جامعة أم القرى
UMM AL-QURA UNIVERSITY

Data Analysis 2 Project

Duration: 1st Semester

Task 3 (MBA)

Submitted By:

Reem Ghazi Alosaimi	***444001268***
Remas Majed Almalki	***444002871***

Introduction:

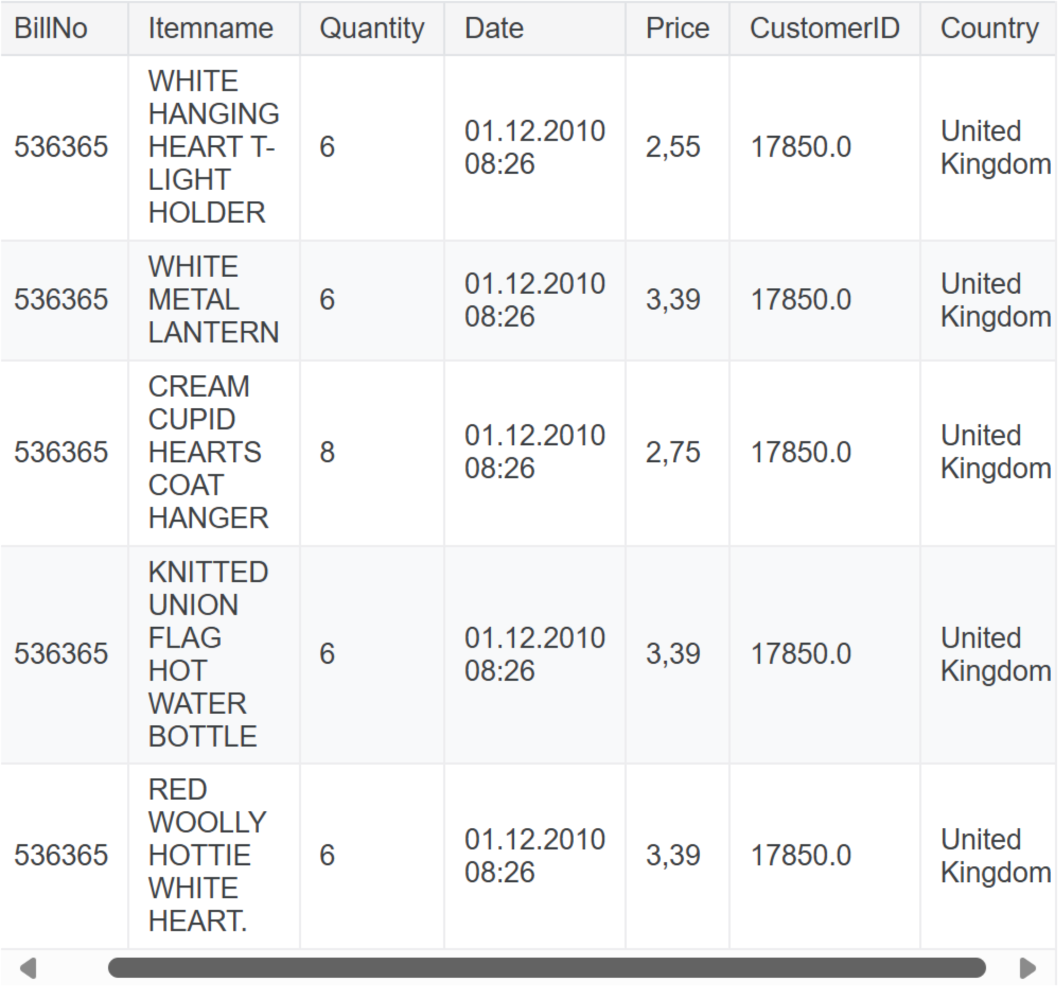
Market Basket Analysis (MBA) involves a collection of techniques and algorithms designed to extract valuable insights from transaction data. Rather than being a single approach, it encompasses a variety of methods that share a common goal: to transform transactional data into structured patterns, enabling further analysis. To begin, let's consider a dataset containing transactions from a retail setting.

As one of the most straightforward yet effective approaches, Market Basket Analysis facilitates the discovery of relationships between items that are frequently purchased together, allowing businesses to understand customer behavior and optimize sales strategies. This process is highly valuable for various applications such as product recommendation, store layout design, inventory management, and promotional strategies.

In Market Basket Analysis, the data typically involves high dimensionality, as each item or combination of items can represent a unique feature. These methods are extensively used for tasks like association rule mining, frequent itemset generation, and product bundling. The advantage of MBA lies in its ability to quickly process and analyze large transactional datasets, uncovering patterns that reveal consumer preferences and behaviors, even when the data contains thousands of products.

MBA models work by estimating the likelihood that a specific product or set of products is associated with other items within a transaction. They use probabilistic and statistical techniques, such as association rules and support-confidence analysis, to identify patterns and relationships within the transactional data. By leveraging these algorithms, Market Basket Analysis remains a robust and efficient solution for uncovering meaningful insights from large and complex retail datasets.

Data Features:



BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
536365	WHITE HANGING HEART T-LIGHT HOLDER	6	01.12.2010 08:26	2,55	17850.0	United Kingdom
536365	WHITE METAL LANTERN	6	01.12.2010 08:26	3,39	17850.0	United Kingdom
536365	CREAM CUPID HEARTS COAT HANGER	8	01.12.2010 08:26	2,75	17850.0	United Kingdom
536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	01.12.2010 08:26	3,39	17850.0	United Kingdom
536365	RED WOOLLY HOTTIE WHITE HEART.	6	01.12.2010 08:26	3,39	17850.0	United Kingdom

ANALYSIS:

Data preparation:

:Steps to follow in preparing the data:

- Converting the date column: At the beginning of the code, we converted the "Date" column to a date format using the pd.to_datetime function to ensure that all date values are structured in the same format, which is necessary for accurate analysis of time periods.

- Adding the date period column: A new column called "Year/Month" was created using the .dt.to_period("M") function, which aggregates the data at the year and month level. This helps in analyzing time trends.

- Cleaning up numeric values: It was noticed that the "Price" column contains decimal points written as commas rather than periods. We replaced the commas with periods and converted the column to float using .astype(float). This step ensures the ability to perform accurate mathematical and analytical operations on prices.

- Handling Missing Values: Missing values were checked using df.isnull().sum(), where the "Itemname" column was found to have 1455 missing values, while the "CustomerID" column had 134041 missing values. The rest of the columns were free of missing values.

- Visualizing Missing Values: Using the missingno library, a chart was created to visualize the missing values.
  This step allows us to visually understand how the missing values are distributed in the data.

- Descriptive Statistics: Finally, we used the .describe() function to generate descriptive statistics such as mean, standard deviation, maximum and minimum values for each column. These statistics summarize the nature of the data and provide an overview of the distribution of values.

Interpreting the code output:
The "Quantity" column: contains a count of 522064 with an average order quantity of around 10 units per row. However, we noticed that there were negative values in the quantities (minimum -9600) indicating possible errors in the data.

"Price" column: Represents the prices of items sold, the average price is around 3.83, but there is also a negative value in the prices (minimum -11062.06) which is not logical, indicating data issues that need to be corrected.

"CustomerID" column: This column contains only 388023 values out of 522064, which means there is a significant shortage of customer IDs (134041 rows with missing values). This can be a challenge when analyzing customer behavior.

Extreme values: The extreme values for quantity and price are unexpectedly high (80995 and 13541.33 respectively), indicating anomalies that may need further .cleaning or verification

Data Filtering and Cleaning:

Filtering out non-positive values: The code df = df[(df["Quantity"] > 0) & (df["Price"] > 0)] was used to filter out all rows that had non-positive values in the "Quantity" and "Price" columns. This step is necessary to ensure that the analysis does not rely on incorrect values such as negative quantities or prices that may be the result of input errors.

Deleting rows that lack item names: Using the code df = df[df["Itemname"]. notnull()], all rows that had missing values in the "Itemname" column were deleted. These rows are not useful for analysis since they lack essential data about the item.

Filling in missing values in customer IDs: The missing values in the "CustomerID" column were filled using the default value '#NV'. This step helps preserve rows that have missing values in customer IDs without completely deleting them, which may result in losing other potentially useful data.

Calculating the total price for each transaction: A new column called "TotalPrice" is created to calculate the total amount for each row using the relationship df["TotalPrice"] = df["Quantity"] * df["Price"]. This amount is the product of quantity multiplied by price,

which makes it easier to analyze the financial performance of the data.

These steps aim to improve the quality of the data, by:
Eliminating values that may distort the analysis.
Ensuring that all essential information such as item names are present.
Calculating the necessary financial groups such as the total price for each transaction.
The resulting data is now more organized and ready for more accurate and reliable analysis.

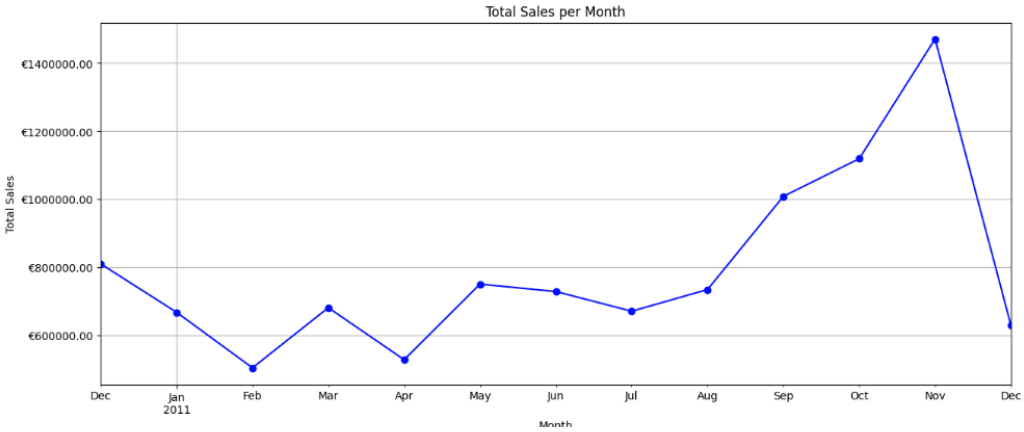
Total Sales Analysis:

We analyzed the time trends of total sales across months. The data was grouped by month and year using the “Year/Month” column and the total sales were calculated for each month. The goal of this process is to understand how sales have changed over time, and thus discover patterns or periods of increase or decrease in sales. We analyzed the time trends of total sales across months. The data was grouped by month and year using the 'Year/Month' column and the total sales for each month were calculated. The goal of this process is to understand how sales have changed over time, and thus discover patterns or periods where sales have increased or decreased.

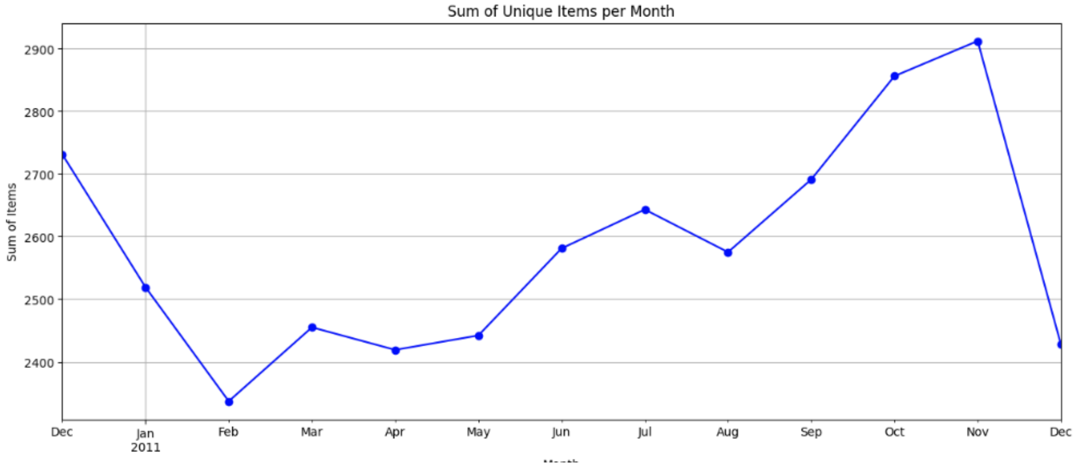
The data was grouped by month and year: The data was grouped using .groupby('Year/Month')['TotalPrice'].sum(), where the total sales for each month are calculated, giving us a comprehensive view of the monthly financial performance.

Data visualization:
A line chart was created using the plot() function to plot the total sales for each month.

A circular 'o' was added at each point on the line to indicate the specific value for each month.
The chart size was set to be larger (15x6) to ensure clarity of detail.
The axes were labeled: the x-axis (horizontal) represents the months, and the y-axis (vertical) represents the total sales.
The financial values on the y-axis were formatted using ticker.FormatStrFormatter to display them as financial values with the euro as the currency symbol.
Added Grid: A grid has been added to the chart to make reading data and trends easier, A line chart shows the total sales for each month. It can be used to see trends whether sales are increasing or decreasing over time.
For example, if we see an increase in sales during certain periods such as holiday months, we can conclude that there is a seasonal effect on sales.



we analyze the uniqueness of items sold per month by grouping the data and calculating the count of unique items. The line plot visualizes the sum of unique items per month.



MBA – Preprocessing:

We prepare the data and transform it into a form that can be used to analyze patterns and interactions between different items purchased together.

- First, select relevant columns: The columns "BillNo" and "Itemname" were selected for use in the analysis, where "BillNo" represents each individual transaction, and "Itemname" represents the item purchased.

- Encoding the items using One-Hot Encoding:

Binary encoding was used to convert item names into columns, where each column represents a specific item, and if that item was purchased in a specific transaction, it is given a value of 1, otherwise it is given a value of 0. This step allows us to create a basket representation of each transaction.

- Grouping the data based on invoice number:

The data was grouped using .groupby('BillNo').sum(), which created a basket matrix where each row represents a single transaction and each column a unique item.

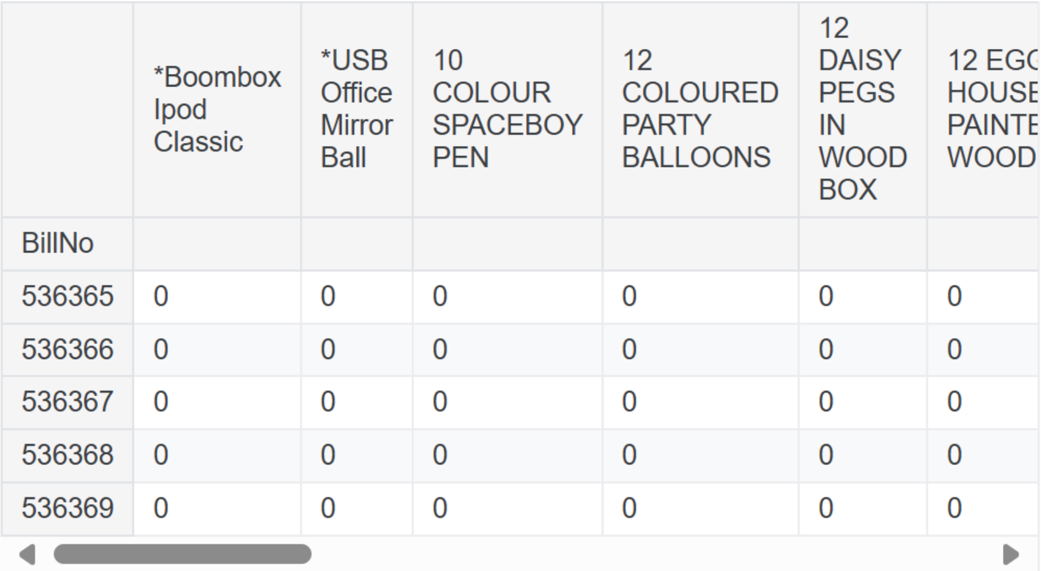
- Converting values to binary:

After collecting the data, all values greater than 0 were converted to 1 using basket[basket > 0] = 1. The goal of this step is to focus on the presence of only items regardless of quantity, which simplifies the analysis of patterns between items.

- Data after encoding: After applying binary encoding and collecting the data, we have a matrix where each row represents a transaction, and each column represents an item. The values in this matrix appear as 0 or 1, so:

"0"means that the item was not purchased in the transaction.

"1"means that the item was purchased.



	*Boombox Ipod Classic	*USB Office Mirror Ball	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 DAISY PEGS IN WOOD BOX	12 EGG HOUSE PAINTE WOOD
BillNo						
536365	0	0	0	0	0	0
536366	0	0	0	0	0	0
536367	0	0	0	0	0	0
536368	0	0	0	0	0	0
536369	0	0	0	0	0	0

Association Rule Mining:

Explanation of basic concepts:

- Confidence: Represents the probability of purchasing an item when another item is purchased. For example, if confidence is 0.6, it means that in 60% of cases where the first item was purchased, the second item was also purchased. High values (such as 0.8 or higher) are a strong indicator of a positive relationship between the items.

- Lift: Measures how much more likely an item is purchased with another item than the two items are purchased independently. A lift value greater than 1.0 indicates that the items are associated with each other more than would be expected by chance, meaning that there is a strong relationship between them.

- Support: Measures how often a given set of items appears in transactions. For example, a support of 0.05 means that the rule appears in 5% of all transactions.

Steps:

- Extract Confidence Rules:

The association_rules function was applied to identify rules with a confidence level greater than 0.8.
The results contain 4 rules, all of which indicate a strong relationship between the items.

An example in the results is the relationship between "PINK REGENCY TEACUP AND SAUCER" and "GREEN REGENCY TEACUP AND SAUCER", where the confidence level was 0.822, meaning that in 82.2% of transactions where the first item was purchased, the second item was also purchased.

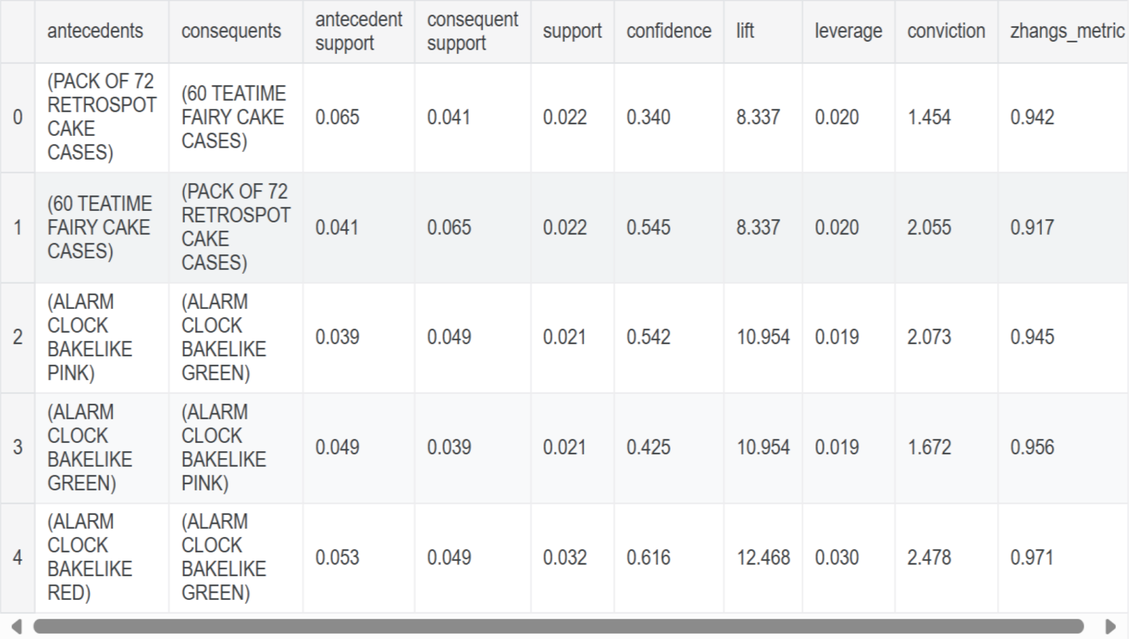
- Lift Rules Extraction:

A minimum lift of 2.5 was used to extract rules that represent strong connections between items.
174rules were extracted. For example, the relationship between "PACK OF 72 RETROSPOT CAKE CASES" and "60 TEATIME FAIRY CAKE CASES" has a lift of 8.337, indicating that the purchase of the second item with the first occurs eight times more often than would be expected by chance.

- Support Rules Extraction:

Rules with a support level higher than 0.03 were extracted, meaning that they appear in at least 3% of transactions.
For example, the relationship between “ALARM CLOCK BAKELIKE RED” and “ALARM CLOCK BAKELIKE GREEN” appears in 3.2% of transactions, with a confidence level of 0.616 and a leverage of 12.468, indicating a strong relationship between the two items.

Association rule extraction provides important insights into customer behavior, helping to make informed decisions about product placement in stores, identifying promotions,



	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(PACK OF 72 RETROSPOT CAKE CASES)	(60 TEATIME FAIRY CAKE CASES)	0.065	0.041	0.022	0.340	8.337	0.020	1.454	0.942
1	(60 TEATIME FAIRY CAKE CASES)	(PACK OF 72 RETROSPOT CAKE CASES)	0.041	0.065	0.022	0.545	8.337	0.020	2.055	0.917
2	(ALARM CLOCK BAKELIKE PINK)	(ALARM CLOCK BAKELIKE GREEN)	0.039	0.049	0.021	0.542	10.954	0.019	2.073	0.945
3	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE PINK)	0.049	0.039	0.021	0.425	10.954	0.019	1.672	0.956
4	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.053	0.049	0.032	0.616	12.468	0.030	2.478	0.971

Analysis of Association Rule Metrics:

We examine the distribution of the key metrics associated with the extracted association rules, namely Confidence, Lift, and Support. These plots provide insight into the distribution and frequencies of these metrics across association rules.

- Confidence Distribution:

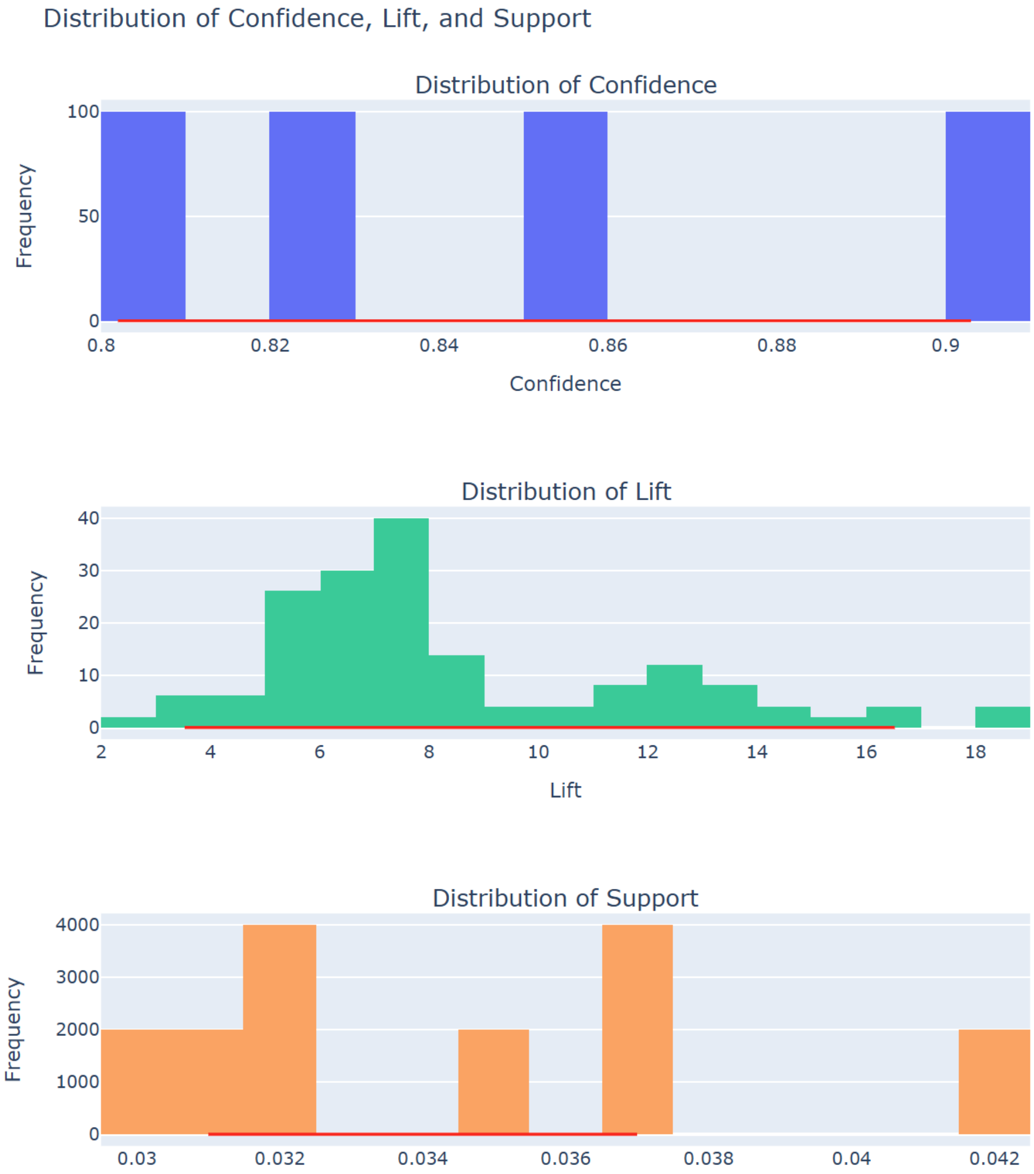
The plot shows the distribution of confidence values for the extracted rules. If we see values concentrated in high ranges (e.g. above 0.8), this indicates that the majority of the extracted rules have strong relationships, meaning that the dependent item is purchased in a high percentage of transactions that include the primary item.

- Lift Distribution:

The plot shows the distribution of lift for the rules. If the majority of the values are above 1, this indicates that there is a strong relationship between the items, as purchasing the dependent item increases the probability of purchasing the primary item more than would be expected by chance.

- Support Distribution:

The plot shows the distribution of support values, which represent the frequency with which rules appear in transactions. If the majority of values are at low levels, this indicates that the rules appear in a relatively small number of transactions, but may be significant if confidence and leverage are high.



Conclusion:

We can achieve several important benefits related to analyzing data and understanding customer behavior and the products they buy from it:

1. Data cleaning and preparation for analysis:
Data filtering and cleaning: The data was cleaned and illogical or missing values (such as negative quantities or incorrect prices) were removed to ensure the accuracy of the analysis.
Data encoding: Product data was converted into a format that could be easily analyzed, such as using One-Hot Encoding to facilitate the analysis of patterns between items sold.
2. Sales analysis:
Monthly sales analysis: Monthly sales trends were analyzed using line charts, which helps to understand periods that saw an increase or decrease in sales. This can help in planning the business based on periods of high activity.
Unique item analysis: This analysis showed the variety of products sold each month, providing insights into product diversification strategies and the performance of different items over time.

3. Market Basket Analysis:
Discovering patterns between products: The Apriori algorithm was used to extract recurring groups of items

that are frequently purchased together. These groups reveal common patterns between products that can be used to improve product placement in-store or cross-promotions.
4. Extract Association Rules:
Confidence, Lift, and Support Analysis: Association rules that explain the relationship between products, such as identifying products that are most frequently purchased together (high confidence) or products that are strongly associated with each other compared to chance (high lift), are extracted. This helps in developing smart marketing strategies based on strong relationships between products.
5. Analysis of Association Rule Metrics:
Distribution Analysis: By analyzing the distributions of confidence, lift, and support metrics, we were able to understand the strength of the extracted patterns. This analysis helps in identifying the most important rules that can be used to improve business planning.
Overall Benefits:
Improved Marketing Strategies: By understanding the relationships between products, it is possible to improve

product organization in stores and target customers with promotions based on frequent purchase patterns.
Improved Inventory Management: It is possible to know which products are purchased together, which helps in improving the ordering process and inventory planning.

Increased Sales: By using association rules, related products can be suggested to customers, which can increase the opportunities for cross-selling and upselling.
Make data-driven decisions: These analytics help provide data-driven insights, allowing businesses to make more accurate and effective decisions based on an understanding of customer behavior.

جامعة أم القرى
UMM AL-QURA UNIVERSITY