IT609: BIG DATA PROCESSING
MARKET BASKET ANALYSIS FOR RETAIL STORES

GROUP MEMBERS:
1. SHREYA ARORA (202218032)
2. MUSKAN KHARE (202218037)
3. DHRUV SOLANKI (202218053)
4. JATAN SAHU (202218061)

PROBLEM STATEMENT

Market basket analysis is a technique used to identify the relationship between the products that are frequently purchased together. The problem involves identifying the frequent itemsets and association rules from a large dataset of transactions. The target is to identify the frequent patterns within a reasonable time frame.

The main goal of this problem statement is to analyse the market basket, from large volumes of transactional data efficiently in lesser time than traditional database handling tools and libraries, and provide insights into the relationships between the products.

USE-CASE OF MARKET BASKET ANALYSIS

By analyzing customer purchase behavior and identifying patterns of frequently purchased items, retailers can gain valuable insights that can help them optimize their product offerings and increase sales.

Here are some specific use cases of Market Basket Analysis:

1. **Cross-selling and Upselling:** By analyzing customer purchase history, businesses can identify complementary products that are frequently purchased together and use this information to cross-sell or upsell customers.
2. **Product Bundling:** Market Basket Analysis can help retail store owners tp identify which products should be bundled together to create attractive packages for customers. This can increase sales and help clear out excess inventory.
3. **Inventory Management:** By understanding which products are frequently purchased together, store owners can optimize their inventory management by stocking the right amount of each product.
4. **Pricing Strategy:** Market Basket Analysis can help businesses to determine the optimal price for products by analyzing the relationships between different products and how they are priced.
5. **Store Layout Optimization:** By analyzing customer purchasing patterns, businesses can optimize their store layout to encourage customers to purchase complementary products or increase sales of slow-moving products.

TOOLS AND TECHNOLOGIES USED:

In order to perform market basket analysis on a large dataset of transactions, a scalable and efficient approach is required. This can be acrhieved through PySpark and MapReduce.

PySpark is a Python library that provides an interface to Apache Spark, a powerful open-source distributed computing system. Apache Spark provides a scalable and efficient platform for processing large amounts of data, making it an ideal choice for performing Market Basket Analysis on retail transaction data.

The MapReduce framework can be used to distribute the computation of market basket analysis across a cluster of computers. The dataset can be split into smaller subsets and distributed across the cluster for parallel processing. The mapper function can then identify the frequent items in each subset of data and emit the key-value pairs for each frequent item. The reducer function can then combine the results from the mappers and identify the frequent itemsets and association rules.

DATASET DESCRIPTION

The Market Basket Analysis dataset publicly available on Kaggle has been used for this project. There are total 7 attributes and 5,35,260 rows.

The features are as follows:

- **BillNo:** 6-digit number assigned to each transaction (21663 unique values)
- **Itemname:** Product name (4187 unique values)
- **Quantity:** The quantities of each product per transaction (691 unique values)
- **Date:** The day and time when each transaction was generated (19642 unique values)
- **Price:** Product price (1268 unique values)
- **CustomerID:** 6-digit number assigned to each customer (2499 unique values)
- **Country:** Name of the country where the retail store is situated (31 unique values)

PRE-PROCESSING

Data preprocessing is an important step in the data analysis pipeline that involves transforming raw data into a format that can be used for analysis. This is done in order to improve data quality and model processing. We performed the following steps to make our dataset suitable for modelling:

1. **Removing and filling null values as required:** Columns like CustomerID had multiple null values potentially because there would be customers who wouldn't have registered with the store. These were filled by a fixed CustomerID. In the column Itemname, there were 620 null values which had to be dropped because it is an essential feature for us and cannot be null.

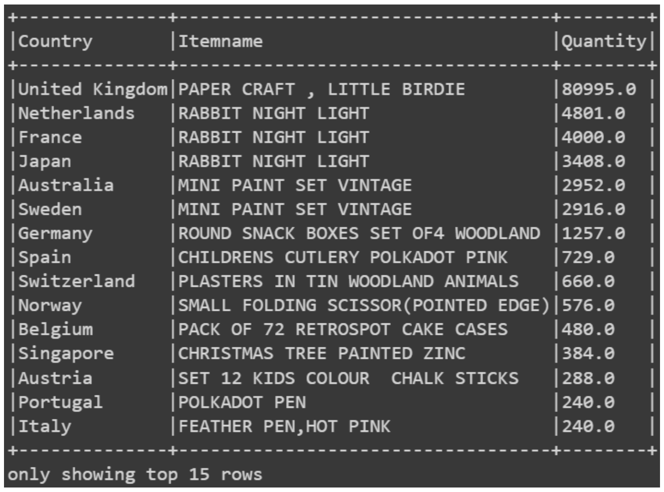
2. **Removing unnecessary rows or noise in the data:** The column Quantity had some values which were negative or zero. Practically, this is not possible. Hence, such rows were removed.
3. **Formatting certain columns in the required format:** The Date column was divided into two columns,Date and Hour, by separating date and time.

EXPLORATORY DATA ANALYSIS

EDA (Exploratory Data Analysis) is an important step in the data analysis process as it helps to understand the underlying patterns, relationships, and trends in the data. It is required to understand data distribution, select appropriate features, find outliers and detect relationships between variables.

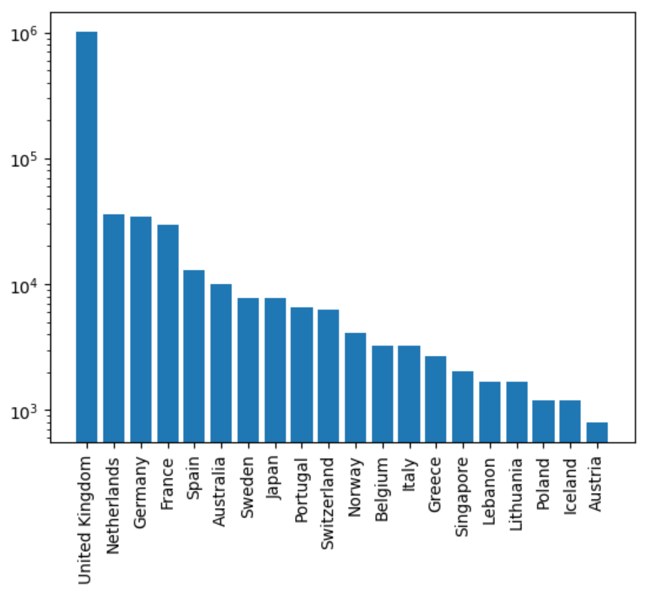
Here are some insights we have drawn from our dataset:

- The following table lists the most common item being sold in each country. Clearly, PAPER CRAFT LITTLE BIRDIE is the best selling item with an enormous quantity in United Kingdom compared to any other country's best selling item.

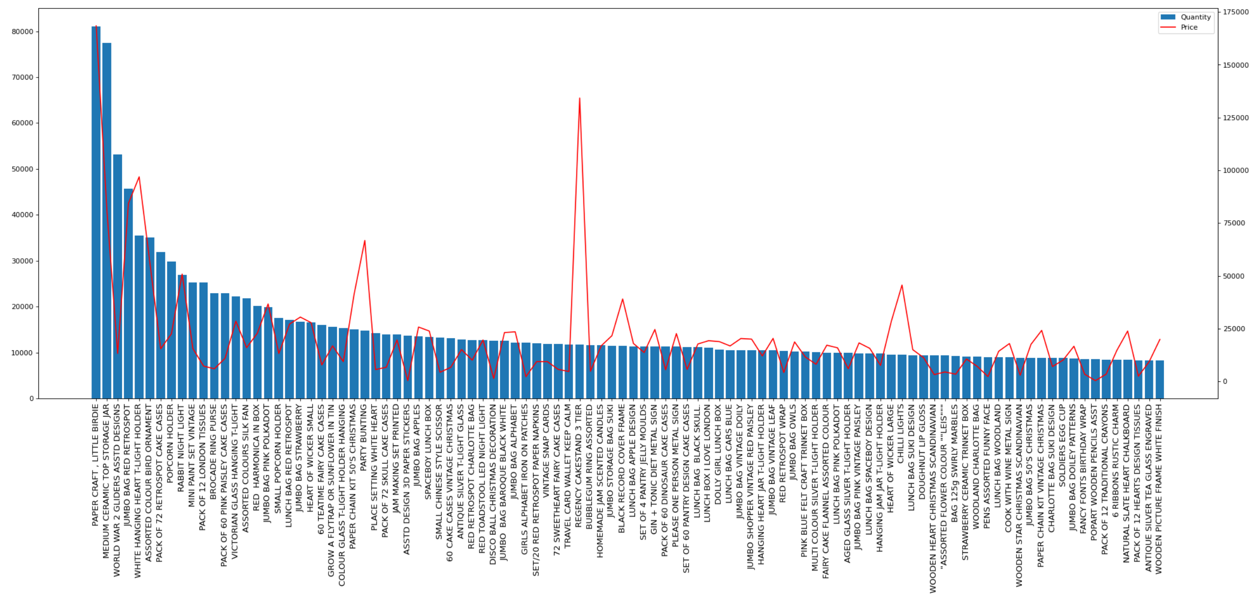


```
+--------------+----------------------------------+--------+
|Country       |Itemname                          |Quantity|
+--------------+----------------------------------+--------+
|United Kingdom|PAPER CRAFT , LITTLE BIRDIE       |80995.0 |
|Netherlands   |RABBIT NIGHT LIGHT                |4801.0  |
|France        |RABBIT NIGHT LIGHT                |4000.0  |
|Japan         |RABBIT NIGHT LIGHT                |3408.0  |
|Australia     |MINI PAINT SET VINTAGE            |2952.0  |
|Sweden        |MINI PAINT SET VINTAGE            |2916.0  |
|Germany       |ROUND SNACK BOXES SET OF4 WOODLAND |1257.0  |
|Spain         |CHILDRENS CUTLERY POLKADOT PINK   |729.0   |
|Switzerland   |PLASTERS IN TIN WOODLAND ANIMALS  |660.0   |
|Norway        |SMALL FOLDING SCISSOR(POINTED EDGE)|576.0   |
|Belgium       |PACK OF 72 RETROSPOT CAKE CASES   |480.0   |
|Singapore     |CHRISTMAS TREE PAINTED ZINC       |384.0   |
|Austria       |SET 12 KIDS COLOUR  CHALK STICKS  |288.0   |
|Portugal      |POLKADOT PEN                      |240.0   |
|Italy         |FEATHER PEN,HOT PINK              |240.0   |
+--------------+----------------------------------+--------+
only showing top 15 rows
```

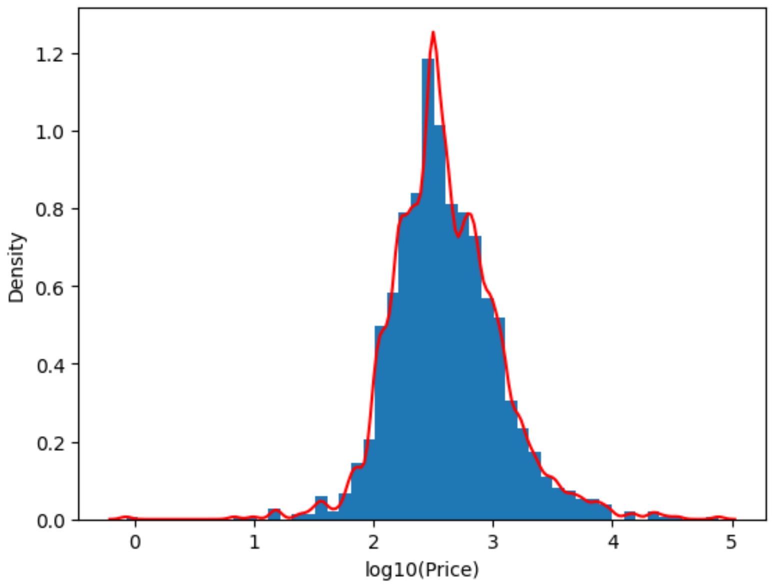
- The following graph represents the total amount earned by the store in each country. Clearly, United Kingdom has the most sales.



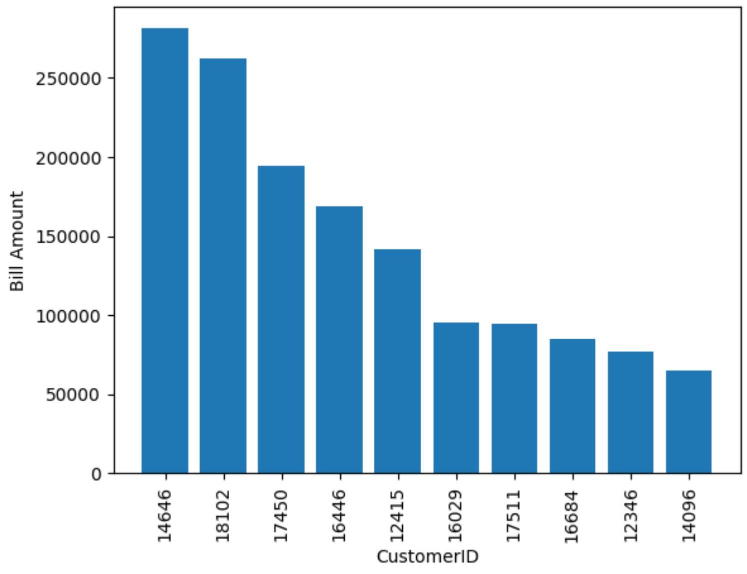
- The bars in the following chart represent the quantity of items sold while the line is the representation of amount earned by the sale of that item. There are some expensive items, like REGENCY CAKESTAND 3 TIER, which have comparatively high sales than products with less price.



- From the chart below, we can interpret that the distribution of expenditure by each customer is normal with average expenditure around 315 units. (Log scale is used in the graph.)



- The following barchart is the depiction of top 10 bill amounts. The most amount spent by a customer at once is 281047.57 currency units.



FREQUENT ITEMSET AND ASSOCIATION RULE GENERATION

There are several algorithms available for frequent itemset and association rule generation in market basket analysis. Some of the most commonly used algorithms are:

1. **Apriori Algorithm:** The Apriori Algorithm is one of the most widely used algorithms for frequent itemset and association rule generation. It works by generating a set of frequent

itemsets by finding all the sets of items that occur together in transactions with a frequency greater than or equal to a minimum support threshold. The algorithm then generates association rules from the frequent itemsets using metrics such as support, confidence, and lift.

2. **FP-Growth Algorithm:** The FP-Growth Algorithm is another popular algorithm for frequent itemset and association rule generation. It builds a frequent pattern tree (FP-tree) from the transactional data and generates a set of frequent itemsets. The association rules are then generated using support and confidence.
3. **PCY (Park-Chen-Yu) Algorithm:** The PCY (Park-Chen-Yu) algorithm is a modification of the Apriori algorithm that improves its efficiency by reducing the number of candidate itemsets that need to be generated and stored. The PCY algorithm works by first counting the frequency of individual items in the dataset and storing this information in a hash table. It then uses this hash table to identify pairs of items that occur together frequently, without generating all possible pairs.

We have implemented FP-Growth and the PCY Algorithm. We have also implemented the Apriori algorithm using MapReduce framework.

Here are some terms that need to be understood before proceeding further:

1. Antecedent: Represents the antecedent or the left-hand side of the association rule. It is a set or array of items that are present in the dataset and act as the condition or premise of the rule.
2. Consequent: Represents the consequent or the right-hand side of the association rule. It is a set or array of items that are predicted or inferred based on the presence of the antecedent.
3. Support: Indicates the fraction of transactions in the dataset that contain both the antecedent and the consequent. It is calculated as the ratio of the number of transactions.
4. Confidence: Indicates the strength of the association rule. It is a measure of how often the consequent appears in transactions that contain the antecedent. Confidence is calculated as the ratio of the support of the rule (support of both antecedent and consequent) to the support of the antecedent. Higher confidence values indicate stronger associations between the antecedent and consequent.
5. Lift: Lift is a measure of how much more likely the consequent is to appear in transactions that contain the antecedent compared to its individual occurrence. It is calculated as the ratio of the confidence of the rule to the support of the consequent. Lift values greater than 1 indicate a positive correlation between the antecedent and consequent, suggesting that the presence of the antecedent increases the likelihood of the consequent.

APRIORI ALGORITHM IMPLEMENTATION

The Apriori algorithm takes advantage of the "Apriori property," which states that any subset of a frequent itemset must also be frequent. By pruning infrequent itemsets, the algorithm reduces the number of candidate itemsets that need to be examined, making it more scalable for large datasets. We can increase efficiency of this algorithm using MapReduce framework. By leveraging the parallel processing capabilities of MapReduce, the Apriori algorithm can efficiently handle large-scale datasets and distribute the computation across multiple nodes or clusters. Here is an overview of how the Apriori algorithm can be implemented using MapReduce:

1. Map Phase:
    a. Input: Transactional data with each transaction as a separate record.
    b. Map function: The map function reads each transaction and emits key-value pairs where the key is an itemset, and the value is a count (initially set to 1) or some identifier indicating that it is a transaction.
    c. Output: Intermediate key-value pairs where the key represents an itemset and the value is either a count or an identifier.
2. Reduce Phase:
    a. Input: Intermediate key-value pairs generated by the map phase.
    b. Reduce function: The reduce function receives key-value pairs with the same key (itemset) and aggregates the values by summing the counts or processing the transactions to build a list of transactions containing the itemset.
    c. Output: Intermediate key-value pairs where the key is an itemset, and the value is either the sum of counts or the list of transactions.

FP-GROWTH ALGORITHM IMPLEMENTATION

The FP-Growth (Frequent Pattern Growth) algorithm is a popular algorithm for mining frequent itemsets and association rules from transactional data. It works by constructing an FP-tree, a compact representation of the transactional data, and then generating frequent itemsets from the FP-tree. It uses the depth-first search mechanism. Here are the steps involved in generating frequent itemsets and association rules using the FP-Growth algorithm:

1. Construct the FP-tree: Construct an FP-tree from the transactional data. Each node in the tree represents an item, and the edges between the nodes represent the transactions in which the items occur.

2. Determine the frequent items: This can be done by scanning the dataset and counting the frequency of each item, and then discarding those that do not meet the minimum support threshold.

3. Generate frequent itemsets: Recursively traverse the FP-tree and identify itemsets that occur together frequently. The algorithm merges itemsets that share the same prefix.

4. Generate association rules: Generate association rules from the frequent itemsets using metrics such as support, confidence, and lift.

The FP-Growth algorithm is an efficient algorithm for frequent itemset and association rule generation because it avoids the generation of candidate itemsets. Instead, it uses the FP-tree to identify frequent itemsets directly, which reduces the number of itemsets that need to be generated and stored. The algorithm is particularly useful when dealing with datasets with a large number of items, as it can handle such datasets more efficiently than algorithms that generate candidate itemsets.

PCY ALGORITHM IMPLEMENTATION

The PCY (Park-Chen-Yu) algorithm is a modification of the Apriori algorithm that improves its efficiency by reducing the number of candidate itemsets that need to be generated and stored. Here are the steps involved in generating frequent itemsets and association rules using the PCY algorithm:

1. Count individual items: Count the frequency of each item in the dataset and store this information in a hash table.

2. Identify frequent item pairs: Use the hash table to identify pairs of items that occur together frequently. To do this, hash each item in the transaction onto a bitmap, and then use the bitmap to count the frequency of pairs of items that hash to the same bucket.

3. Generate candidate itemsets: Generate candidate itemsets by joining pairs of frequent items that occur together frequently. Only pairs of items that have hashed to the same bucket are considered, which reduces the number of candidate itemsets that need to be generated.

4. Count the frequency of candidate itemsets and discard those that do not meet the minimum support threshold.

5. Generate association rules: Generate association rules from the frequent itemsets using metrics such as support, confidence, and lift.

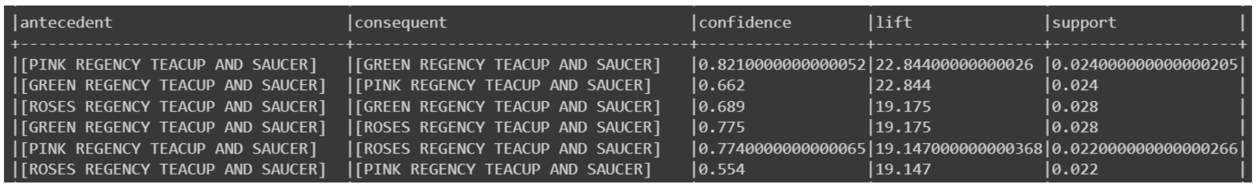
The PCY algorithm is an alternative algorithm for frequent itemset and association rule generation because it reduces the number of candidate itemsets that need to be generated and stored. By only considering pairs of items that occur together frequently, the algorithm can identify frequent itemsets more quickly than the Apriori algorithm. The PCY algorithm is particularly useful when dealing with large datasets, as it can handle datasets that are too large to fit into memory by processing them in a sequential manner.

Note: The current dataset was too large to handle the time and space complexity of PCY algorithm. Hence we implemented it on a different dataset.

RESULTS

Few of the most frequent itemsets are: {'HAND WARMER UNION JACK'}, { 'HAND WARMER RED POLKA DOT'}, { 'ASSORTED COLOUR BIRD ORNAMENT'}, {“PAPER CHAIN KIT 50'S CHRISTMAS"}, { 'WOOD S/3 CABINET ANT WHITE FINISH'}, {'VINTAGE BILLBOARD DRINK ME MUG'}, {'WOODEN PICTURE FRAME WHITE FINISH', 'GLASS STAR FROSTED T-LIGHT HOLDER'}, {'VINTAGE BILLBOARD LOVE/HATE MUG', 'RED WOOLLY HOTTIE WHITE HEART.'}, {'RETRO COFFEE MUGS ASSORTED', 'GLASS STAR FROSTED T-LIGHT HOLDER'}, {'WHITE METAL LANTERN', 'WOOD S/3 CABINET ANT WHITE FINISH'}.

The top five association rules with maximum support and confidence are:



```
|antecedent                         |consequent                         |confidence        |lift              |support            |
+-----------------------------------+-----------------------------------+------------------+------------------+-------------------+
|[PINK REGENCY TEACUP AND SAUCER]   |[GREEN REGENCY TEACUP AND SAUCER]  |0.8210000000000052|22.84400000000026 |0.02400000000000205|
|[GREEN REGENCY TEACUP AND SAUCER]  |[PINK REGENCY TEACUP AND SAUCER]   |0.662             |22.844            |0.024              |
|[ROSES REGENCY TEACUP AND SAUCER]  |[GREEN REGENCY TEACUP AND SAUCER]  |0.689             |19.175            |0.028              |
|[GREEN REGENCY TEACUP AND SAUCER]  |[ROSES REGENCY TEACUP AND SAUCER]  |0.775             |19.175            |0.028              |
|[PINK REGENCY TEACUP AND SAUCER]   |[ROSES REGENCY TEACUP AND SAUCER]  |0.7740000000000065|19.147000000000368|0.02200000000000266|
|[ROSES REGENCY TEACUP AND SAUCER]  |[PINK REGENCY TEACUP AND SAUCER]   |0.554             |19.147            |0.022              |
```