Market Basket Analysis

Astha Jain(aj887@drexel.edu), Madhu Bandru(mb4236@drexel.edu), Spandana Bendi(sb4262@drexel.edu)

*dept. Data Science, College of Computing and Informatics Drexel University*

Philadelphia, PA

***Abstract*—** **This project aims to perform “Market Basket Analysis,” a technique used by large retailers on the data, which provides the information to the retailer to understand the purchase behavior of the buyer, which can help the retailer decide. We use two algorithms, FP growth, and Apriori, to figure out the association rules using pandas, PySpark and SparkSql. The data used are retail transactions occurring between December 2010 to December 2011 in France.**

1. **INTRODUCTION**
2. *Background*

Market basket analysis is an advanced analytics technique that leverages data mining and statistical methods to increase sales by understanding customer purchasing patterns. It can be used effectively to improve the overall spending from the customer by bundling frequently purchased items at a discounted price. Large retailers use it to discover associations between their items. It works by looking for combinations of bought items together often, providing information to understand the purchase behavior. Association Rules Mining is one of the fundamental concepts of machine learning being used in Market Basket Analysis.

1. *Data Source*

The dataset used is available at *UCI Machine Learning Repository*. It contains transactional data of a year different countries and registered non-store online retail. The dataset has one Comma-Separated Value (CSV) file.

1. **METHODOLOGY**

*Association rules:[1]* Association rules help uncover all such relationships between items from huge databases. It consists of an antecedent and a consequent, both of which are a list of items.

*Confidence:* Confidence is the frequency with which X and Y are purchased together over the frequency with which X is purchased alone.

*Support:* Support is defined as the frequency with which items X and Y are purchased together over the total number of transactions. This metric tells us how frequent an itemset is in all the transactions.

*Lift*: Lift controls for the *support* (frequency) of consequent while calculating the conditional probability of occurrence of {Y} given {X}. *Lift* is a very literal term given to this measure.

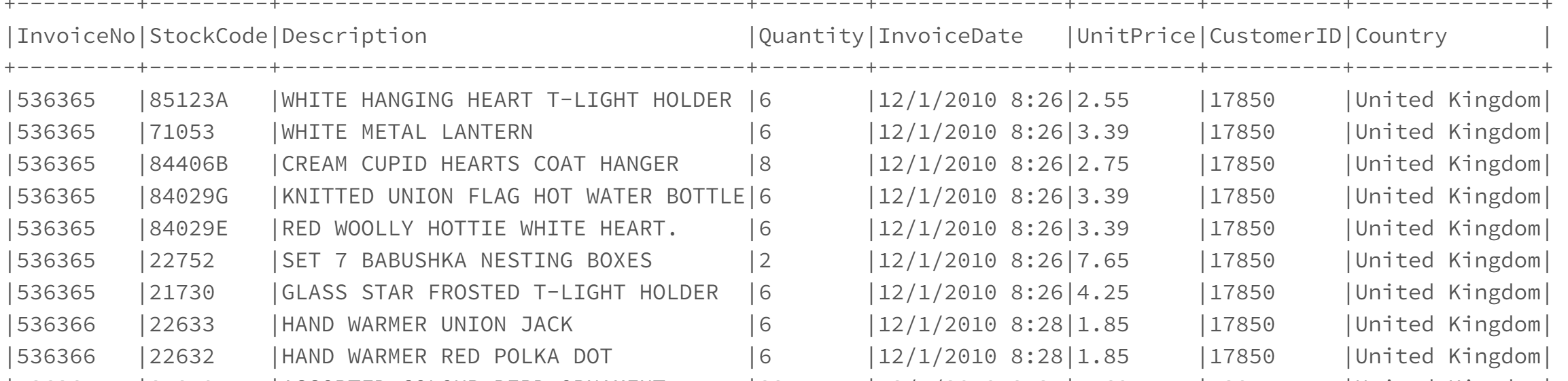
1. *FP-Growth:*

where “FP” stands for frequent pattern. Given a dataset of transactions, we need to follow following steps:

1. Calculate item frequencies and identify frequent items.
2. Use a suffix tree (FP-tree) structure to encode transactions without generating candidate sets explicitly, which are usually expensive to generate.
3. Frequent item sets can be extracted from the FP-tree.
4. Association rules are extracted from model and then input items are transformed against all the association rules and we summarize the consequents as prediction.
5. *Apriori Algorithm:*

Apriori is one of the algorithms that is used for frequent pattern mining

1. Scan the transaction data base to get the support ‘S’ each 1-itemset, compare ‘S’ with min\_sup, and get a support of 1-itemsets.
2. Use join to generate a set of candidate k-item set. Use apriori property to prune the unfrequented k-item sets from this set.
3. Scan the transaction database to get the support ‘S’ of each candidate k-item set in the given set, compare ‘S’ with min\_sup, and get a set of frequent k-item set
4. If the candidate set is NULL, for each frequent item set 1, generate all nonempty subsets of 1.
5. For every nonempty subsets of 1, output the rule “s=>(1-s)” if confidence C of the rule “s=>(1-s)” min\_conf
6. If the candidate set is not NULL, go to step 2.
7. **Description of dataset**



Data Set have 541909 rows & 8 Columns. Below are the Eight Columns.

1.**InvoiceNo**: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'C', it indicates a cancellation.

2.**StockCode**: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.

3.**Description**: Product (item) name. Nominal.

4.**Quantity**: The quantities of each product (item) per transaction. Numeric.

5.**InvoiceDate**: Invoice Date and time. Numeric, the day and time when each transaction was generated.

6.**UnitPrice**: Unit price. Numeric, Product price per unit in sterling.

7.**CustomerID**: Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.

8.**Country**: Country name. Nominal, the name of the country where each customer resides.

1. **Data analysis**

We did analysis to answer below questions:

1. What time of day do customers purchase?

We have created new column order\_hour\_of\_day and plotted the graph which shows the customers place orders between 12 pm and 3 pm.

Chart, line chart

Description automatically generated

1. Which are the top countries from which orders are placed?

Chart

Description automatically generated

From the above graph, United Kingdom has the highest orders following France, Spain, Belgium and Portugal.

1. Which are the most purchased items?

Chart, sunburst chart

Description automatically generated

The above graph represents the top 10 items out which.

White hanging heart t-light holder is most purchased items from the dataset comprising 14% of the top 10 items then jumbo bag red retro spot with 13%.

4. How many items did a customer buy in an order?

Chart, bar chart

Description automatically generated

Interestingly, InvoiceNo – 573585 has the highest items in order with approx. 1114 items which is huge. Second largest order is of 749 items.

1. **Packages used in the project**

For EDA we used Data Bricks and also implemented FP algorithm in the same. For apriori Algorithm we used Pandas, Pyspark in Google Collab. The below libraries are used to implement algorithms.

1. *pyspark.ml Package:* The main Pyspark package for all of Pyspark machine learning classes such as: Standard Scalar, Pipeline, Vector Assembler, and all classification models, FP-growth.
2. *Pandas:* Panda’s package is a powerful data frames handling package in python. It helps reading and writing CSV files quite efficiently and has many helpful functions for data frames slicing, joining and transformation.
3. **EXPERIMENT**

For implementing FP-growth, first, we would be baskets are created for each order in our dataset. We would create a baskets data frame with two columns: first, the InvoiceNo, and second, the list of items (Descriptions) bought in that particular order.

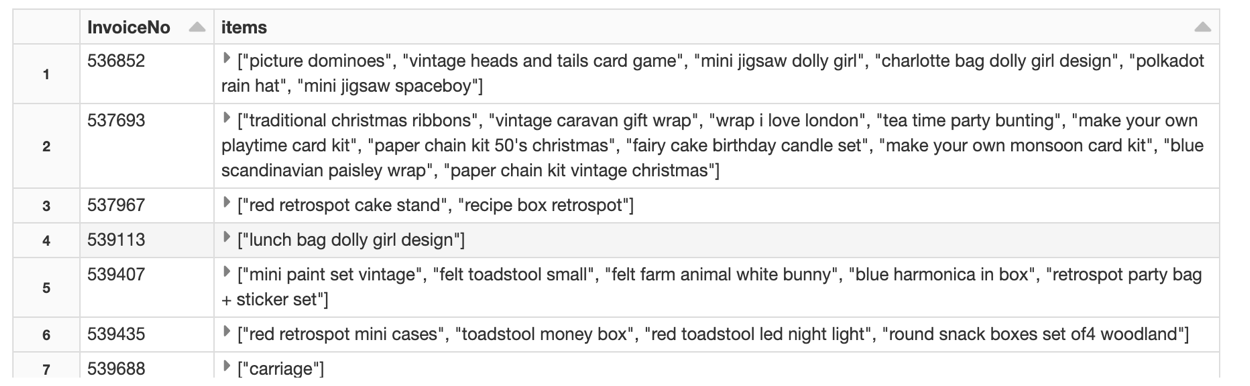
Below is a snippet of the top 5 rows of the baskets data frame fed into the FP-growth algorithm.

Graphical user interface, text, application

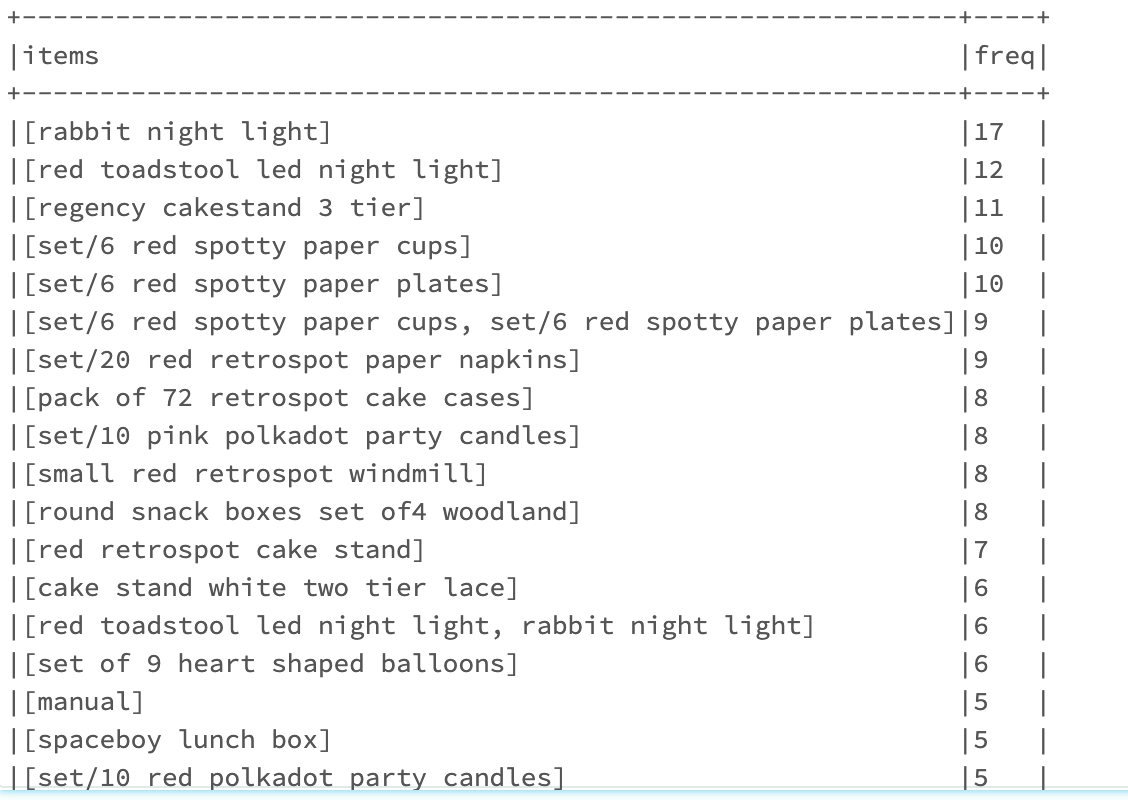
Description automatically generated

**Implementation of FP-growth algorithm**

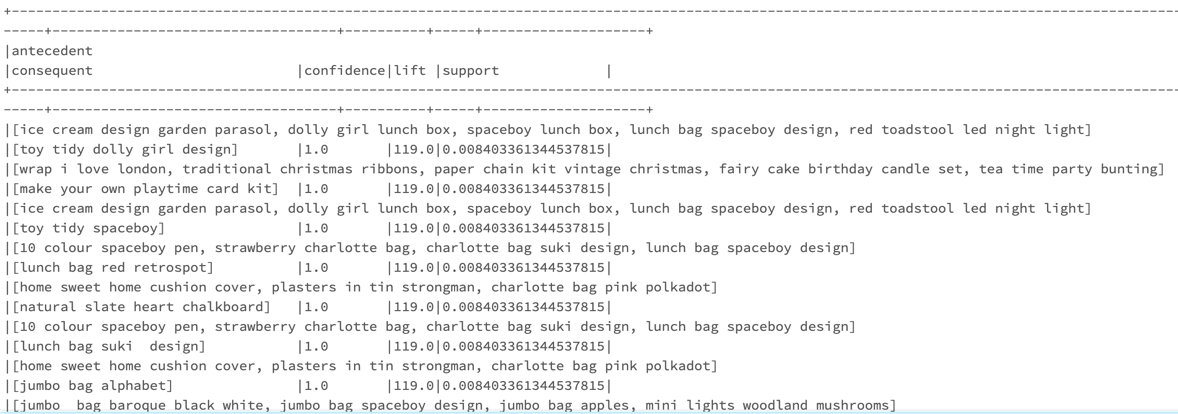
1. Taking minimum support value as 0.001 and minimum confidence value as 0. Both thresholds are user-defined. Minimum support value is selected such that there is a minimum 1 item in our item sets and it does not take much time for computation.
2. Generating frequent item-sets and building association rules.



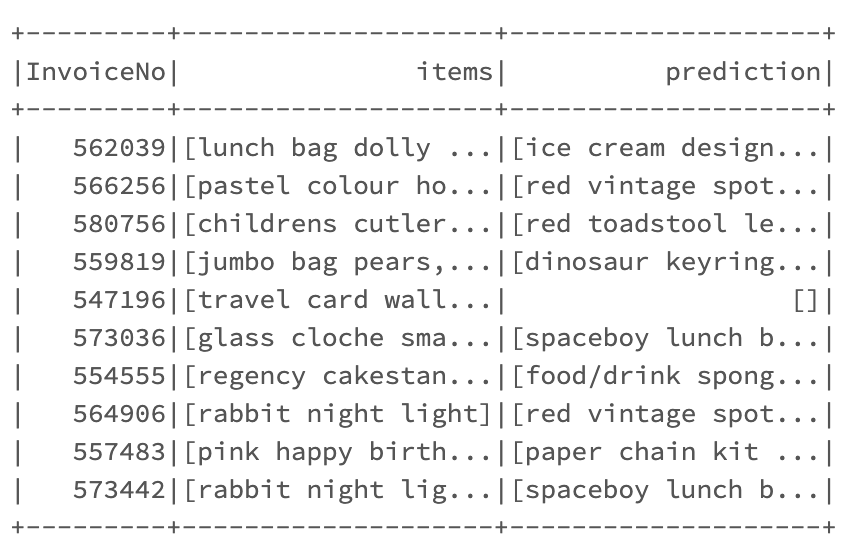
1. Exploring frequent itemset:



1. Display association rules



1. Transformed table with predictions

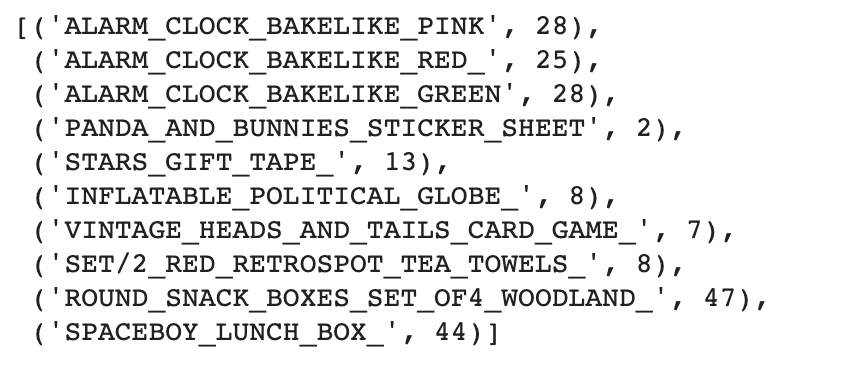


**FP-Growth Results:** From Association rules, we can see the confidence is 1.0 and lift is greater than one, which vouches for the high association between consequent and antecedent. The greater the value of the lift, the greater the chances of preference to buy antecedent if the customer has already purchased consequent. Lift is the measure that will help store managers for making decision on product placements on the aisle.

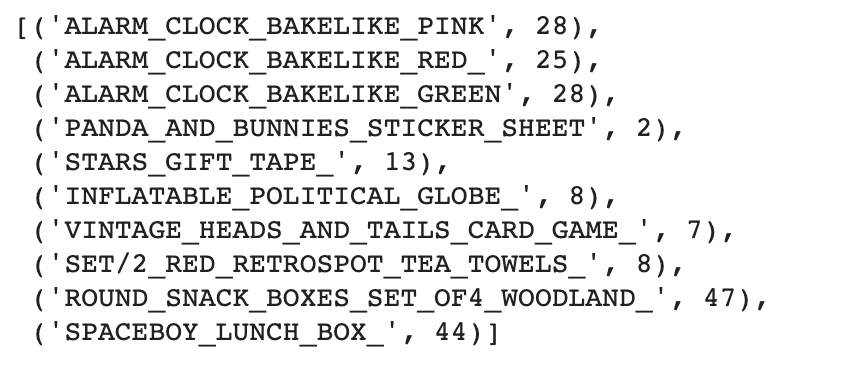
**Implementation of Apriori algorithm**

We ran Apriori for country France only as it was taking longer processing time for UK.

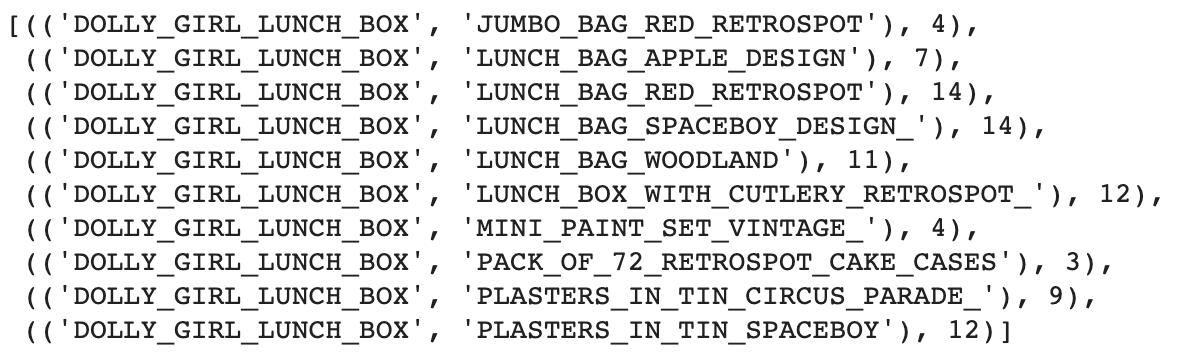
1. Reading data with PySpark and load the preprocessed data for France from text file to Spark RDD object.
2. Parsing data to Spark RDD objects



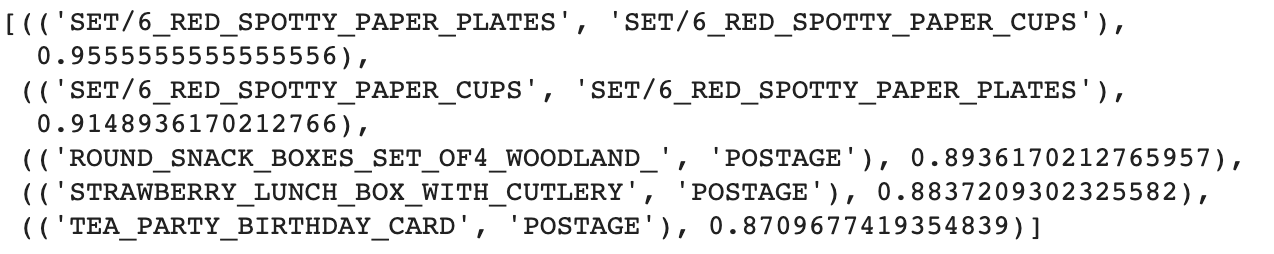
1. Filter the items which pass min support value. In our case min support value is 0.08



1. With the filtered items form two item pair set and along with frequency.



1. Calculating confidence values for two item pair set



**Apriori Algorithm Result:**

From step 5, we have items pairs and their confidence values. For 1st record, the item pair has the highest confidence value. Customers who buy “SET/6\_RED\_SPOTTY\_PAPER\_PLATES” will 95% also buy “SET/6\_RED\_SPOTTY\_PAPER\_CUPS”.

1. **CONCLUSION & FUTURE WORK**

After studying the mining process of association rules of the apriori and FP-growth algorithms, the Apriori algorithm creates all itemset by scanning the whole transactional database. On the other hand, the FP growth algorithm only makes frequent itemset based on the user's minimal support. Because Apriori scans the entire database many times, it uses more resources, and the time it takes to construct the association rules grows exponentially as the database size increases. On the other hand, the FP growth method does not scan the entire database several times, and the scanning time rises linearly. As a result, the FP growth algorithm outperforms the Apriori algorithm.

Based on associations rules and predictions from FP-Growth Algorithm, we are further interested in building a recommender system for online retail application which helps the organization sales to increase.

1. **REFERENCES**

[1] <https://towardsdatascience.com/association-rules-2-aa9a77241654>

[2] [https://en.wikipedia.org/wiki/Apriori\_algorithm](https://en.wikipedia.org/wiki/Apriori_algorithm/)

[3] <https://medium.com/analytics-vidhya/shopper-behavior-exploration-and-market-basket-analysis-using-spark-650656d6a0e1>

[4] <https://spark.apache.org/docs/latest/api/python/pyspark.sql.html>

[5] <https://matplotlib.org/index.html>

[6] <https://spark.apache.org/docs/latest/api/python/pyspark.ml.html>

[7] <https://spark.apache.org/docs/latest/ml-frequent-pattern-mining.html>

[8] <https://towardsdatascience.com/big-data-market-basket-analysis-with-apriori-algorithm-on-spark-9ab094b5ac2c>

[9] <https://dwgeek.com/mining-frequent-itemsets-apriori-algorithm.html/>