# Department of Computing CS-404 Big Data Analytics

**Class: BESE-11 & BSCS-11**

**Spring 2024**

**Lab Manual 03: Mining Frequent Itemset [Rapid Miner & Python]**

**Date: 20-02-2024**

**Time: 10:00-12:50 & 14:00-16:50**

**Instructor:** Dr. Syed Imran Ali & Dr. Muhammad Daud Abdullah Asif

**Lab Engineer:** Engr. Masabah Bint E Islam

**Lab : 03 :** **Mining Frequent Itemset [Rapid Miner & Python]**

**Aim:** This lab is designed to explore the mining of frequent itemset in market basket analysis, utilizing RapidMiner and Python as the primary tools. The focus will be on the understanding and application of various algorithms implementation for mining frequent items, itemsets and association rules through pipeline process and programming on large datasets. Mastery of these techniques is vital as they significantly impact the accuracy and efficiency of subsequent big data analytics tasks.

**Objective:** The objectives of this lab are to equip you with a comprehensive understanding and practical skills in the following areas:

* Introduction to association rule mining algorithms.
* Application of apriori and fp-growth algorithm through Rapid Miner Design Process.
* Coding for rule mining algorithm in python programming language.
* Implementation of algorithms for to be used in subsequent analysis of big data analysis

**Tools/Software:** Rapid Miner, IDE for python (i.e. Google Colab, Pycharm etc)

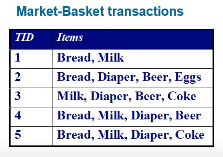
**Deliverables:** Submit a single file on LMS before the due date as communicated by Lab Engineer.

Note: Please ensure your own work, add screenshots from each step/ activity properly and submit in a Word / PDF Report Lab Report.

# Market Basket Analysis

Identification of frequently purchased items/services OR activities from given transactional/behavioral data is called Market Basket Analysis. This analysis is not limited to products. The basket analysis is also useful to explore the visited pages of a website, the questions raised in support of tickets, and even the dishes ordered in a restaurant. Given a set of transactions, where each transaction contains a set of items. The goal is to extract a set of rules that represent the interesting relationships among items.

Example:



A black text on a white background

Description automatically generated A white background with black text

Description automatically generated

**Algorithms for Market Basket Analysis – Mining Frequent Item-sets**

**Apriori Algorithm**

Market Basket analysis uses “Apriori” algorithm to efficiently search large databases of transactional data for association rules. The main measures in this algorithm are support, confidence, and lift. Support indicates the frequency of the products being sold together. A high support means that the basket analysis statistics are more reliable. Additionally, higher support makes the basket of products more interesting financially. Confidence indicates the direction for cross-selling. Lift indicates the strength of the relationship between the products. For example, item 1 and item 2 appear together 90% of transaction than support is 90/100 equals 0.9. Similar, % of transactions that contain two products containing one of the two products gives us confidence measure and lift measures the strength of a rule over the random occurrence of the itemset, helping in understanding the association strength between items.

**FP-Growth Algorithm**

The FP-Growth (Frequent Pattern Growth) algorithm is an efficient and scalable method used in market basket analysis for identifying frequent item sets in transactional databases without generating candidate itemsets, unlike the Apriori algorithm. This algorithm works in two steps: (1) construction of the FP-tree (Frequent Pattern tree) which captures the frequency of itemsets within the transaction database in a compact structure, and (2) extracting frequent itemsets directly from the FP-tree. Key measures in FP-Growth include support, confidence, and lift, similar to the Apriori algorithm. Support in FP-Growth indicates the frequency of the itemsets appearing in the database; it helps in identifying the most common itemsets. Confidence measures the likelihood of occurrence of an item Y given the presence of another item X in the transactions. Lift measures the strength of a rule over the random occurrence of the itemset, helping in understanding the association strength between items.

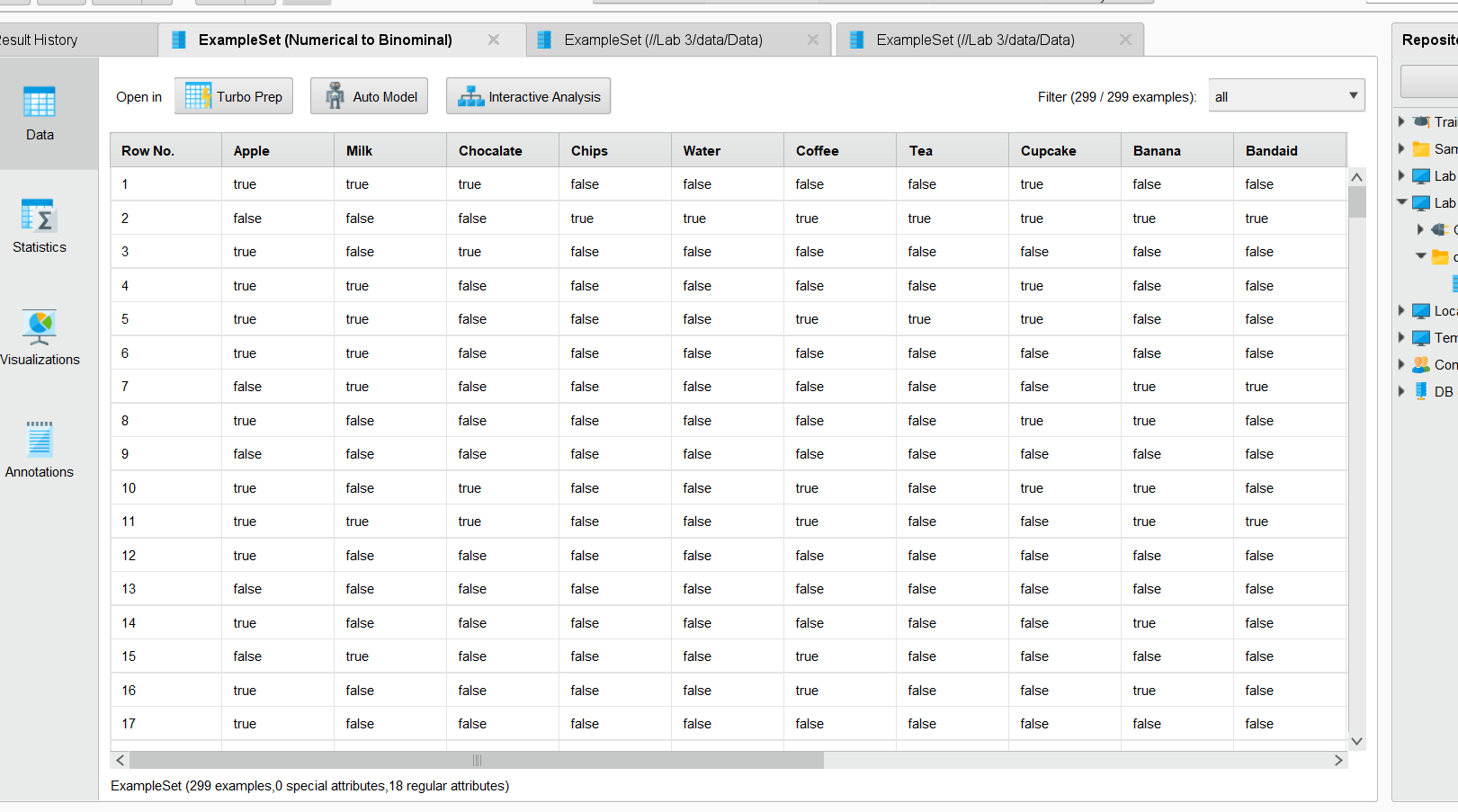
For example, if item A and item B appear together in 50 out of 100 transactions, the support for the itemset {A, B} is 50/100 = 0.5. If item A appears in 60 transactions, the confidence of the rule {A} => {B} (the likelihood that B is purchased when A is purchased) is 50/60 = 0.83. Lift would then be calculated based on the support of A and B together divided by the individual supports of A and B, indicating the strength of the association compared to random chance.

The FP-Growth algorithm greatly reduces the complexity of mining frequent itemsets by using an FP-tree to store the database in a compressed form and by mining the tree directly. This avoids the costly step of candidate generation and testing used by the Apriori algorithm, making FP-Growth suitable for large datasets.

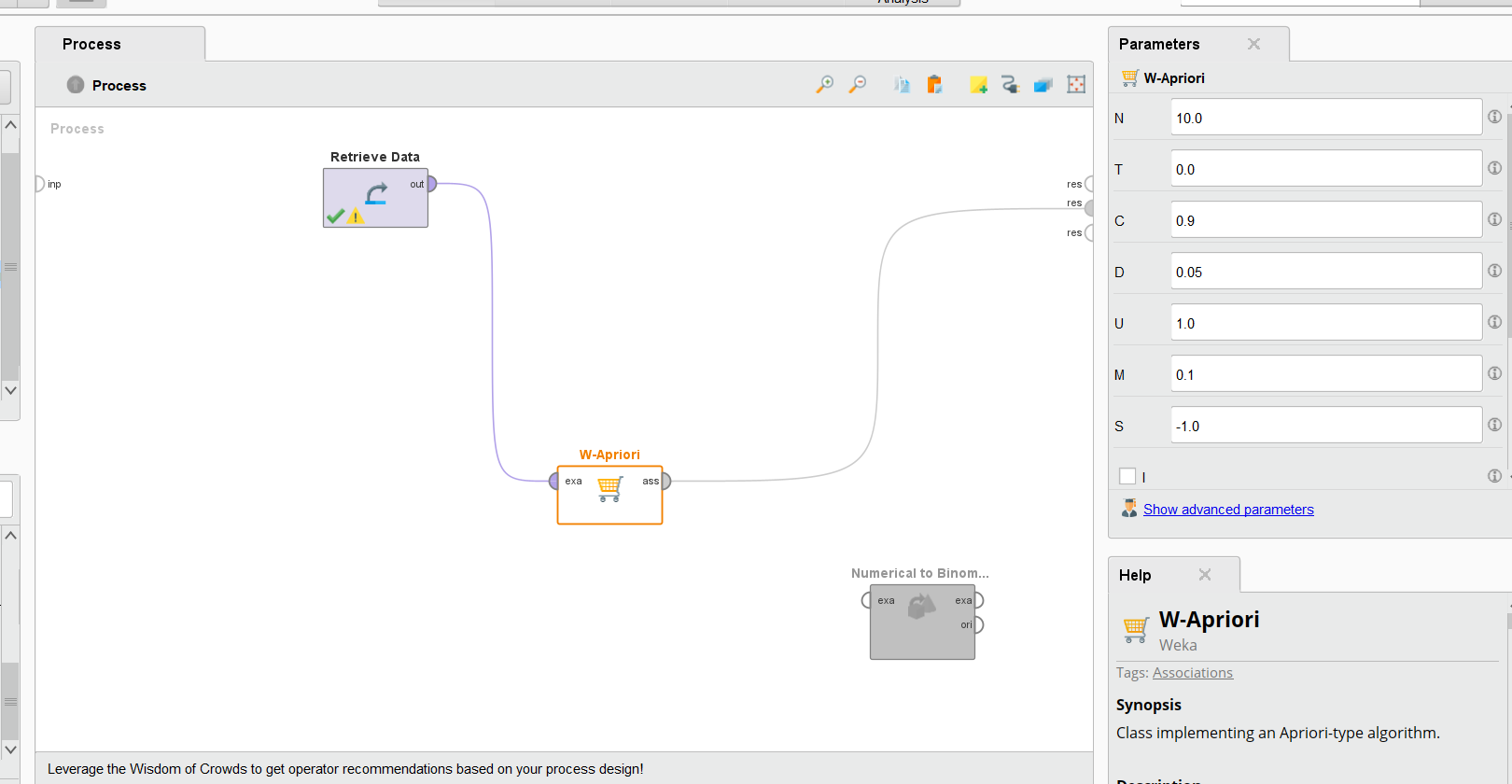
Tasks

# Task 1: Apriori Algorithm in Rapid Miner

1. Open the Rapid Miner Studio and start a Blank Process.
2. The apriori algorithm is not existing in preinstalled operators therefore we have to install an extension for this. In the top bar click on Extensions > Marketplace. The Marketplace would open in a pop up window. Write Weka in the text box and click search. Install the extension successfully and close.
3. Now import the given dataset of Data.xlsx in the data folder in your repository.
4. Drag it over to Design and drop in. Connect to res and view.
5. Go to Operators > Blending> Attributes> Types> Numerical to Binomial. Drag and drop into the design pane. Connect data to input and connect exa to res. Run to view the conversion.

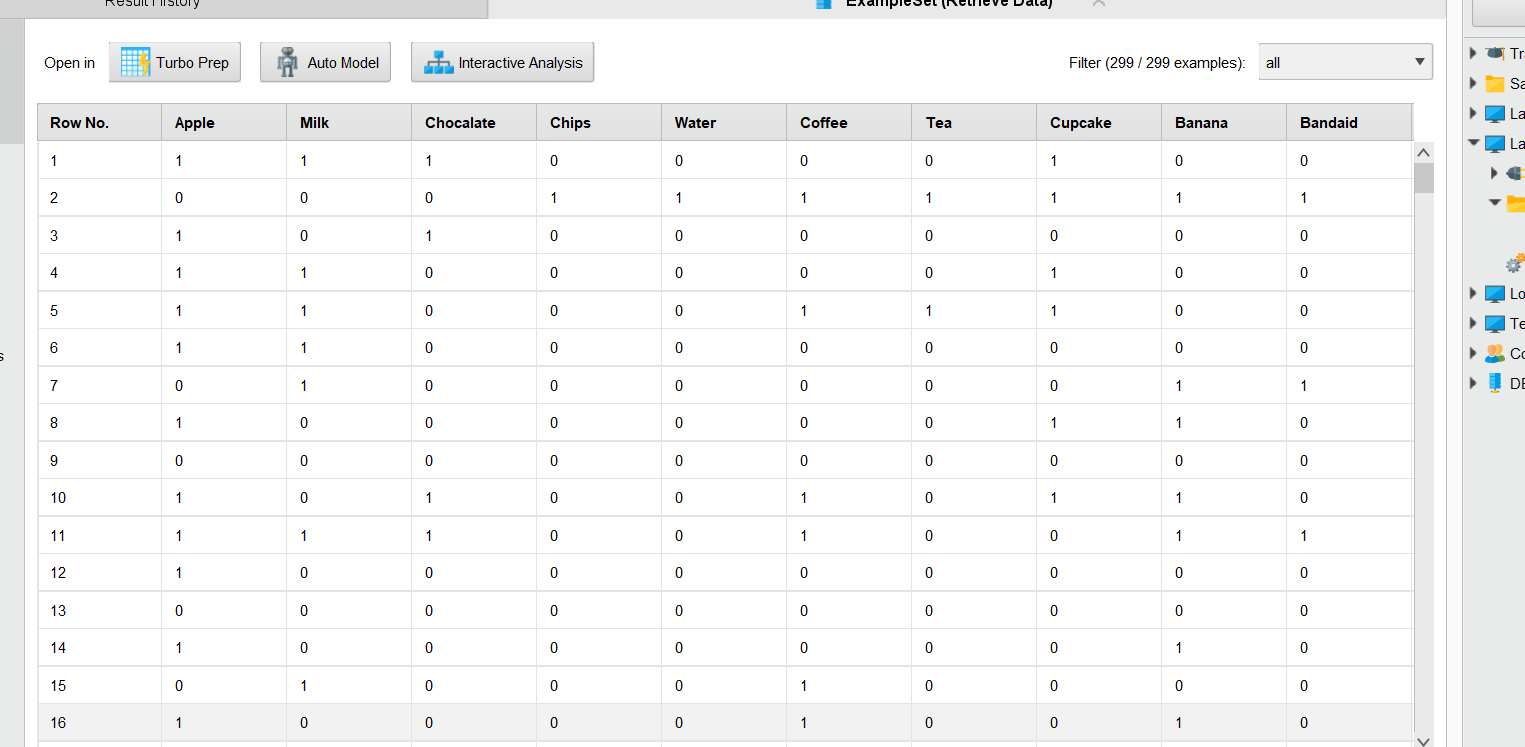


1. Go to Operators > Extensions > Weka > Modelling > Association and take the W-Apriori. Drag and drop to design. Connect the data to input and connect the ass to output.
2. Click on W-Apriori algorithm and see parameters in the right tab. Set the support & confidence to different thresholds and view results.

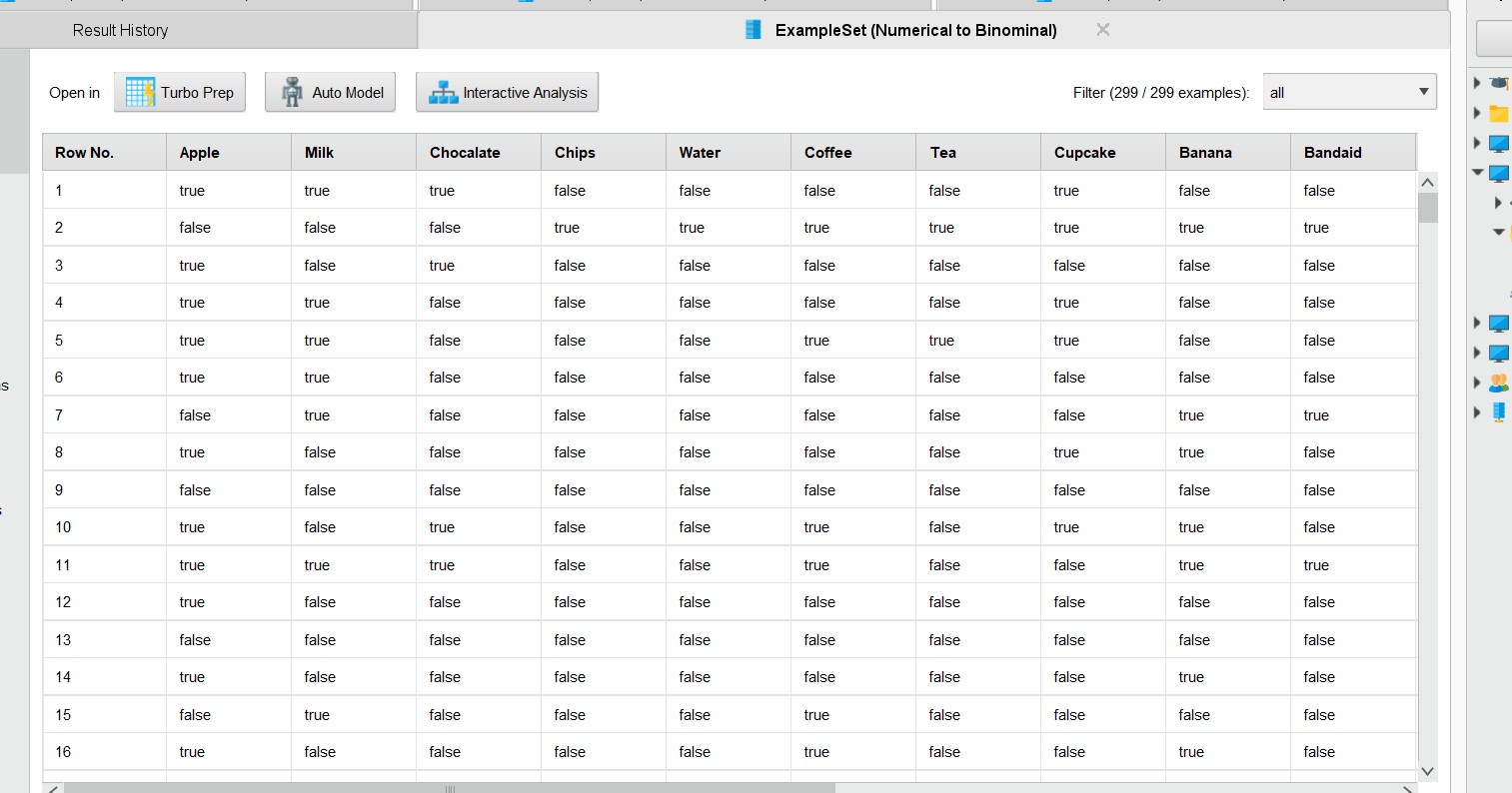


# Task 2: Fp-Growth Algorithm in Rapid Miner

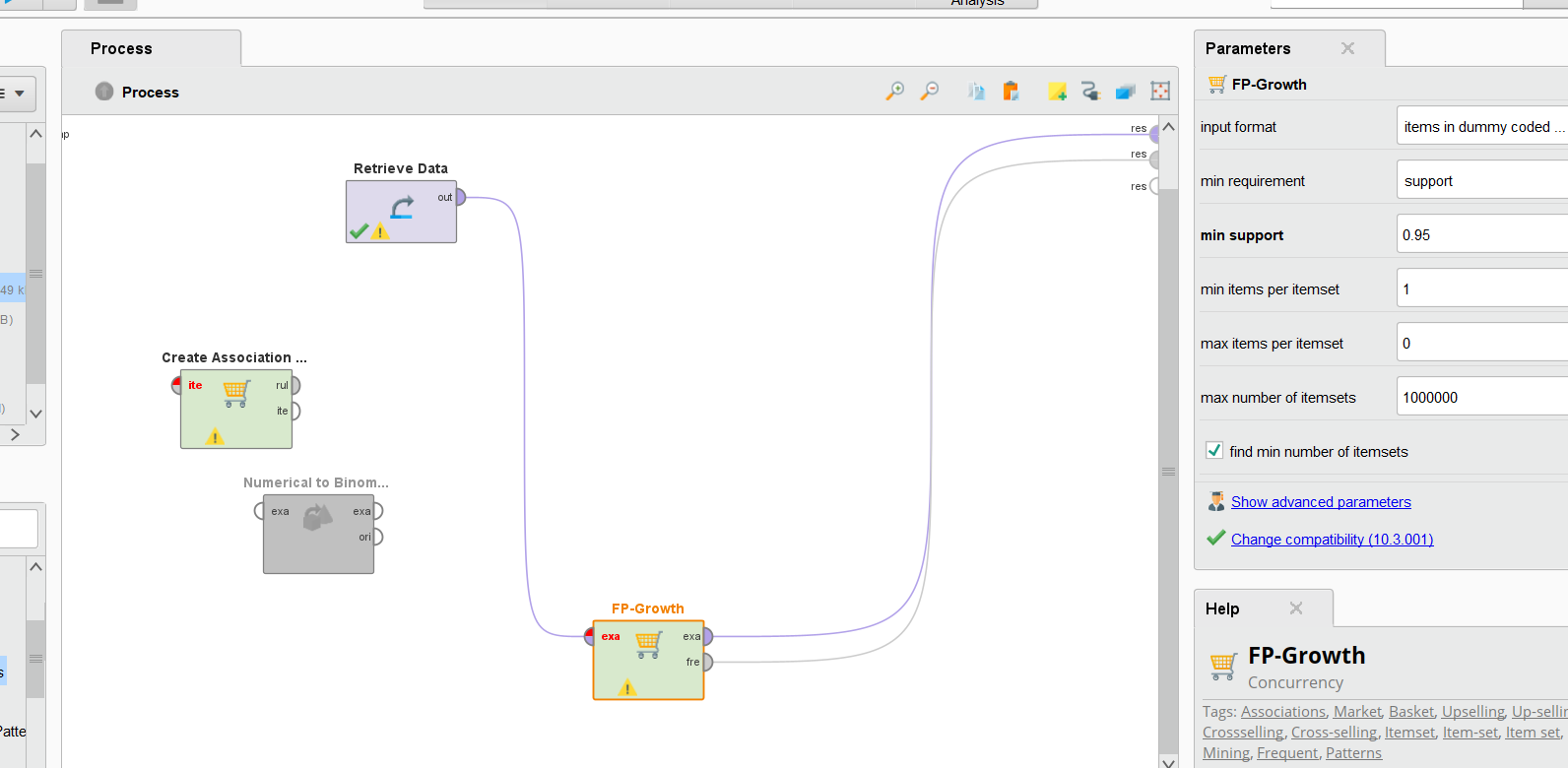
1. Open the Rapid Miner Studio and start a Blank Process.
2. Now import the given dataset of Data.xlsx into the data folder in your repository.
3. Drag the dataset in the Design pane and drop in. Connect to res and view.



1. Go to Operators > Blending> Attributes> Types> Numerical to Binomial. Drag and drop into the design pane. Connect data to input and connect exa to res. Run to view the conversion.

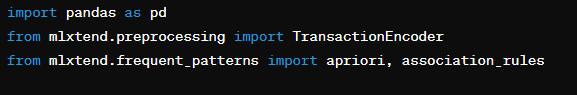


1. Go to Operators > Modelling > Associations and take the fp growth operator. Drag and drop to design. Connect the data to input and connect the exa and fre to res and run to view output.
2. Click on fp growth block and see parameters in the right tab. Set the support to different thresholds and view results.
3. Now go to Operators > Modelling > Associations and take the ‘Create Association Rules’ operator. In the parameters tab, set confidence threshold to different levels and run for viewing results.

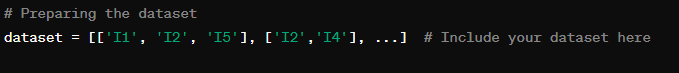


**Task 3 – Apriori Algorithm in Python**

1. Open the respective IDE for python programming.
2. Take any
3. Import necessary libraries and prepare the dataset.



1. Create a customized dataset or import from opensource given datasets.



1. Use the TransactionEncoder from mlxtend.preprocessing to encode the dataset into a one-hot encoded DataFrame.

A computer screen with white text

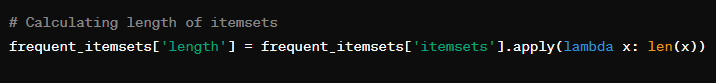
Description automatically generated

1. Apply the apriori algorithm from mlxtend.frequent\_patterns to find frequent itemsets with a minimum support threshold.

A black background with white text

Description automatically generated

1. Calculate the length of each itemset in the frequent itemsets DataFrame.

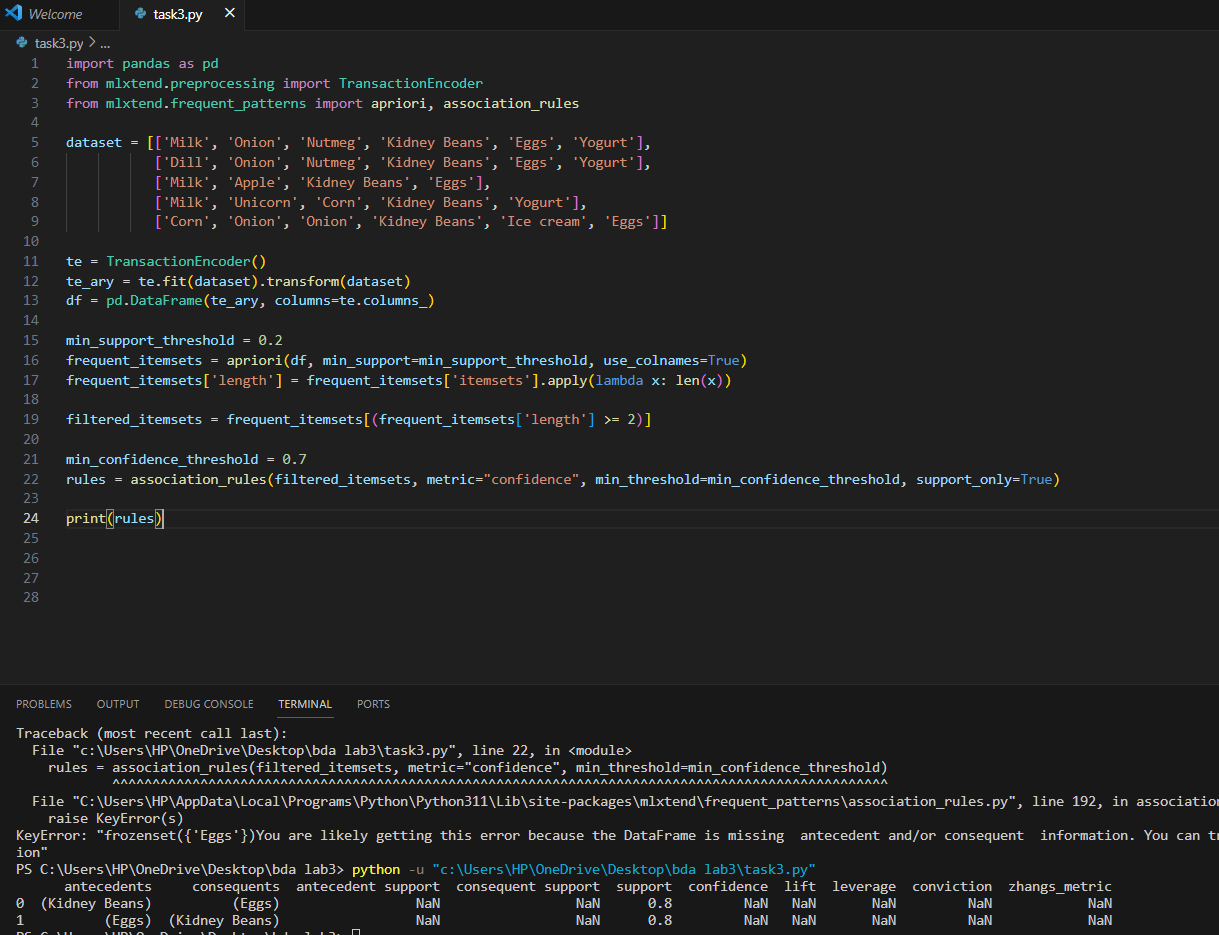


1. Filter the frequent itemsets based on a certain length or support criteria.
2. Use the association\_rules function from mlxtend.frequent\_patterns to extract association rules with a high confidence level.

A black background with white text

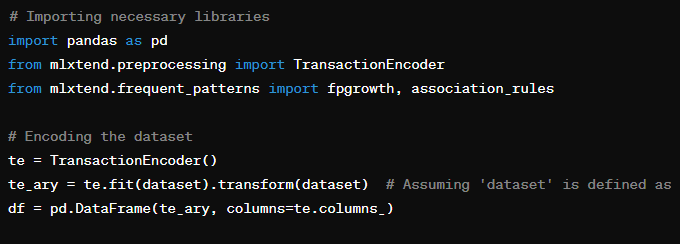
Description automatically generated

1. Display the rules mined through this algorithm.



**Task 4 – Fp Growth Algorithm in Python**

1. Follow the initial steps from Task 3 to import libraries and prepare the dataset.
2. Encode the dataset using the TransactionEncoder.



1. Apply the fpgrowth algorithm from mlxtend.frequent\_patterns to find frequent itemsets with a minimum support threshold.

A black background with white text

Description automatically generated

1. Optionally, calculate the length of each itemset and filter the frequent itemsets based on certain criteria.

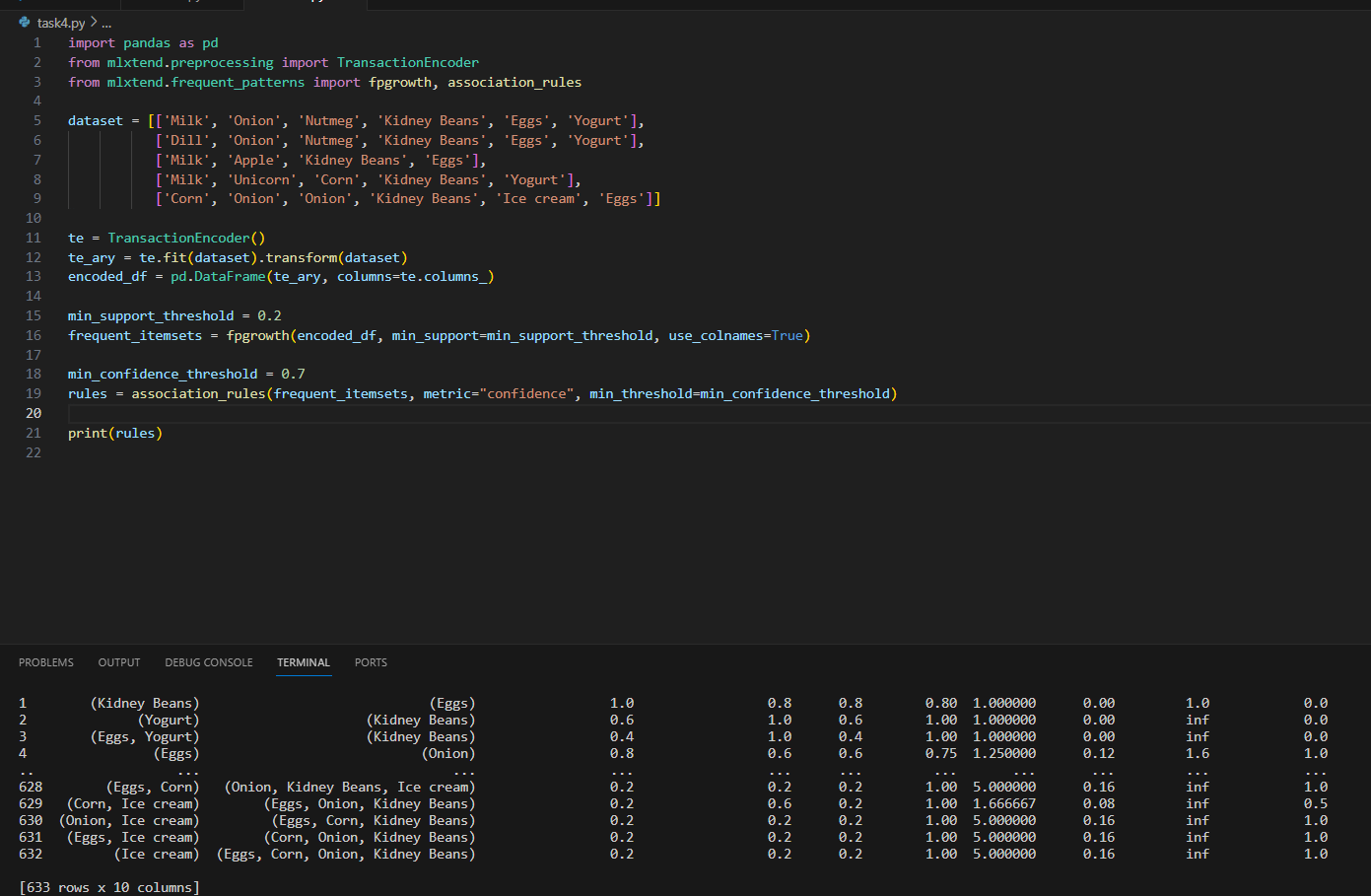
A screen shot of a computer code

Description automatically generated

1. Extract association rules with a high confidence level using the association\_rules function.

A black background with white text

Description automatically generated



# Task 5 – Applying Practice to opensource Dataset.

The goal is to analyze an open-source dataset to identify frequent itemsets and derive association rules using both the Apriori and FP-Growth algorithms. This exercise will help you understand the practical applications of these algorithms in real-world data mining tasks.

**Steps:**

1. Select an Open Source Dataset: Choose an open-source dataset relevant to market basket analysis or any dataset that involves transactions or item-based records. Websites like Kaggle, UCI Machine Learning Repository, or GitHub are good sources for such datasets.
2. Prepare the Dataset: Import the dataset into your Python environment and preprocess it as necessary. This may involve cleaning the data, handling missing values, and converting the dataset into a format suitable for the Apriori and FP-Growth algorithms.
3. Data Encoding: Use the TransactionEncoder from mlxtend.preprocessing to encode the dataset into a one-hot encoded DataFrame, making it suitable for frequent itemset mining.
4. Apply Apriori Algorithm: Utilize the apriori function from mlxtend.frequent\_patterns to identify frequent itemsets within the dataset based on a minimum support threshold you define.
5. Apply FP-Growth Algorithm: Similarly, apply the fpgrowth function to the same dataset to find frequent itemsets, allowing you to compare the performance and results of both algorithms.
6. Analyze the Results: For both algorithms, analyze the resulting frequent itemsets. Calculate the length of each itemset and filter based on certain criteria like minimum support, confidence, and lift to derive meaningful association rules.
7. Interpretation and Insights: Interpret the derived rules and itemsets to gain insights into the dataset. Look for patterns or trends that could be of interest or have practical applications.
8. Comparison and Evaluation: Compare the performance and outcomes of the Apriori and FP-Growth algorithms. Discuss any differences in execution time, efficiency, and the quality of the patterns discovered.

A screen shot of a computer program

Description automatically generated

