جامعـــــة أم القـــرى
UMM AL-QURA UNIVERSITY

DATA ANALYSIS

COURSE PRESENTER
(DR. Omaimah)

Online Retail project

STUDENT	ID
SHAHAD AMER	444005703
AREEJ TALEB	444002403

DEPARTMENT OF (INFORMATION SCIENCE-DATA SCIENCE)
COLLEGE OF COMPUTER AND INFORMATION SYSTEMS
UMM AL-QURA UNIVERSITY

2024

INTRODUCTION

Online retail datasets are collections of transactional data generated by e-commerce platforms. This dataset captures a wide range of information related to customer purchases, product details, and sales trends. They serve as valuable resources for analyzing consumer behaviour and informing business strategies.

OBJECTIVE

The primary goal of market basket analysis is to identify associations between products to optimize marketing strategies, enhance cross-selling opportunities, and improve inventory management.

DATA PREPARATION

There are important steps to clean and prepare a retail dataset for analysis. Here’s a clear explanation about what we did:

Dataset info: first examined the dataset to understand its structure, including the number of entries and the types of information available.

Column Renaming: rename the columns to make them clearer. For instance, changed 'StockCod' to 'product_code' and 'UnitPrice' to 'price_per_unit'.

Whitespace Removal: cleaned up the 'product_description' column by removing any extra spaces at the beginning or end of the text. This ensures that all descriptions are formatted consistently.

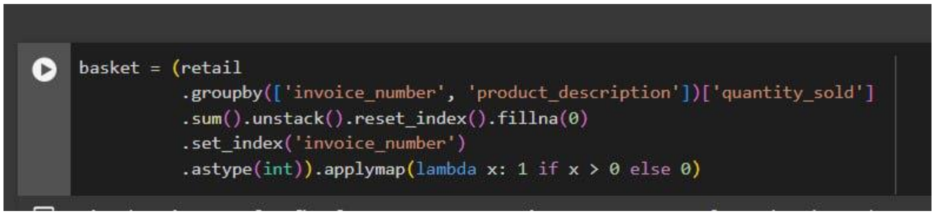
Duplicate Removal: eliminated duplicate entries from the dataset, ensuring that each transaction is represented only once.

Missing Value Handling: removed any rows that had missing values in the 'invoice_number' column. Since this field is essential for identifying transactions, it's important to have complete data here.

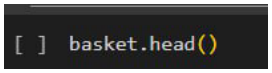
Data Type Conversion: changed the 'invoice_number' column to a string format. This is typically done to treat these identifiers correctly, as they are not used for mathematical operations.

Filtering Records: filtered out any transactions that had an 'invoice_number' containing the letter 'C', which likely indicates canceled transactions. This helps in focusing on valid transactions only.

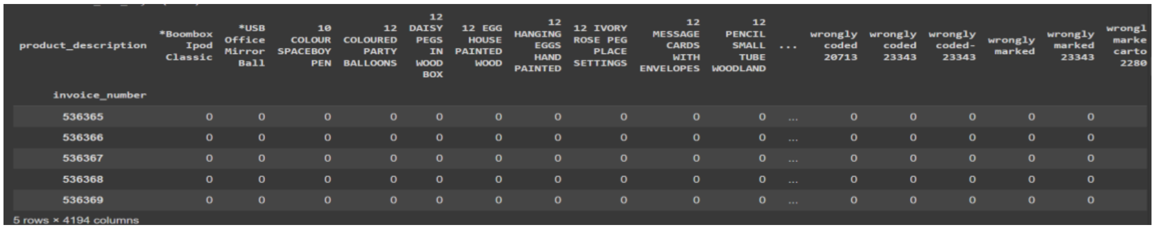
MARKET BASKET ANALYSIS



```
basket = (retail
          .groupby(['invoice_number', 'product_description'])['quantity_sold']
          .sum().unstack().reset_index().fillna(0)
          .set_index('invoice_number')
          .astype(int)).applymap(lambda x: 1 if x > 0 else 0)
```



```
[ ]  basket.head()
```



```
product_description *Boombox *USB Office 10 COLOUR 12 COLOURED 12 DAISY 12 EGG HOUSE 12 HANGING 12 IVORY ROSE 12 MESSAGE CARDS 12 PENCIL SMALL ... wrongly coded wrongly coded wrongly coded- wrongly wrongly marked wrongl marke
                    Ipod Classic Mirror Ball SPACEBOY PEN PARTY BALLOONS PEGS IN WOOD BOX PAINTED WOOD EGGS HAND PAINTED PEG PLACE SETTINGS WITH ENVELOPES TUBE WOODLAND 20713 23343 23343 marked 23343 carto 2280

invoice_number
536365               0   0   0   0   0   0   0   0   0   0 ...   0   0   0   0   0
536366               0   0   0   0   0   0   0   0   0   0 ...   0   0   0   0   0
536367               0   0   0   0   0   0   0   0   0   0 ...   0   0   0   0   0
536368               0   0   0   0   0   0   0   0   0   0 ...   0   0   0   0   0
536369               0   0   0   0   0   0   0   0   0   0 ...   0   0   0   0   0
5 rows × 4194 columns
```

This Groupby line groups the retail DataFrame by two columns: invoice_number and product_description.
It sums the quantity_sold for each combination of invoice number and product description. This step aggregates the sales data so that if multiple quantities of a product were sold in a single invoice, they are combined.

The unstack() method pivots the DataFrame, converting the unique product descriptions into columns.

This resets the index of the DataFrame, converting the current index (which is now just invoice_number) back into a regular column. This makes it easier to work with the data in subsequent steps.

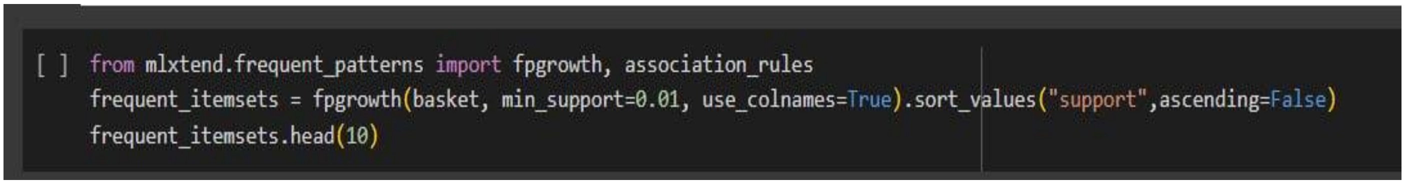
Filling NaN Values
This replaces any NaN values in the DataFrame with 0

Setting the Index
This line sets the invoice_number column as the new index of the DataFrame

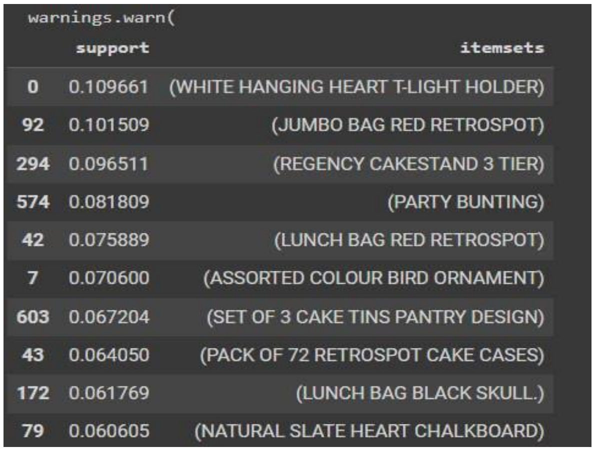
Converting to Integer Type

Applying a Lambda Function
This applies a function to every element of the DataFrame. If the quantity sold is greater than 0, it replaces it with 1 (indicating that the product was purchased in that transaction); otherwise, it replaces it with 0 (indicating that the product was not purchased).

Frequent item sets



```
[ ]  from mlxtend.frequent_patterns import fpgrowth, association_rules
     frequent_itemsets = fpgrowth(basket, min_support=0.01, use_colnames=True).sort_values("support",ascending=False)
     frequent_itemsets.head(10)
```



```
warnings.warn(
```

	support	itemsets
0	0.109661	(WHITE HANGING HEART T-LIGHT HOLDER)
92	0.101509	(JUMBO BAG RED RETROSPOT)
294	0.096511	(REGENCY CAKESTAND 3 TIER)
574	0.081809	(PARTY BUNTING)
42	0.075889	(LUNCH BAG RED RETROSPOT)
7	0.070600	(ASSORTED COLOUR BIRD ORNAMENT)
603	0.067204	(SET OF 3 CAKE TINS PANTRY DESIGN)
43	0.064050	(PACK OF 72 RETROSPOT CAKE CASES)
172	0.061769	(LUNCH BAG BLACK SKULL.)
79	0.060605	(NATURAL SLATE HEART CHALKBOARD)

applies the fpgrowth algorithm to the basket DataFrame

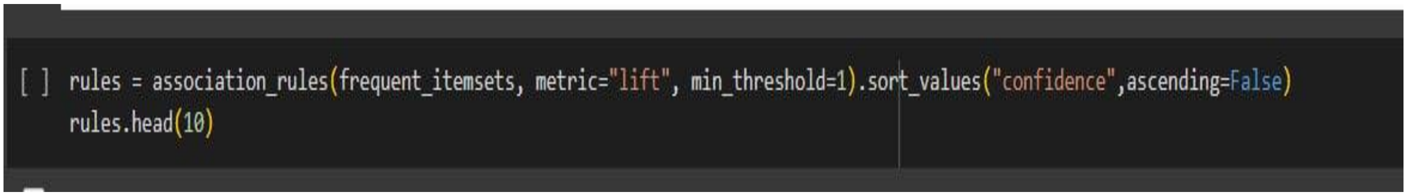
min_support=0.01 means that only itemsets that appear in at least 1% of the transactions will be considered. This threshold helps to focus on itemsets that are common enough to be significant.

use_colnames=True ensures that the output will show the actual product names instead of numerical indices.

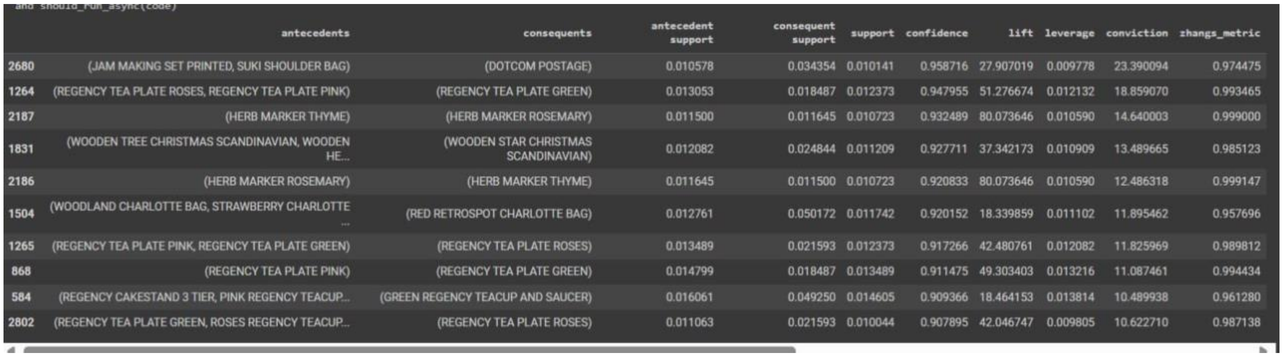
sort_values this sorts the resulting frequent itemsets by their support values . in descending order. By sorting, the most common itemsets will appear first.

The itemset with the highest support is:
(WHITE HANGING HEART T-LIGHT HOLDER) with a support value of 0.109661.
This means that this item appears in approximately 10.97% of all transactions.

Association rules



```
[ ] rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1).sort_values("confidence",ascending=False)
    rules.head(10)
```



```
and should_run_async(code)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
2680	(JAM MAKING SET PRINTED, SUKI SHOULDER BAG)	(DOTCOM POSTAGE)	0.010578	0.034354	0.010141	0.958716	27.907019	0.009778	23.390094	0.974475
1264	(REGENCY TEA PLATE ROSES, REGENCY TEA PLATE PINK)	(REGENCY TEA PLATE GREEN)	0.013053	0.018487	0.012373	0.947955	51.276674	0.012132	18.859070	0.993465
2187	(HERB MARKER THYME)	(HERB MARKER ROSEMARY)	0.011500	0.011645	0.010723	0.932489	80.073646	0.010590	14.640003	0.999000
1831	(WOODEN TREE CHRISTMAS SCANDINAVIAN, WOODEN HE...	(WOODEN STAR CHRISTMAS SCANDINAVIAN)	0.012082	0.024844	0.011209	0.927711	37.342173	0.010909	13.489665	0.985123
2186	(HERB MARKER ROSEMARY)	(HERB MARKER THYME)	0.011645	0.011500	0.010723	0.920833	80.073646	0.010590	12.486318	0.999147
1504	(WOODLAND CHARLOTTE BAG, STRAWBERRY CHARLOTTE ...	(RED RETROSPOT CHARLOTTE BAG)	0.012761	0.050172	0.011742	0.920152	18.339859	0.011102	11.895462	0.957696
1265	(REGENCY TEA PLATE PINK, REGENCY TEA PLATE GREEN)	(REGENCY TEA PLATE ROSES)	0.013489	0.021593	0.012373	0.917266	42.480761	0.012082	11.825969	0.989812
868	(REGENCY TEA PLATE PINK)	(REGENCY TEA PLATE GREEN)	0.014799	0.018487	0.013489	0.911475	49.303403	0.013216	11.087461	0.994434
584	(REGENCY CAKESTAND 3 TIER, PINK REGENCY TEACUP...	(GREEN REGENCY TEACUP AND SAUCER)	0.016061	0.049250	0.014605	0.909366	18.464153	0.013814	10.489938	0.961280
2802	(REGENCY TEA PLATE GREEN, ROSES REGENCY TEACUP...	(REGENCY TEA PLATE ROSES)	0.011063	0.021593	0.010044	0.907895	42.046747	0.009805	10.622710	0.987138

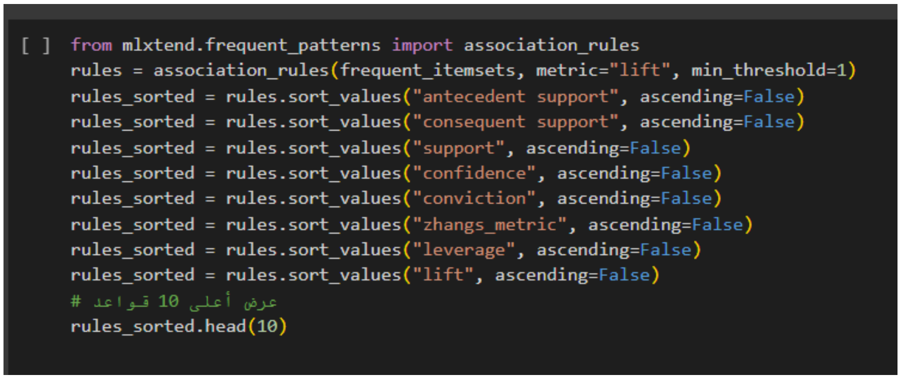
applies the association_rules function to the frequent_itemsets DataFrame

Metric: The metric="lift" parameter specifies that the rules should be evaluated based on "lift." Lift measures how much more likely two items are to be purchased together than would be expected if they were independent. A lift greater than 1 indicates a positive association between the items.

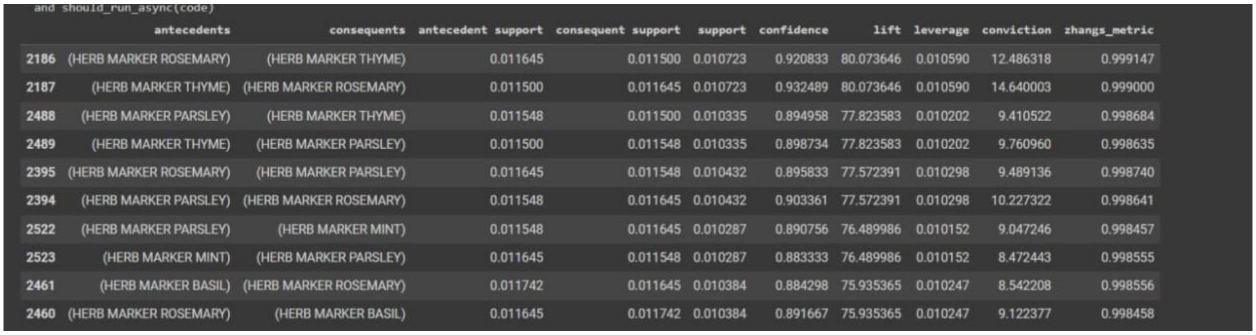
Min Threshold: The min_threshold=1 means that only rules with a lift value of 1 or higher will be included.

This sorts the resulting rules by their confidence values in descending order. Confidence is the proportion of the times the rule's consequent (the item being predicted) appears when the antecedent (the item being observed) is present. A higher confidence indicates a stronger rule.

The rule with the highest confidence is:
(JAM MAKING SET PRINTED, SUKI SHOULDER BAG) → (DOTCOM POSTAGE) with a confidence of 0.958716 this means that when customers purchase the "JAM MAKING SET PRINTED" and "SUKI SHOULDER BAG," there is a 95.87% chance that they will also purchase "DOTCOM POSTAGE."



```
[ ]  from mlxtend.frequent_patterns import association_rules
     rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
     rules_sorted = rules.sort_values("antecedent support", ascending=False)
     rules_sorted = rules.sort_values("consequent support", ascending=False)
     rules_sorted = rules.sort_values("support", ascending=False)
     rules_sorted = rules.sort_values("confidence", ascending=False)
     rules_sorted = rules.sort_values("conviction", ascending=False)
     rules_sorted = rules.sort_values("zhangs_metric", ascending=False)
     rules_sorted = rules.sort_values("leverage", ascending=False)
     rules_sorted = rules.sort_values("lift", ascending=False)
     # عرض أعلى 10 قواعد
     rules_sorted.head(10)
```



and should_run_async(code)

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
2186	(HERB MARKER ROSEMARY)	(HERB MARKER THYME)	0.011645	0.011500	0.010723	0.920833	80.073646	0.010590	12.486318	0.999147
2187	(HERB MARKER THYME)	(HERB MARKER ROSEMARY)	0.011500	0.011645	0.010723	0.932489	80.073646	0.010590	14.640003	0.999000
2488	(HERB MARKER PARSLEY)	(HERB MARKER THYME)	0.011548	0.011500	0.010335	0.894958	77.823583	0.010202	9.410522	0.998684
2489	(HERB MARKER THYME)	(HERB MARKER PARSLEY)	0.011500	0.011548	0.010335	0.898734	77.823583	0.010202	9.760960	0.998635
2395	(HERB MARKER ROSEMARY)	(HERB MARKER PARSLEY)	0.011645	0.011548	0.010432	0.895833	77.572391	0.010298	9.489136	0.998740
2394	(HERB MARKER PARSLEY)	(HERB MARKER ROSEMARY)	0.011548	0.011645	0.010432	0.903361	77.572391	0.010298	10.227322	0.998641
2522	(HERB MARKER PARSLEY)	(HERB MARKER MINT)	0.011548	0.011645	0.010287	0.890756	76.489986	0.010152	9.047246	0.998457
2523	(HERB MARKER MINT)	(HERB MARKER PARSLEY)	0.011645	0.011548	0.010287	0.883333	76.489986	0.010152	8.472443	0.998555
2461	(HERB MARKER BASIL)	(HERB MARKER ROSEMARY)	0.011742	0.011645	0.010384	0.884298	75.935365	0.010247	8.542208	0.998556
2460	(HERB MARKER ROSEMARY)	(HERB MARKER BASIL)	0.011645	0.011742	0.010384	0.891667	75.935365	0.010247	9.122377	0.998458

Generates rules using "lift" as the primary metric, with a minimum threshold of 1, indicating only rules with a meaningful association are considered

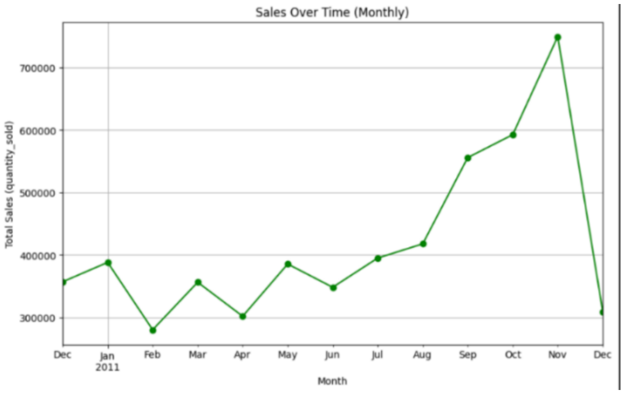
Sorts the rules multiple times based on different metrics, including:

- Antecedent support
- Consequent support
- Overall support
- Confidence
- Conviction
- Zhang's metric
- Leverage
- Finally, by lift

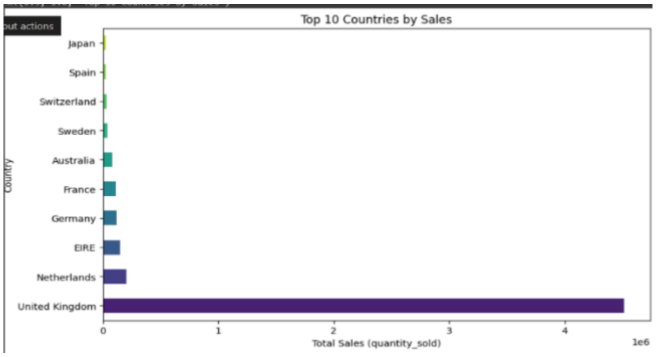
Displays the top 10 rules based on the highest lift values, which helps identify the strongest associations between items

The ultimate goal is to uncover meaningful relationships between items in the dataset, enabling businesses to make informed decisions about marketing, product placement, and promotions based on customer purchasing behavior.

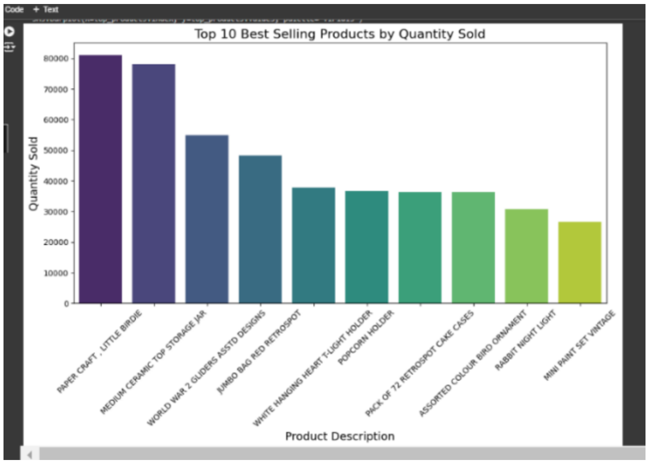
VISUALISATION



It shows the sales over time (month), the highest sales were in November.



It shows the top 10 country by sales, and the best-selling country was United Kingdom.



It shows the top 10 selling products by quantity sold, the most selling product was paper craft, little bride.