# Market Basket Analysis: Project Report

---

### 1. Introduction

This project focuses on performing **Market Basket Analysis** (MBA), which is a key technique in data mining used to find associations between products purchased together by customers in a retail environment. The analysis uncovers **frequent itemsets** (sets of products that frequently appear in transactions) and generates **association rules** that highlight how items are related to each other. These rules help businesses understand customer behavior and improve product recommendations, inventory management, and targeted marketing.

In this report, I will walk through the steps involved in the project, the algorithms used, how they were enhanced over existing research, my approach to market basket analysis, how clustering improved the results, and how I implemented a **Streamlit interface** to visualize and interact with the results.

---

### 2. Data Collection and Preprocessing

#### 2.1 Data Collection

The dataset consists of **transaction data** that includes:

- **Transaction IDs**: Unique identifiers for each purchase.
- **Product IDs or Names**: Products purchased in a transaction.
- **Quantity Purchased**: The number of units of each product in the transaction.
- **Customer IDs**: Identifier for the customer making the purchase (optional, but helpful for clustering).
- **Timestamp**: Time when the transaction occurred.

In this project, I utilized **public datasets** obtained from retail sources, which I processed into a structured format compatible with the Market Basket Analysis algorithms.

#### 2.2 Data Preprocessing

Data preprocessing is a crucial step to ensure that the data is ready for analysis. The key steps involved in preprocessing the data were:

1. **Data Cleaning**:

- **Handling Missing Values**: Any rows containing missing or invalid product data were removed or imputed using suitable methods.
- **Duplicate Removal**: Duplicate transactions or redundant data were cleaned up to avoid skewing results.

2. **Data Transformation**:
   - **Transaction Matrix Creation**: The raw transaction data was transformed into a **binary matrix** format, where each row represents a transaction, and each column represents a product. The entries are 1 if the product was bought in the transaction, and 0 otherwise.
3. Example:

| Transaction ID | Milk | Bread | Butter | Chees |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 1 |
| 2 | 0 | 1 | 1 | 0 |
| 3 | 1 | 0 | 1 | 1 |

---

## 3. Market Basket Analysis with Apriori Algorithm

### 3.1 The Apriori Algorithm

The **Apriori algorithm** is a classic algorithm for frequent itemset mining. It operates by identifying frequent itemsets in a dataset and generating **association rules** from those itemsets.

The core steps in the Apriori algorithm are:

1. **Finding Frequent Itemsets**: The algorithm starts by finding itemsets that appear frequently (above a specified **support** threshold). Support is the fraction of transactions that contain the itemset.
   - **Support Calculation**: If 30 out of 1000 transactions contain Milk, then the support of Milk is 30/1000 = 0.03.
2. **Generating Association Rules**: After frequent itemsets are identified, the algorithm generates **association rules**. Each rule has a form like:
   A⇒BA \Rightarrow BA⇒B
   This means that if **A** is purchased, then **B** is likely to be purchased. The quality of the rule is measured by:
   - **Confidence**: The probability that **B** is purchased given **A** was purchased.
   - **Lift**: The ratio of the observed support of **A and B** together to the expected support if they were independent.
3. **Pruning Rules**: In the classic Apriori approach, too many rules are generated. To overcome this, I implemented a **rule pruning mechanism** that filters out rules with low confidence or lift values, keeping only the most relevant and insightful rules.

### 3.2 Optimization and Improvements

While the traditional Apriori algorithm can be computationally expensive, especially with large datasets, I implemented the following optimizations:

- **Parallel Processing**: I leveraged **Azure Databricks**, which provides a distributed computing environment to speed up the process of identifying frequent itemsets by parallelizing the computations.
- **Rule Pruning**: A post-processing step was added to filter out irrelevant rules. This reduces noise and makes the results more meaningful.

---

## 4. Azure Cloud Integration

### 4.1 Azure Workflow

Azure cloud infrastructure was utilized for data storage, processing, and experimentation. This enhanced the scalability, efficiency, and automation of the project.

1. **Azure Databricks**:
   - **Distributed Computing**: By running the Apriori algorithm on **Azure Databricks**, the computational burden of handling large datasets was mitigated. Databricks provided a parallel computing environment using Apache Spark, significantly speeding up the analysis.
   - **Optimization**: The process of finding frequent itemsets and generating rules was parallelized across multiple nodes, reducing execution time from hours to minutes.
2. **Azure Storage**:
   - **Blob Storage**: All the datasets (raw, cleaned, and processed) were stored in **Azure Blob Storage**. This ensured that the data could be easily accessed and managed securely.
3. **Azure Machine Learning**:
   - **Experimentation**: Azure ML was used to track experiments, evaluate models, and optimize hyperparameters for clustering and Apriori analysis.

---

## 5. Customer Segmentation with Clustering

### 5.1 K-Means Clustering

To enhance the results of Market Basket Analysis, I integrated **K-means clustering** to segment customers based on their purchasing behavior. This approach allowed for more personalized recommendations.

**Steps Involved**:

1. **Feature Selection**: I selected relevant features like product categories, quantities, and frequency of purchases for clustering.
2. **K-means Algorithm**: The K-means algorithm was used to group customers into clusters based on their buying patterns. Each cluster represents a set of customers with similar purchasing behavior.
3. **Cluster Analysis**: After clustering, I applied the Apriori algorithm to each cluster separately. This led to more targeted rules, improving the quality of recommendations.

**Benefit**: By segmenting customers, the analysis becomes more personalized. For example, a customer who frequently buys dairy products can receive recommendations based on the purchase behavior of similar customers.

---

## 6. Streamlit Interface for Visualization

### 6.1 Streamlit Dashboard

To make the results of Market Basket Analysis accessible and understandable to non-technical users, I developed an interactive **Streamlit dashboard**. The key features of the interface included:

1. **Visualization of Association Rules**: Users could visualize the discovered association rules in a table, with information about the support, confidence, and lift of each rule. The interface also allowed users to filter rules based on these metrics.
2. **Interactive Clustering Insights**: The dashboard provided an overview of the customer segments generated by K-means clustering. Users could explore the characteristics of each cluster and how purchasing behaviors differ across segments.
3. **Real-Time Exploration**: The interface allowed users to adjust parameters like minimum support and confidence to refine the rules. This feature made the analysis more interactive and useful for business decision-makers.

---

## 7. Comparison with Existing Research

### 7.1 Improvements Over Existing Research

In the **original research paper**, the authors implemented a traditional version of the Apriori algorithm without optimizations or personalization features. My contributions have significantly enhanced the approach:

1. **Scalability**: By using Azure Databricks for distributed computing, I improved the performance of the Apriori algorithm on large datasets.
2. **Personalization**: By adding customer segmentation with K-means clustering, I ensured that the association rules were more tailored and relevant for specific customer groups.

3. **Rule Pruning**: I implemented a mechanism to filter weak rules, improving the overall quality of the results.
4. **User Interface**: I developed a **Streamlit dashboard** to visualize the results and make them more accessible to business stakeholders.

#### 7.2 Gaps Filled

The gaps in the original research that I addressed include:

1. **Scalability for Large Datasets**: The original method may struggle with large datasets. Using distributed computing on Azure solved this.
2. **Customer-Centric Recommendations**: Adding clustering allows for more personalized recommendations, addressing the one-size-fits-all approach of the original paper.
3. **Usability**: I added an interactive dashboard, making it easier for non-technical users to explore the data and findings.

---

### 8. Conclusion

This project successfully implemented **Market Basket Analysis** using the **Apriori algorithm**, optimizing it with parallel computing, rule pruning, and clustering for more personalized insights. The integration of **Azure cloud services** for processing and storage ensured scalability, while the **Streamlit dashboard** allowed stakeholders to interact with and visualize the results.

Through these improvements, the project offers more efficient, actionable, and personalized recommendations for businesses looking to leverage their transaction data.

---

### 9. Future Directions

1. **FP-Growth Algorithm**: For even faster itemset mining, the **FP-growth algorithm** could be integrated as it is more efficient than Apriori for large datasets.
2. **Real-Time Data Processing**: Incorporating **real-time data** streams could allow the system to update its association rules and recommendations dynamically.
3. **Advanced Models**: Exploring **neural networks** or other deep learning techniques could uncover more complex patterns in customer purchasing behavior.