket Analysis using the Apriori and FP Growth algorithms on the Groceries dataset, sourced from Kaggle (www.kaggle.com, n.d.). This analysis will highlight the similarities and differences between these algorithms, allowing us to compare the machine-learning results obtained from both methods.

Moreover, we were tasked to design an interactive dashboard that summarizes the key aspects and identifies through its visualisation why this dataset is suitable for Machine Learning models in an online retail business. Regards it, I decided to use the Groceries dataset.

Lastly, I will discuss the rationale and justification for all stages of data preparation necessary for the visualizations, ensuring clarity and effectiveness in conveying the most significant insights from the data. Through this comprehensive approach, this project aims to demonstrate the potential of machine learning in enhancing online retail operations and user satisfaction.

# DATA UNDERSTANDING

Groceries dataset

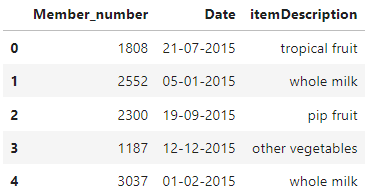
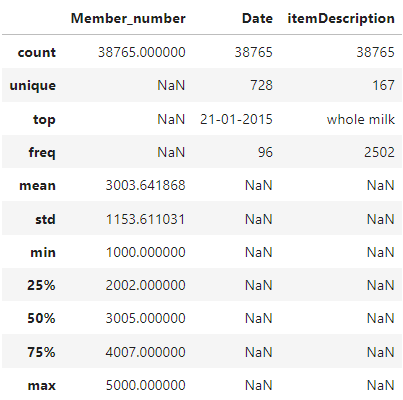
The dataset has 38765 rows of the purchase orders of people from a grocery store from 2014 to 2015.

Initially, it had three columns:

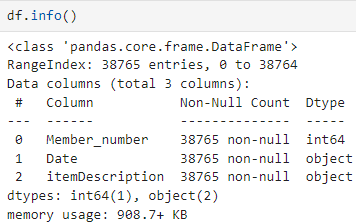
* Member\_number: Represents the customer number,
* Date: The date when the transaction was made,
* Item Description.

The first image below provides a brief overview of the dataset using the “.head” function. Besides, statistical details are displayed using the “describe” function. By that, we can notice that “whole milk” is the top purchased item and in

21/05/2015 was the pick of purchases.

This dataset doesn’t contain any null values.

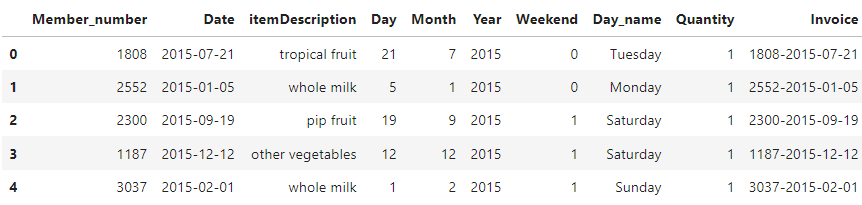


# Data Preparation

In order to analyse the dataset further, gain meaningful insights and employ Market Basket Analyses, some steps included:

* Transforming the feature “Date” into date format,
* Adding three new features for “Month”, “Day” and “Year”,
* Whether the purchase was made at the weekend or not,
* Day name (Monday, Tuesday…),
* Quantity,
* Invoice's values combine values from columns “Member\_number” and “Date”.

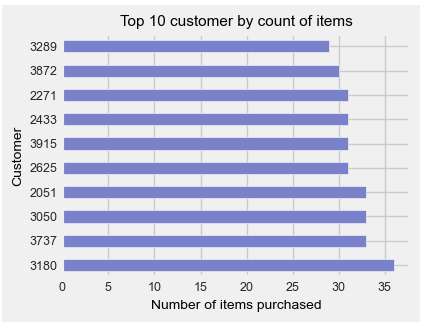
So far, the adjusted dataset looks like this:



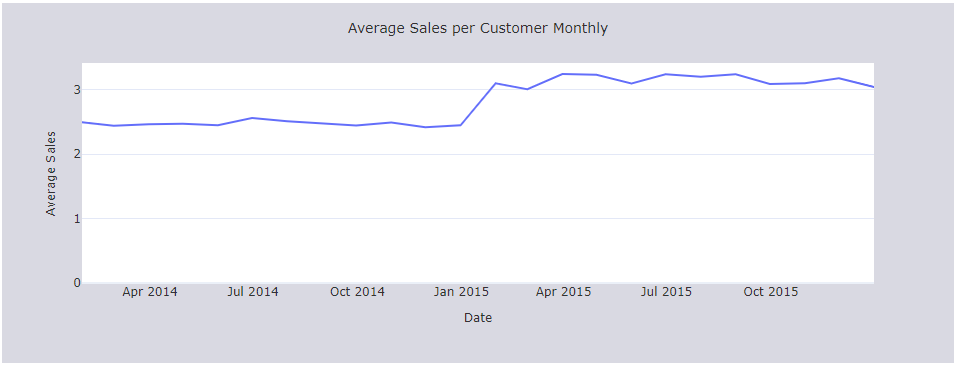
# Exploratory Data Analysis (EDA)

## Customer

Top 10 Customers based on the count of purchased items:



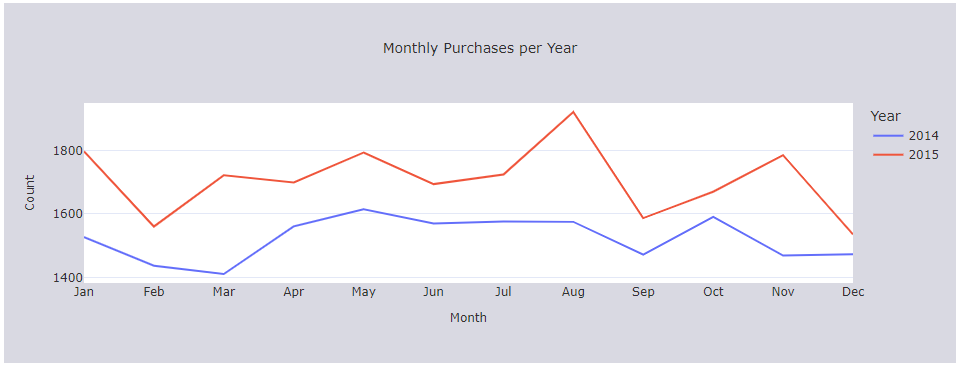
Average Purchases per Customer Monthly



## When the purchases were made?

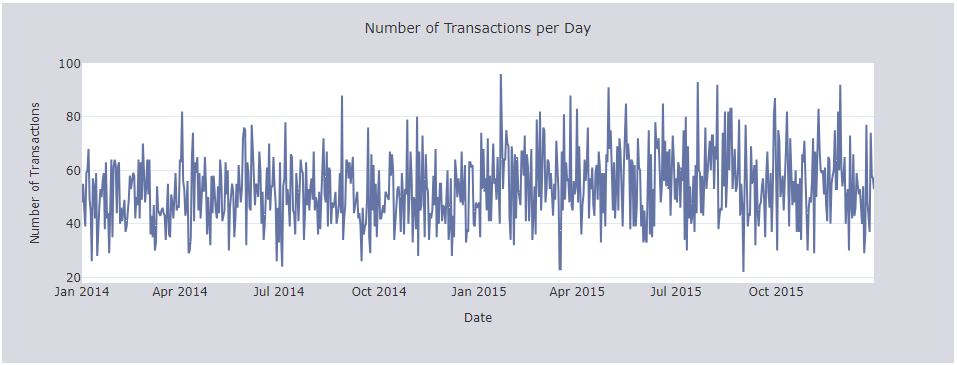
### Monthly Transactions per Year

By that, we can see if there is any trend over the months between the two years.



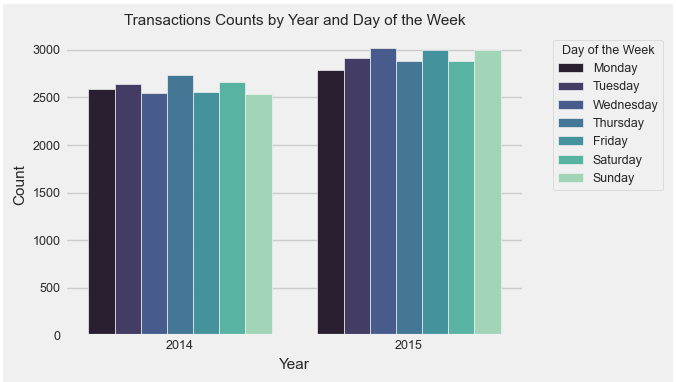
### Number of Transactions per Day

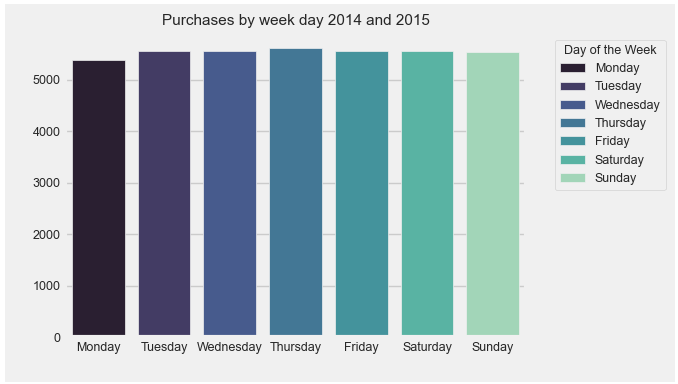
There is no pattern among the days, but, notably, on some days there are more items purchased than others.



### Transaction Counts by Year and Day of the Week

Comparing 2014 with 2015, it seems that customer behaviour may have changed, for instance, while in 2014 most of the purchase items were made on Thursdays, in 2015 it was on Wednesdays.





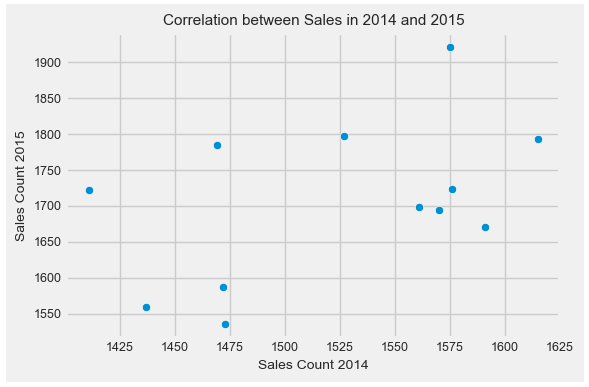
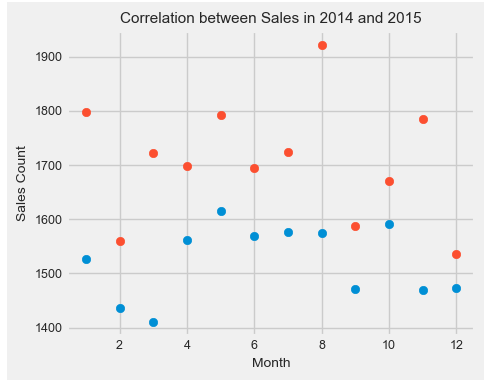
### Days Between Purchases

A new dataset is created, firstly grouping the “Member\_number” and calculating the difference of the days among the “Date”, then selecting only those higher than zero because those with zero means they have purchased just once. It is an important detail because we can see how often the customer usually buys in that particular Grocery Store.



### Correlation between sales in 2014 and 2015

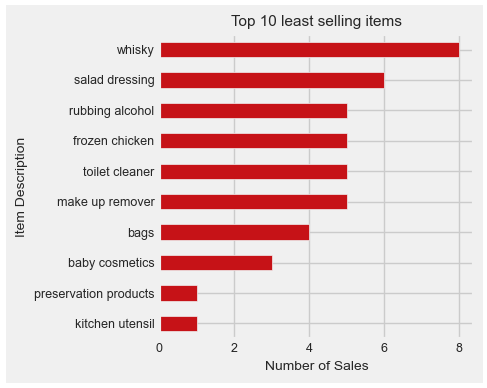
Employing the correlation function, it achieves a Correlation between sales in 2014 and 2015 of 0.4654. It suggests there is a partial correlation between both of the years. The scatterplot below shows the count of transactions per month between the years. At some months the count is similar, but considering that 2015 has a higher number of transactions, it is normal to have some differences.

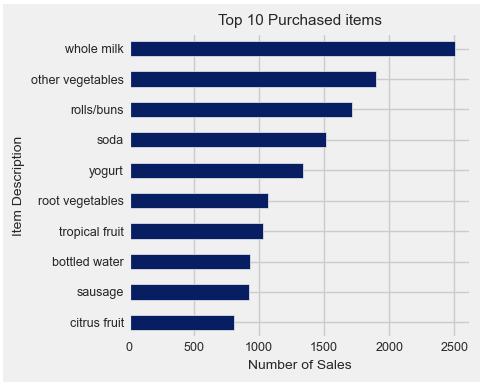
## Top purchased items

Analyzing the top 10 purchased items in a grocery store is crucial for understanding customer preferences, generating revenue, managing inventory effectively, planning marketing strategies, and staying competitive in the market. By focusing on these top-selling items, grocery stores can optimize operations and enhance the overall shopping experience for customers.

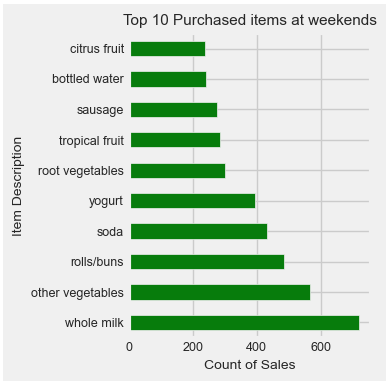
### Top 10 least selling items



### Top 10 selling items



### Top 10 selling items at weekends



### Frequency of the Purchased Items

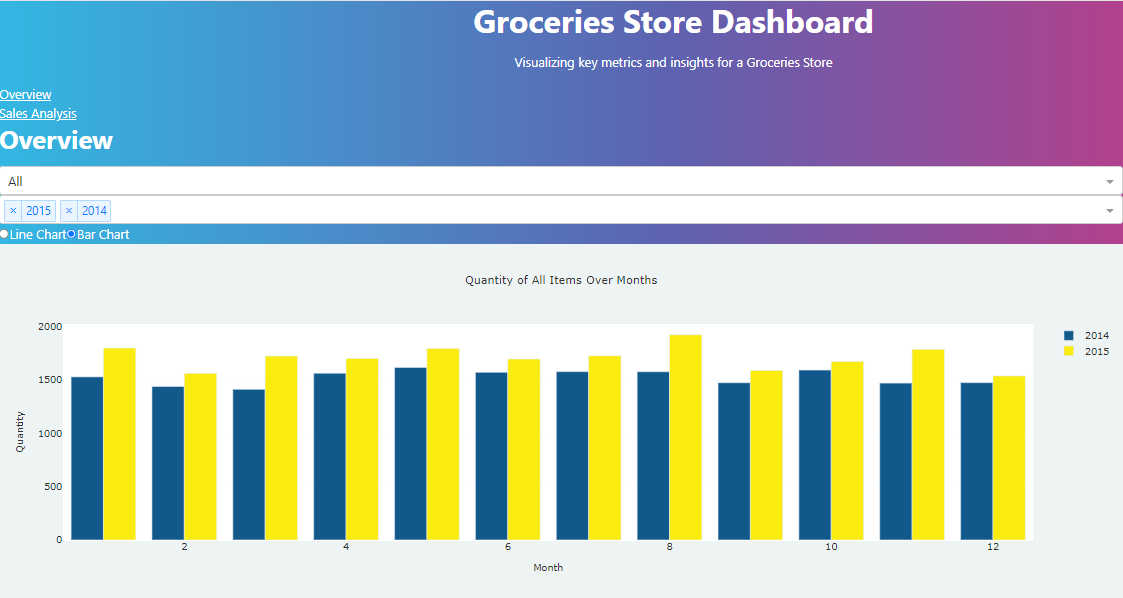
A *treemap* is particularly useful for visualizing the frequency of items purchased because it clearly represents items in a visual format, with each item's box size proportional to its purchase frequency.



In summary, Exploratory data analysis (EDA) revealed insights into customer behaviour, transaction trends, and item popularity. Key findings include the identification of top and least-selling items, trends in weekend purchases, and the frequency of purchased items. Analysing the top 10 purchased items proved instrumental in understanding customer preferences, optimizing revenue generation, and informing inventory management strategies. Additionally, visualizations such as treemaps effectively represented item frequency, aiding in data interpretation. Overall, the analysis offers valuable insights for strategic decision-making in the grocery store business.

# Iterative Dashboard Groceries

As required an interactive dashboard was created. The interface is intuitive and user-friendly, making it easy for young adults to navigate and interact with the various features of the dashboard. Dropdown menus and radio buttons allow for quick selection and customization of data views. This interactivity encourages engagement and allows users to explore the data according to their preferences. For more information, please access the dashboard file.



<http://127.0.0.1:8050/>

# Market Basket Analyses

Market basket analysis (MBA) is one such top retail application of machine learning. It helps retailers know what products people are purchasing together so that the store/website layout can be designed in the same manner.

Terminologies used in market basket analysis:

**Itemset**: It refers to the set of items that are purchased together by a customer at the same time. By default, we state it as a logical rule with IF and THEN. For instance, IF or ‘**Antecedent’** (Bread, Butter), THEN or ‘**Consequent’** (Milk).

**Support count:** It is the frequency of a particular item set appearing in the transaction database. It is also stated as a probability. For instance, if milk has a support count of 50 out of a possible 500 transactions, then the probability is 50/500 or 0.1.

**Confidence**: It refers to the conditional probability that represents what items have a possibility of being purchased together.

Algorithms used in market basket analysis

There are several options to employ MBA, in this study, I will focus on two of them, Apriori and FP growth algorithms.

## APRIORI:

The Apriori algorithm works in two steps that are:

* It identifies the itemsets systematically that occur frequently in the dataset and support greater than the pre-specified threshold value.
* Next, it calculates the confidence of all possible rules. However, it only keeps those items states that have confidence greater than a pre-specified threshold.

It is further classified into three components.

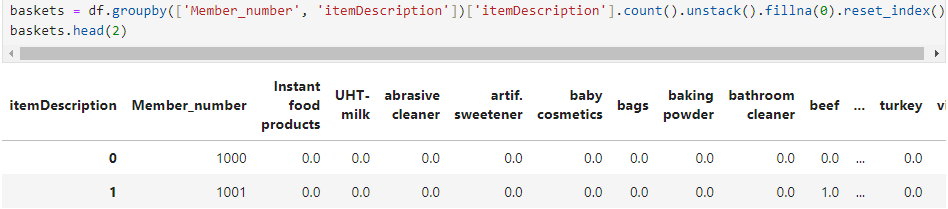
**Support**: Identifies frequent itemsets with support greater than the threshold.

**Lift:** It is the ratio of the confidence percent to the support percent.

**Confidence:** It is the ratio of combined transactions to individual transactions.

**Implementing the Apriori Algorithm:**

The first step is to create a basket. The chosen approach depends on the dataset and its properties. I decided to use one of the most common that groups the “Member\_number” and the “item\_description” is moved into columns where the values are the counted items purchased by each customer.



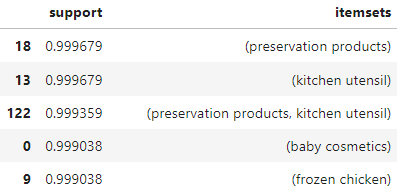
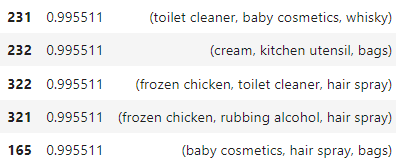
Then it is encoded using One Hot Encoder where True is signed when “1” and False otherwise.



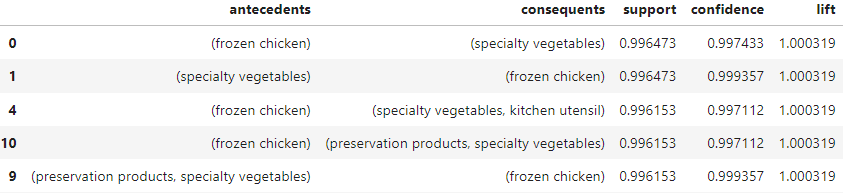
In order to evaluate the model, the dataset is divided into train and test.

The “min\_support” is settled with 0.99 and the metric is “lift” because it measures the ratio of the confidence to the support.

Displaying the “frequent\_itemsets” showcase the frequent items purchased together:

Displaying the “rules” it shows that if a customer purchases *frozen chicken* they are likely to purchase *specialty vegetables*. The three components suggest strong metrics:



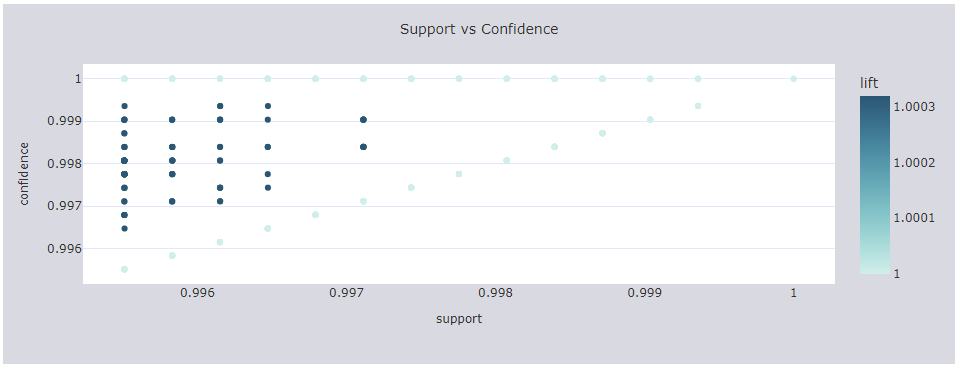
Association rules for each split in cross-validation:



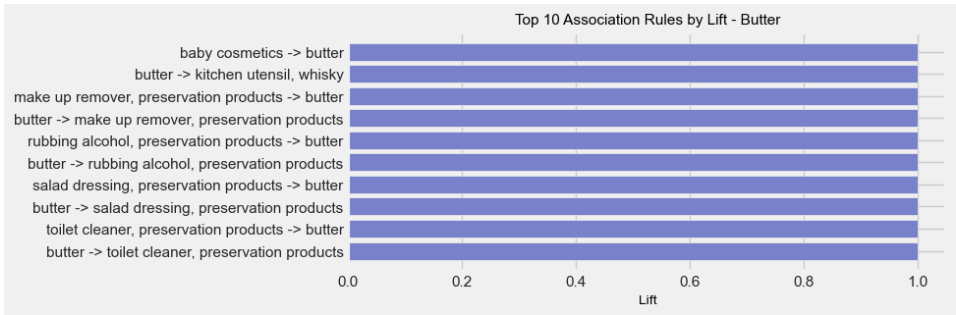
Employing cross-validation we can test the accuracy of the rules on the test data:



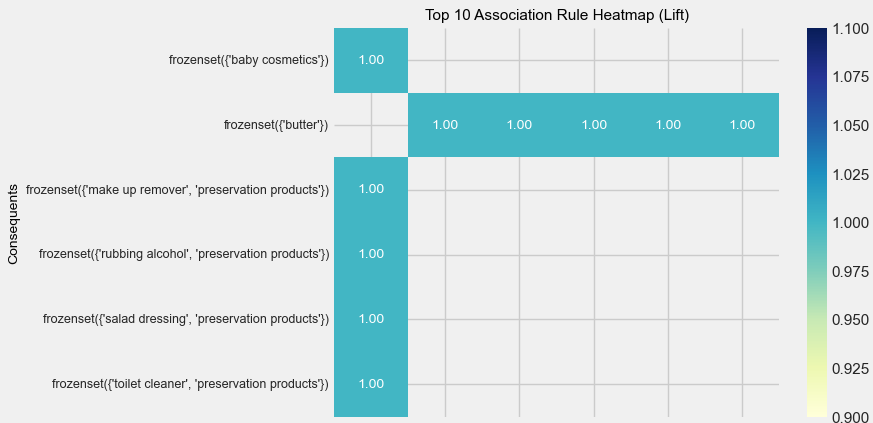
Visualising the outcomes:

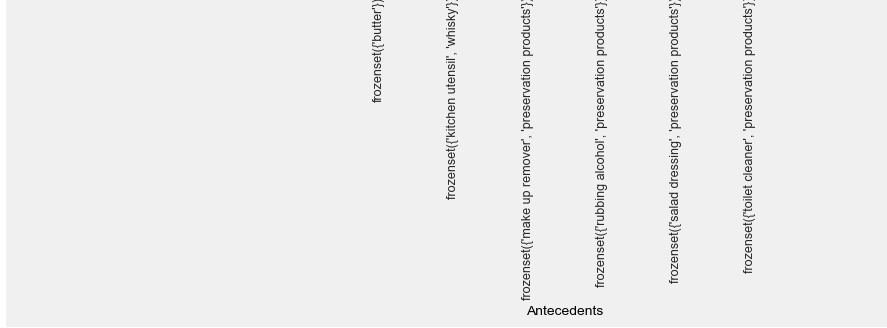


Picking a different item we can analysing the top 10 Association Rules by lift:



A heatmap is a good approach to visualize the association rules between antecedents and consequents. The displayed values correspond to the lift:



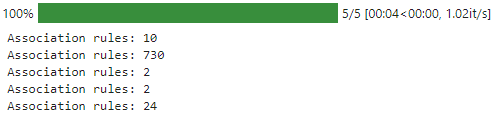


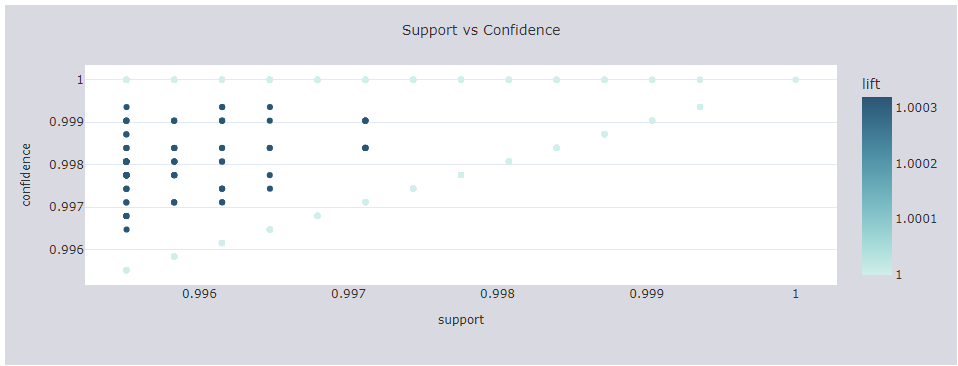
## FP-growth algorithm / Market Basket Analysis

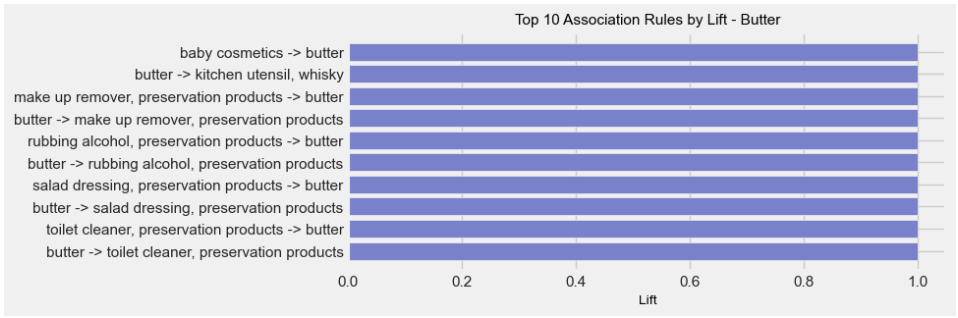
The FP Growth algorithm is a popular method for frequent pattern mining in data mining. It works by constructing a frequent pattern tree (FP-tree) from the input dataset. The FP-tree is a compressed representation of the dataset that captures the frequency and association information of the items in the data (Jodha, 2023).

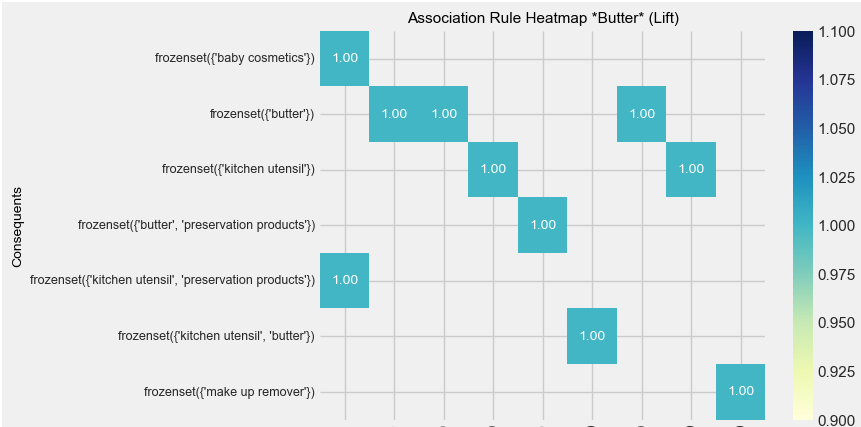
The FP Growth algorithm in data mining has several advantages over other frequent pattern mining algorithms, such as Apriori. The Apriori algorithm is not suitable for handling large datasets because it generates a large number of candidates and requires multiple scans of the database to my frequent items. In comparison, the FP Growth algorithm requires only a single scan of the data and a small amount of memory to construct the FP tree.

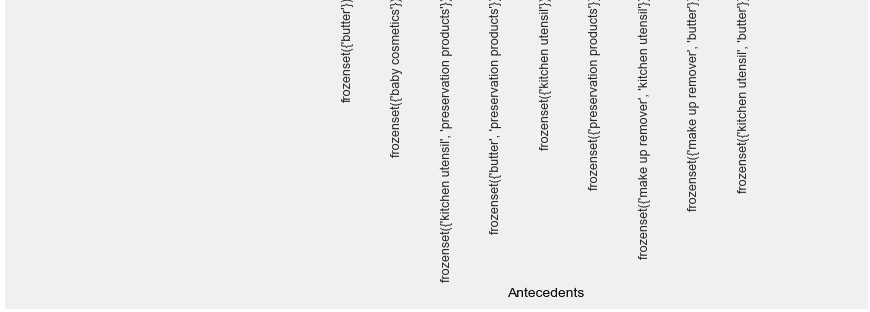
To employ FP Growth I used the same approach, adjusting only the algorithm to FPgrowth. It achieved the same average accuracy but in less time consuming:





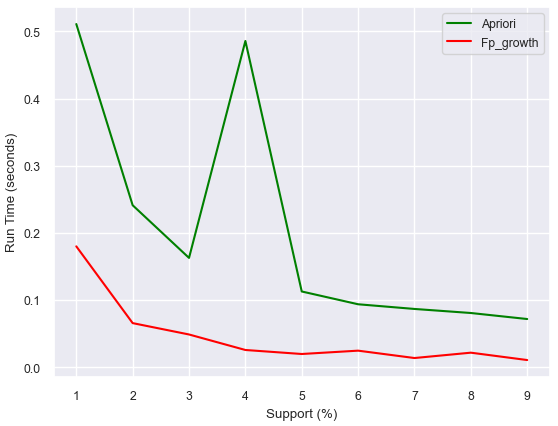




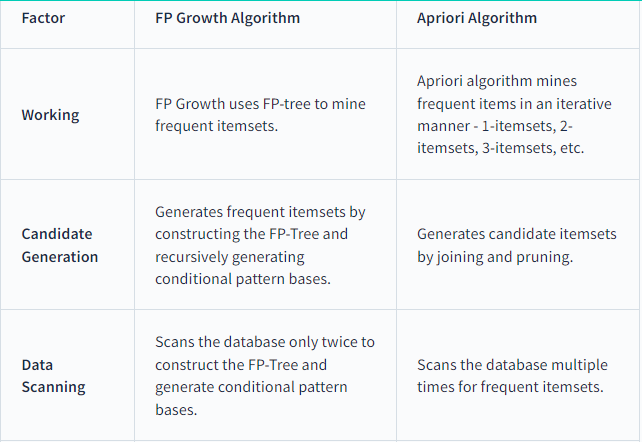
## Compare Apriori vs Fp\_growth

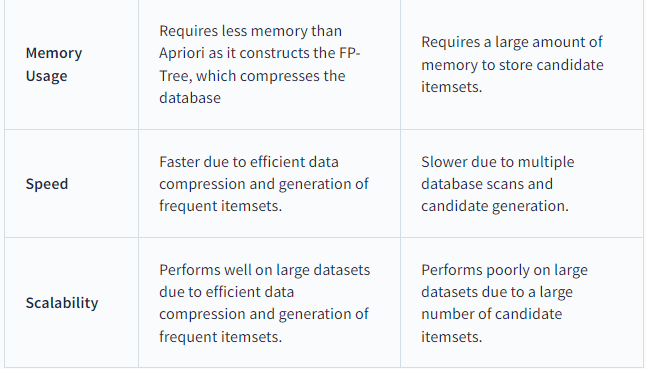
Despite both algorithms achieving high average accuracy, a perfect score can sometimes indicate overfitting, where the model has learned the training data too well, including the noise and exceptions. This can reduce its ability to generalise to new, unseen data. One way to solve it is using cross-validation, which was done. Further analyses would be essential to investigate it.

Analysing the run time in seconds vs Support within both algorithms is notable that FP-Growth is less time-consuming. It is essential when dealing with large datasets.

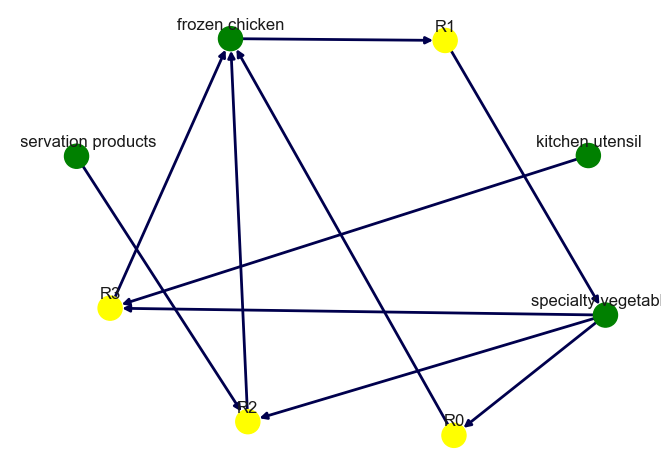
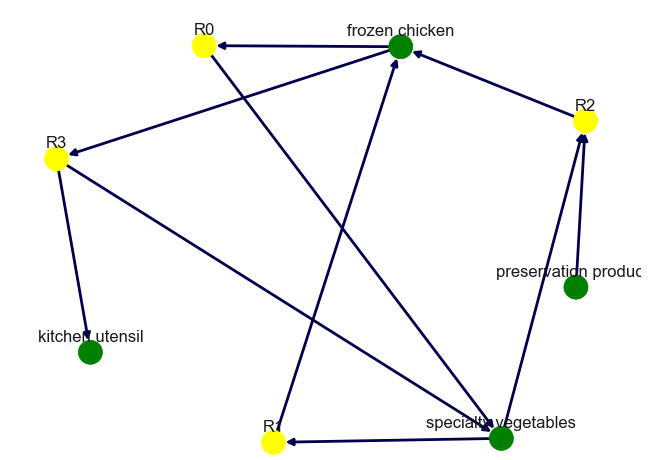


The main differences between both:





The FP-Growth and Apriori algorithms produce similar rules in the first plot, indicating that customers who buy "frozen chicken" are likely also to buy "preservation products." However, the high score metrics, nearly to one, suggest an unrealistic certainty in the association rules, potentially compromising the reliability of the recommendations. Thus, despite high accuracy, it's crucial to understand the algorithm outcomes and seek ways for improvement.



The image above displays the rule nodes ('R0', 'R1', etc.) representing specific association rules, while item nodes (green) represent the items involved in those rules, for example, if a customer purchases *frozen chicken* it brings to rule 1 (R1) that is *specialty vegetables*.

## Content and Collaborative filtering / Grocery Store

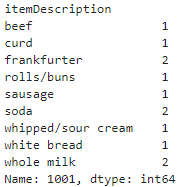
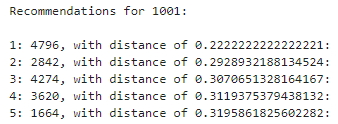
### Create User-Based Collaborative Filtering

This project aims to build a Grocery Recommendation System that will provide suggestions to the customers about the relevant products they might want to buy next. There are two general types of collaborative filtering:

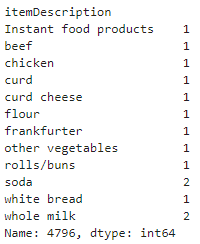
User to user and Item to item. In the Grocery content, only *User to user* will be employed.

I generated user carts by listing the products each user has purchased. This data is represented as a sparse matrix (csr\_matrix), where each row represents a user and each column represents an item. The matrix entries indicate whether an item was purchased by a user (1 if purchased, 0 if not). Using the NearestNeighbors algorithm, I then find similar users based on this matrix.

Testing the model, I selected the user 1001. Below the first image are the items which they have purchased, and besides, its recommendations:

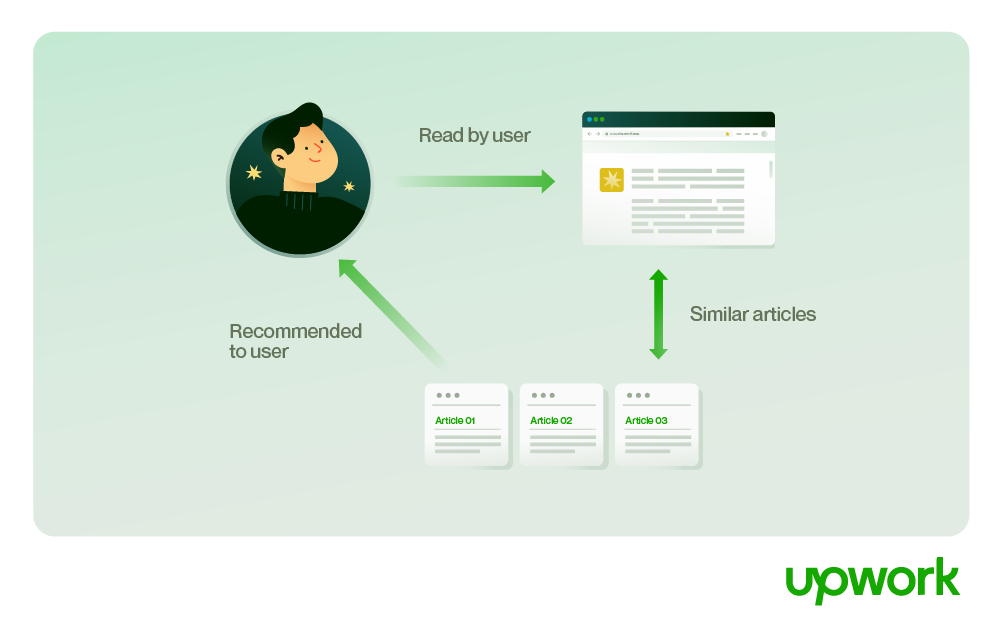
Displaying the first user recommended is notable the similarities in their purchase, both of them picked beef, curd and two whole milks.



### Content-Based

Imagine a digital world that understands your preferences better than you do. This is the essence of content-based filtering, a sophisticated aspect of artificial intelligence (AI) and machine learning (ML).

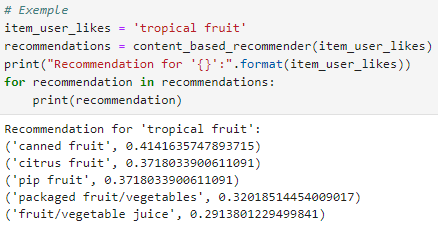
It basically takes into account the user’s activities and then makes personalised suggestions. The image below provides a clear illustration of it.



In the context of grocery stores, content-based filtering is particularly useful for online shopping. The use of online grocery services has substantially increased over the years.

To employ this approach, I used the Term frequency Inverse document frequency (TFIDF) in the “item\_description” which is a statistical formula to convert text documents into vectors based on the relevancy of the word, containing the information about less relevant and most relevant words in the document coupled with Cosine similarity. It measures the similarity between two non-zero vectors(TFIDF transformed earlier) in an inner product space by the cosine of the angle between them, yielding a value from -1 to 1: -1 indicates opposite vectors, 0 indicates orthogonal vectors, and 1 indicates similar vectors.

Using the trained model, I chose "tropical fruit" to showcase the recommended items along with their cosine scores:



It suggested items similar to fruit, with canned fruit being the most similar, achieving a cosine score of 0.41.

By combining User-Based Collaborative Filtering and Content-Based Filtering, the Grocery Recommendation System can effectively enhance customer experience by providing personalized and relevant product recommendations.

# Movies Recommender

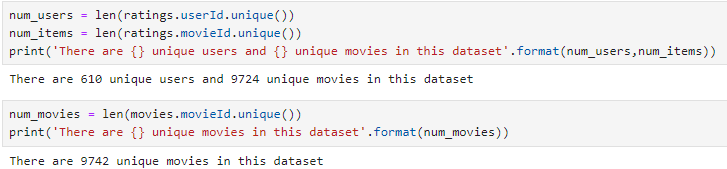
To delve further into the recommendation system, I utilized an additional dataset obtained from MovieLens.

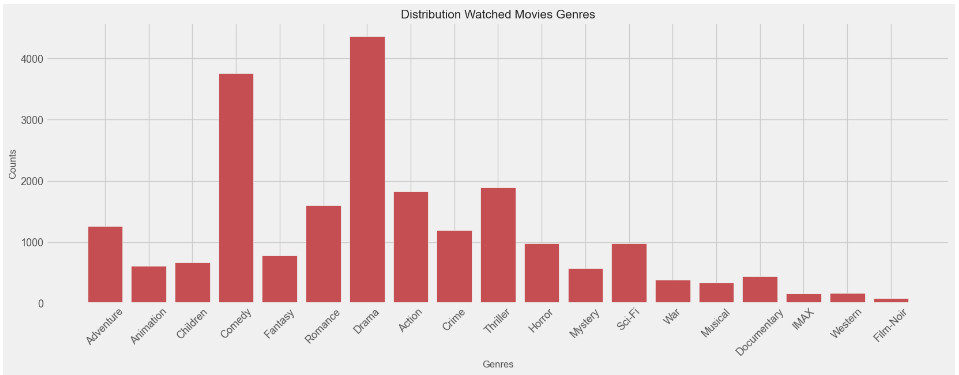
This dataset (ml-latest-small) describes 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service. It contains 100836 ratings and 3683 tag applications across 9742 movies. These data were created by 610 users between March 29, 1996 and September 24, 2018. This dataset was generated on September 26, 2018.

Three files are used in this project:

* Movies: Contains “'movieId'”, “title” and “genres”,
* Ratings: Contains 'userId', 'movieId' and 'rating',
* Tags: “userId”, “movieId”, “tag”, “timestamp”.

The dataset doesn’t contain any null values.



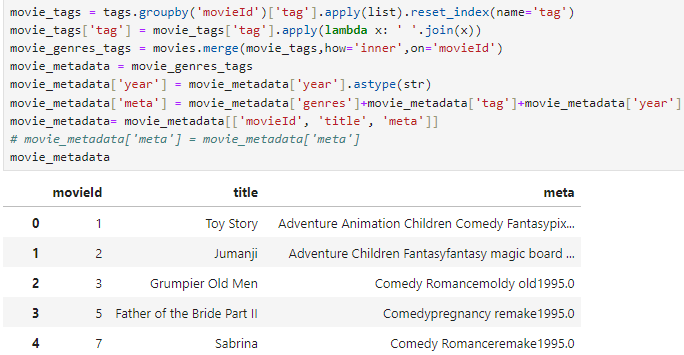


## CONTENT BASED RECOMMENDER

### Item-Item

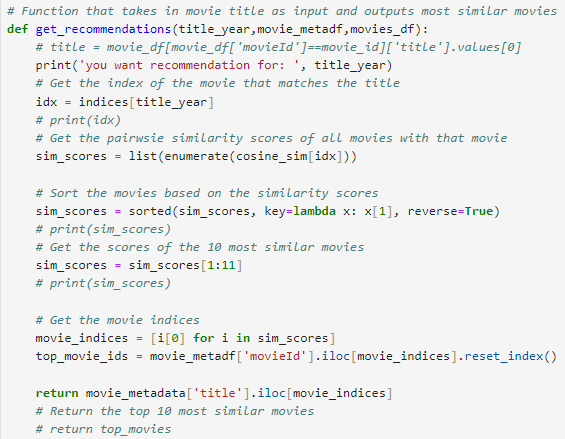
In the context of movies, this approach relies on movies watched by a user.

Initially, I constructed the dataset to be used. It incorporates a new attribute named "meta," encompassing the genre, year, and tag. I retain the "movieId" and the “title” as well.

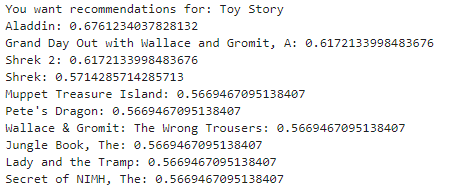


To transform the words into vectors I used CountVectorizer. It works similarly to TFIDF but is less time-consuming due to its process. It converts text into fixed-length vectors by counting how many times each word appears. The tokens are then stored as a bag-of-words. After that, the cosine similarity is calculated.

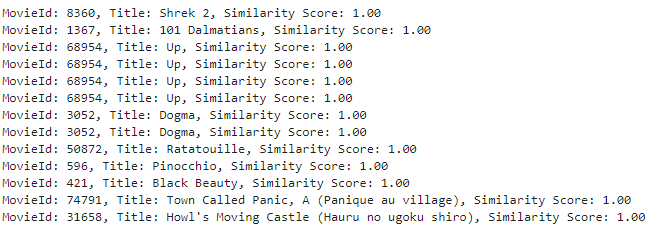
Getting recommendations:



With a trained model, we can test the recommendations. Selecting “Toy Story” it suggests movies in the same genre:



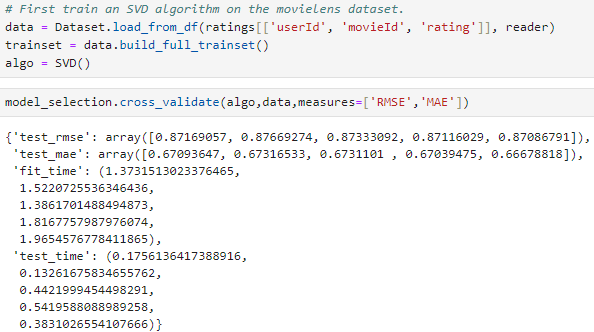
Alternatively, I have employed the model using Word2Vec, which is a different approach to converts text into vectors. As a result, it displays similar movies, but with a higher cosine similarity score:



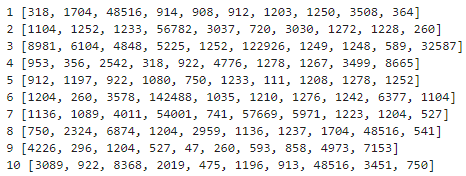
Content-based recommenders, while useful, have their limitations. They struggle to capture intricate interdependencies or nuanced preferences, highlighting the importance of exploring various approaches and comprehending the recommendation process. For instance, I might prefer adventure movies that incorporate historical elements, rather than those that focus solely on action. Such nuances often evade the capabilities of these recommenders.

## COLLABORATIVE FILTERING

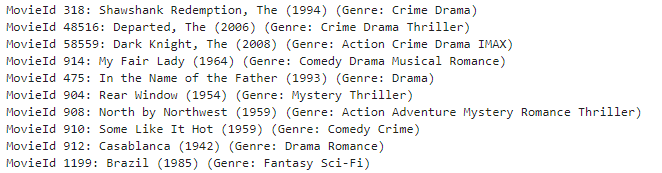
In a more general sense, collaborative filtering is the process of filtering for information among users. These models analyse user preferences and recommend content based on similar users' behaviour. For example, if User A enjoys the same shows as User B, the algorithm suggests shows that User B liked but User A hasn't seen yet.



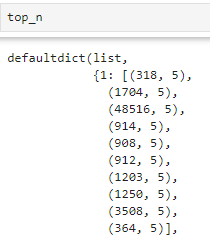
Based on the model, it displays the 10 suggested movies for the 10 first users:



For user 1, here are the suggested movies along with their respective genres:

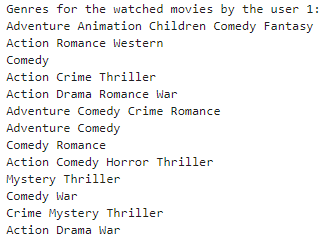


Below we can visualise the rating estimation for each suggested movie by the recommender system:



Considering an RMSE of 0.87, it indicates that the model is providing reasonable predictions.

To validate the relevance of the suggested movies, I opted to examine a few movies previously watched by user 1. From the visual analysis, it appears that the recommender system indeed picked movies with comparable genres.



## Summary

This project focused on developing a movie recommendation system using collaborative filtering and content-based approaches. Content-based filtering may be more suitable for recommending products with distinct attributes, while collaborative filtering may be more effective for identifying patterns in user behavior and preferences. By carefully considering the strengths and limitations of each approach, businesses can implement recommendation systems that effectively meet the needs and preferences of their users. Leveraging the MovieLens dataset, containing user ratings and tags for thousands of movies, the system provided personalized movie recommendations based on user preferences and movie attributes. Content-based methods utilized vectorization techniques like CountVectorizer and Word2Vec to analyze movie metadata and suggest similar movies. Collaborative filtering employed Singular Value Decomposition (SVD) to identify similar users and recommend movies based on their preferences. Through rigorous testing and validation, the system demonstrated its ability to provide relevant and accurate movie recommendations, enhancing user experience and engagement in movie selection.

<https://github.com/CCT-Dublin/integrated-ca2-dvt-and-mlb-Caroline2023190>

|  |  |  |
| --- | --- | --- |
|  | Citation | Written |
| Introduction |  | 215 |
| Data Understanding |  | 91 |
| Data Preparation |  | 66 |
| EDA |  | 352 |
| Market Basket Analyses |  | 164 |
| Apriori |  | 293 |
| FP-growth algorithm | 50 | 149 |
| Compare Apriori vs Fp\_growth |  | 198 |
| Content and Collaborative filtering |  | 390 |
| Grocery Store |  |  |
| Movies Recommender |  | 96 |
| CONTENT BASED RECOMMENDER |  | 202 |
| COLLABORATIVE FILTERING |  | 146 |
| Summary |  | 153 |
|  |  |  |
|  |  | 2515 |

## References

Abhinav Ajitsaria (2019). *Build a Recommendation Engine With Collaborative Filtering*. [online] Realpython.com. Available at: https://realpython.com/build-recommendation-engine-collaborative-filtering/.

Chaudhary, S. (n.d.). *Understanding Market Basket Analysis in Data Mining*. [online] www.turing.com. Available at: https://www.turing.com/kb/market-basket-analysis.

Code With Prince (2021). *Web Data Dashboard with Plotly express and Flask Python and JavaScript*. [online] YouTube. Available at: https://www.youtube.com/watch?v=B97qWOUvlnU [Accessed 24 May 2024].

dash-bootstrap-components.opensource.faculty.ai. (n.d.). *Theme explorer - dbc docs*. [online] Available at: https://dash-bootstrap-components.opensource.faculty.ai/docs/themes/explorer/ [Accessed 23 May 2024].

dash.plotly.com. (n.d.). *Dropdown | Dash for Python Documentation | Plotly*. [online] Available at: https://dash.plotly.com/dash-core-components/dropdown.

GitHub. (n.d.). *Content\_based\_movie\_recommender/EG\_project5.ipynb at master · emmagrimaldi/Content\_based\_movie\_recommender*. [online] Available at: https://github.com/emmagrimaldi/Content\_based\_movie\_recommender/blob/master/EG\_project5.ipynb [Accessed 15 May 2024].

GroupLens. (2015). *MovieLens Latest Datasets*. [online] Available at: https://grouplens.org/datasets/movielens/latest/.

Grouplens.org. (2018). Available at: https://files.grouplens.org/datasets/movielens/ml-latest-small-README.html.

Harper, F.M. and Konstan, J.A. (2015). The MovieLens Datasets. *ACM Transactions on Interactive Intelligent Systems*, 5(4), pp.1–19. doi:https://doi.org/10.1145/2827872.

Jodha, R. (2023). *FP Growth Algorithm in Data Mining*. [online] Scaler Topics. Available at: https://www.scaler.com/topics/data-mining-tutorial/fp-growth-in-data-mining/.

plotly.com. (n.d.). *Discrete Colors*. [online] Available at: https://plotly.com/python/discrete-color/.

plotly.com. (n.d.). *Setting the Font, Title, Legend Entries, and Axis Titles*. [online] Available at: https://plotly.com/python/figure-labels/.

plotlygraphs (2019a). *Line Charts*. [online] plotly.com. Available at: https://plotly.com/python/line-charts/.

plotlygraphs (2019b). *Setting Graph Size*. [online] plotly.com. Available at: https://plotly.com/python/setting-graph-size/.

PYTHON CHARTS | The definitive Python data visualization site. (2022). *Matplotlib style sheets*. [online] Available at: https://python-charts.com/matplotlib/styles/?utm\_content=cmp-true [Accessed 25 May 2024].

seaborn.pydata.org. (n.d.). *seaborn.barplot — seaborn 0.11.1 documentation*. [online] Available at: https://seaborn.pydata.org/generated/seaborn.barplot.html.

The Upwork Team (2021). *What Content-Based Filtering is & Why You Should Use it | Upwork*. [online] www.upwork.com. Available at: https://www.upwork.com/resources/what-is-content-based-filtering.

www.kaggle.com. (n.d.). *Groceries dataset*. [online] Available at: https://www.kaggle.com/datasets/heeraldedhia/groceries-dataset.