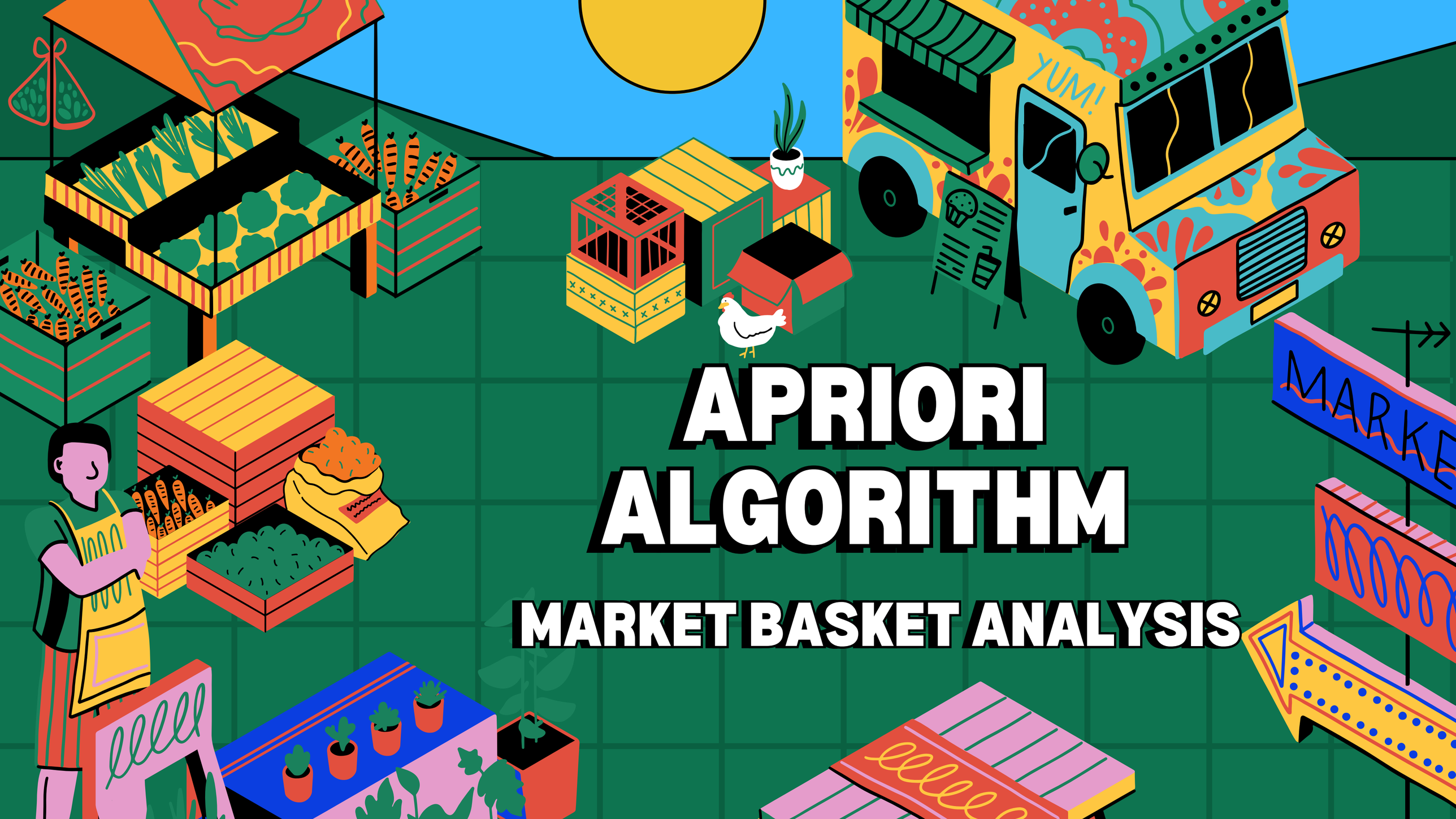


APRIORI ALGORITHM

MARKET BASKET ANALYSIS

INTRODUCTION

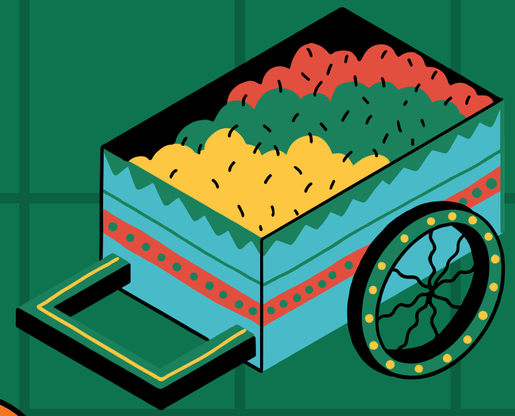
The Apriori algorithm is a popular **Data mining method** used to find frequent itemsets and generate association rules.
It’s widely used in market basket analysis to uncover patterns like:
"Customers who buy bread often also buy butter."

MARKET OF CONTENTS

1 What is Apriori Algorithm

2 How Apriori works

3 The Dataset & Visualiziation



4 Results

5 Conclusion

WHAT IS APRIORI ALGORITHM

- Association Rule learning algorithm
- All subsets of a frequent itemset must also be frequent

- **Key metrics**
  - **Support**

$$\text{Support}(x) = \frac{\text{Transaction containing X}}{\text{Total Transactions}}$$

  - **Confidence**

$$\text{Confidence}(A \Rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}$$

  - **Lift**

$$\text{Lift}(A \Rightarrow B) = \frac{\text{Support}(AUB)}{\text{Support}(A) * \text{Support}(B)}$$

HOW APRIORI ALGORITHM WORKS

Step 1 : Data preprocessing & One-Hot Encoding transaction formation

```
125   my_data["singleTransaction"] = my_data["Member_number"].astype(str) + '_' + my_data["Date"].astype(str)
126   transactions = my_data.groupby("singleTransaction")["itemDescription"].apply(list).tolist()
```

Transactions	Milk	Bread	Butter	Cookies
1	0	1	1	0
2	1	1	0	0
3	1	0	0	1
4	0	1	1	0
5	1	1	1	1

```
te = TransactionEncoder()
te_ary = te.fit(transactions).transform(transactions)
df_encoded = pd.DataFrame(te_ary, columns=te.columns_)
```

HOW APRIORI ALGORITHM WORKS

Step 2 : Generate frequent itemset

$$\text{Support (x)} = \frac{\text{Transaction containing X}}{\text{Total Transactions}}$$

$$\text{Support (Milk)} = \frac{3}{5} = \mathbf{0.6}$$

$$\text{Support (bread)} = \frac{4}{5} = \mathbf{0.8}$$

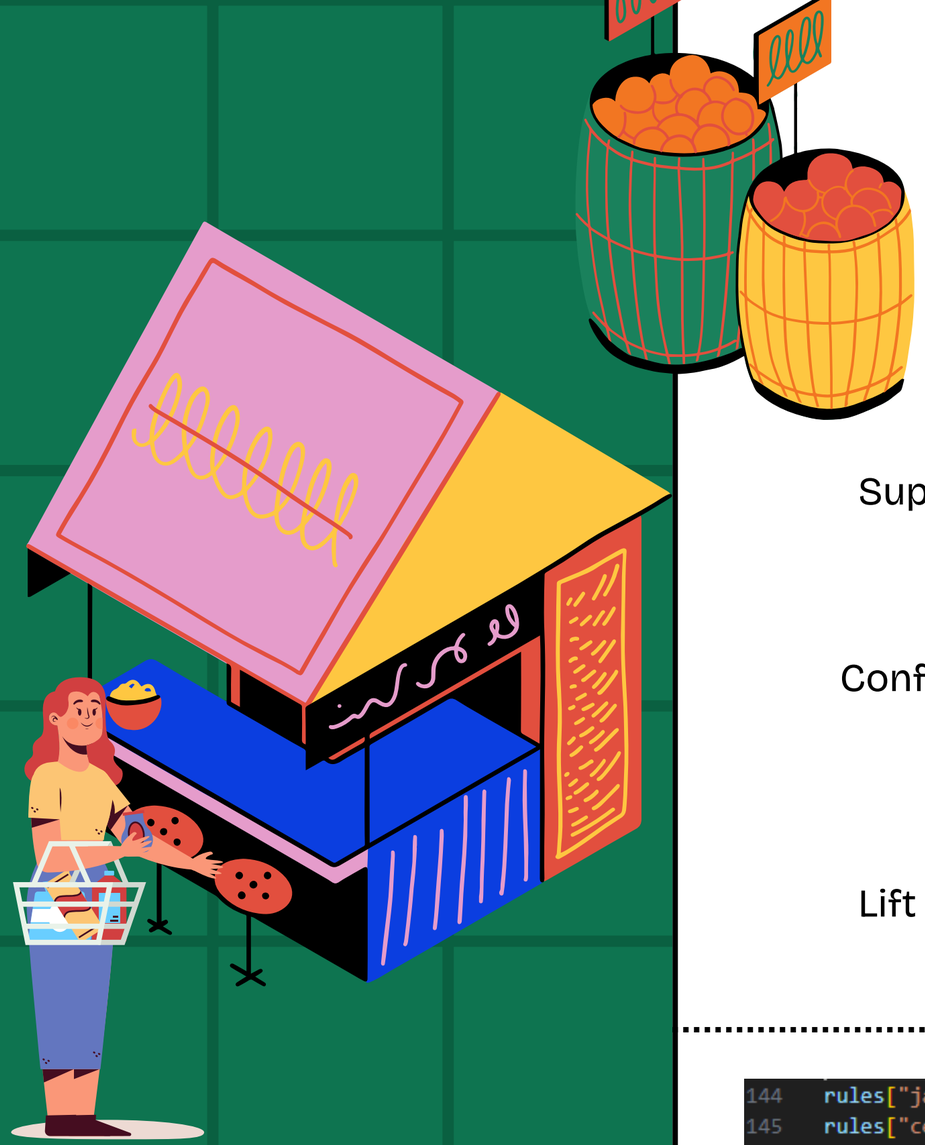
```
frequent_itemsets = apriori(df_encoded, min_support=0.0005, use_colnames=True)
frequent_itemsets["length"] = frequent_itemsets["itemsets"].apply(len)
```

Step 3 : Generate Association rules

A = {milk}, Rule: milk => {bread, butter}

A = {bread}, Rule: bread => {milk, butter}

A = {milk, bread}, Rule: {milk, bread} => butter



HOW APRIORI ALGORITHM WORKS

Step 3 : Calculate Rule Metrics

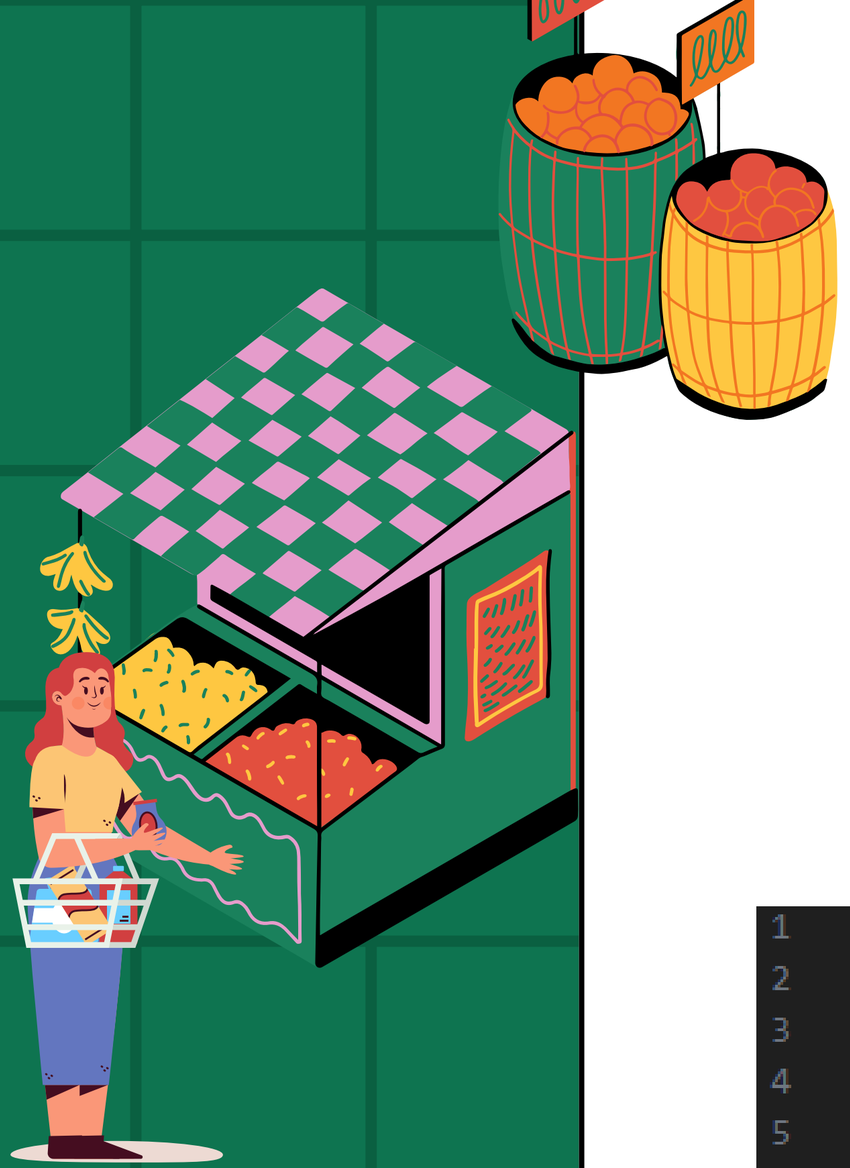
$$\text{Support } (A \Rightarrow B) = \frac{\text{Transaction containing both A \& B}}{\text{Total Transactions}}$$

$$\text{Confidence } (x) = \frac{\text{Support } (A \cup B)}{\text{Support } (A)}$$

$$\text{Lift } (A \Rightarrow B) = \frac{\text{Confidence } (A \Rightarrow B)}{\text{Support}(B)}$$

```
144    rules["jaccard"] = rules["support"] / (rules["antecedent support"] + rules["consequent support"] - rules["support"])
145    rules["certainty"] = rules["confidence"] - rules["consequent support"]
146    rules["kulczynski"] = 0.5 * (rules["confidence"] + rules["support"] / rules["consequent support"])
```

- Jaccard Index = $\frac{P(A \cap B)}{P(A) + P(B) - P(A \cap B)}$
- Certainity = Confidence - Support (B)
- Kulczynski = 1/2 (P(B|A) + P(A|B))



DATASET &VISUALIZATION

Source : Groceries_dataset.csv

Structure:

- Member_number
- Date
- itemDescription

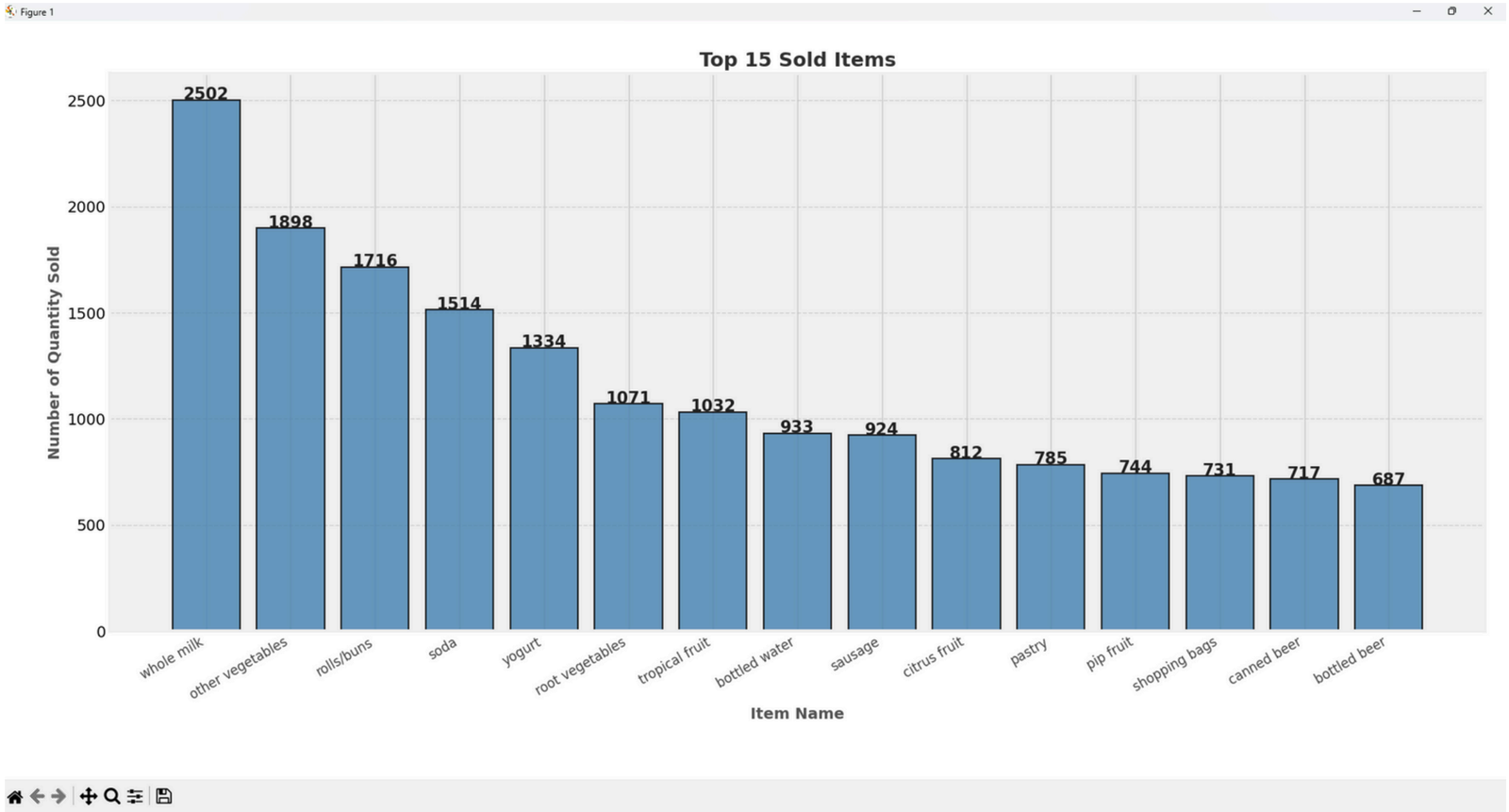
Visualization:

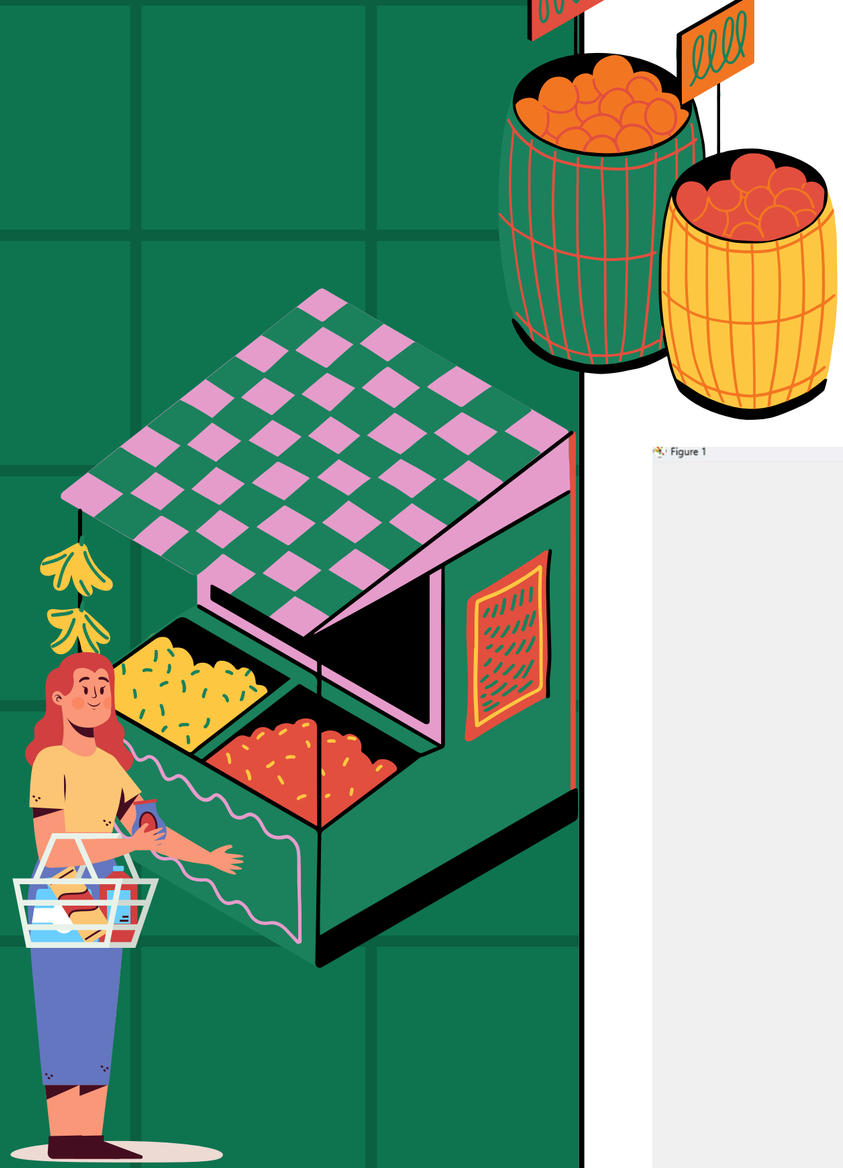
- Bar Chart
- Heatmap
- Network Graph

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import networkx as nx
from itertools import combinations
from collections import Counter
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules
import time
```

DATASET &VISUALIZATION

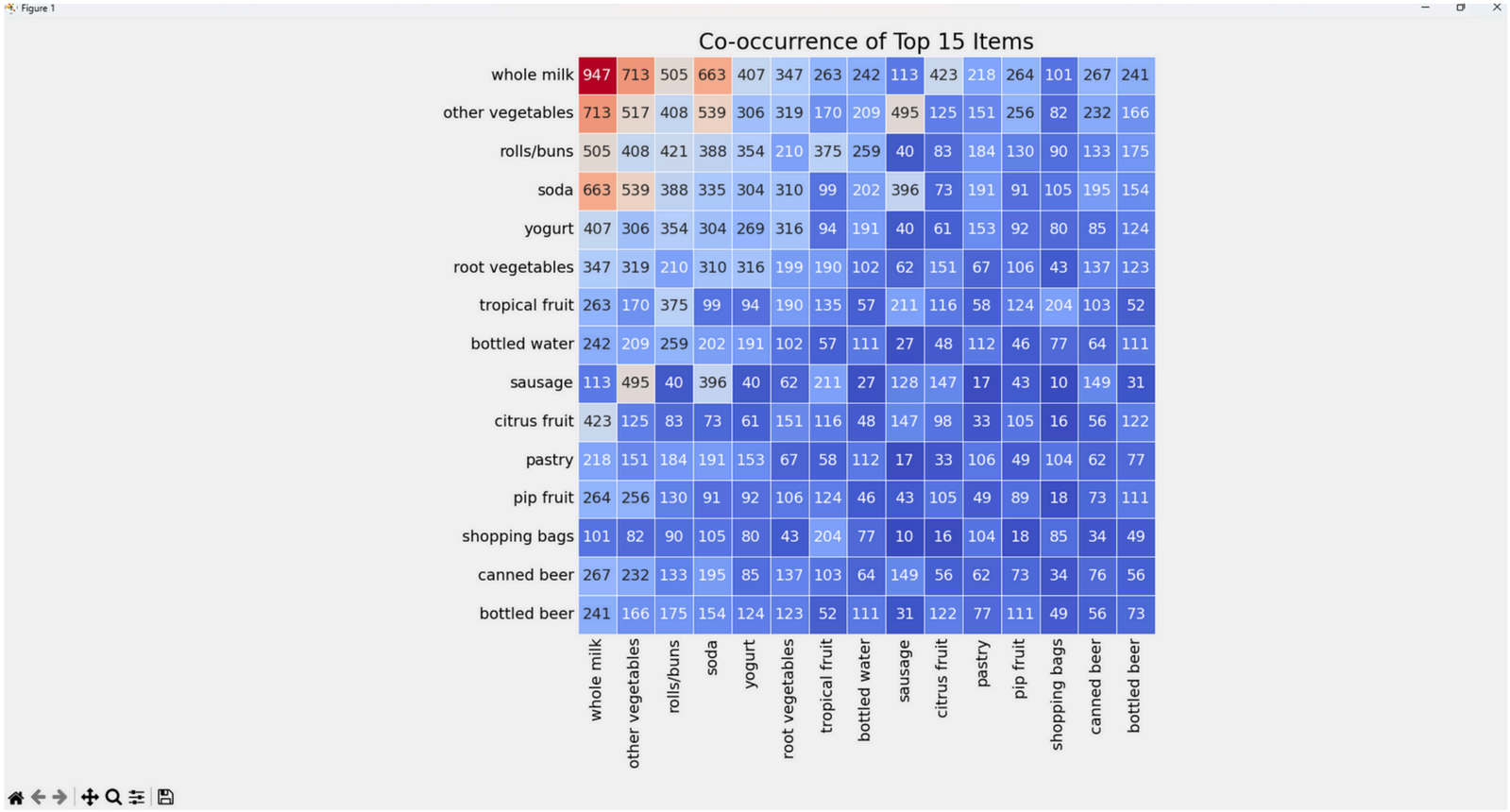
- **Bar Chart**

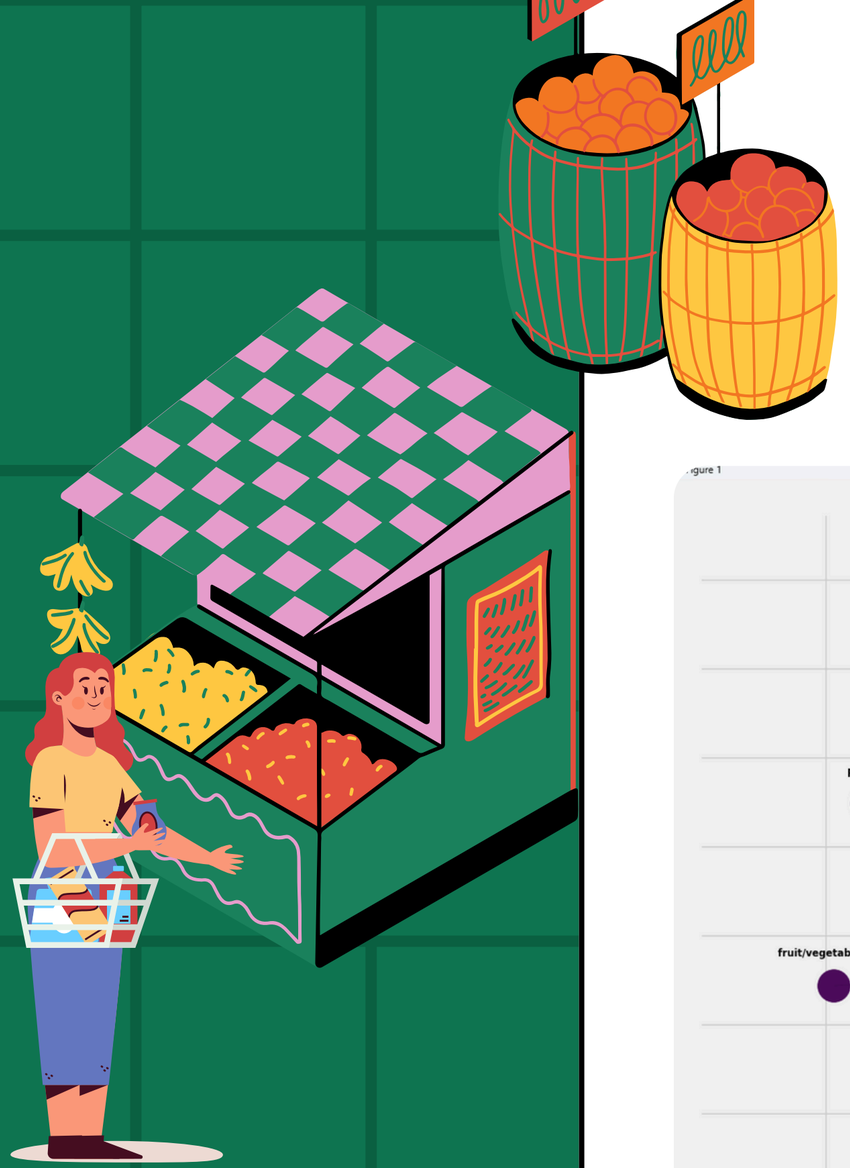




DATASET &VISUALIZATION

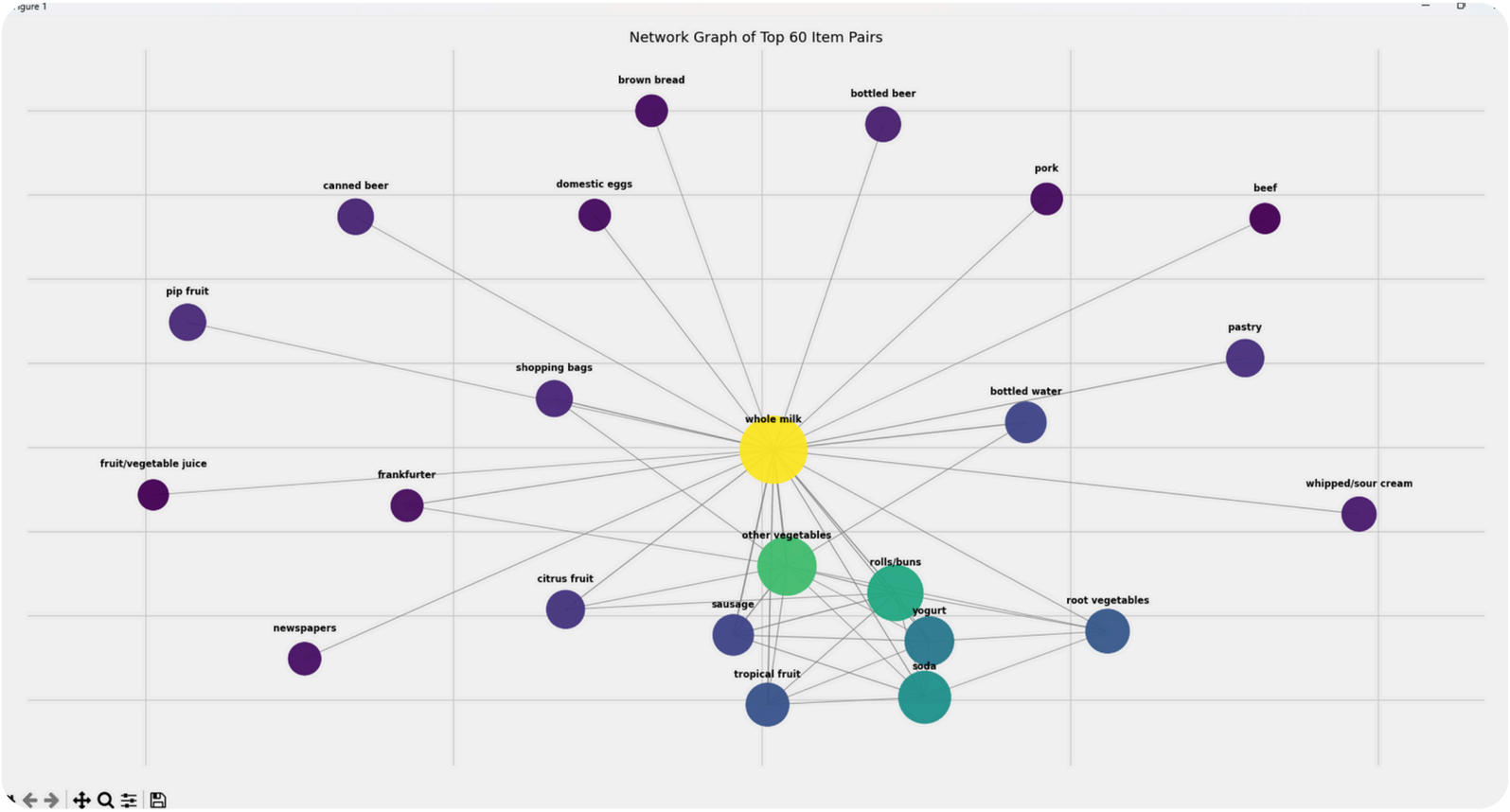
- **Heatmap**





DATASET &VISUALIZATION

- **Network Graph**



RESULTS

Antecedents	Consequent	Confidence	Lift
Sausage, Pork	Whole Milk	39.1%	2.48
Sweet spreads	pip fruit	11.7%	2.40
Yogurt	Sausage	13.2%	2.18

Outcomes :

- Whole milk is a key product – it appears in many strong rules.
- Sausage is often bought together with other proteins and milk.
- Items like sweet spreads and pip fruits form niche but strong associations.

CONCLUSION

- **Benefits**

Easy to understand and implement
Works well for market basket and transactional data
Helps identify product affinities (e.g., what items are bought together)

- **Limitations**

Can be computationally expensive on large datasets
Requires setting proper support/confidence thresholds

- **Use Cases**

Retail product placement and bundling
Fraud detection & web usage mining

THANK YOU

SALE ON QUESTIONS

50% OFF