

## ▼ OEIT6 - Data Analytics

Experiment 7: Apriori Algorithm and Association rule mining with WEKA

Name: Gaurav Panchal

UID: 2019120046

```
import numpy as np
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
```

```
# Loading the Data
data = pd.read_csv('./Online Retail.csv')
data.head()
```

|   | InvoiceNo | StockCode | Description                         | Quantity | InvoiceDate    | UnitPrice | CustomerID | Country        |
|---|-----------|-----------|-------------------------------------|----------|----------------|-----------|------------|----------------|
| 0 | 536365    | 85123A    | WHITE HANGING HEART T-LIGHT HOLDER  | 6        | 12/1/2010 8:26 | 2.55      | 17850.0    | United Kingdom |
| 1 | 536365    | 71053     | WHITE METAL LANTERN                 | 6        | 12/1/2010 8:26 | 3.39      | 17850.0    | United Kingdom |
| 2 | 536365    | 84406B    | CREAM CUPID HEARTS COAT HANGER      | 8        | 12/1/2010 8:26 | 2.75      | 17850.0    | United Kingdom |
| 3 | 536365    | 84029G    | KNITTED UNION FLAG HOT WATER BOTTLE | 6        | 12/1/2010 8:26 | 3.39      | 17850.0    | United Kingdom |
| 4 | 536365    | 84029E    | RED WOOLLY HOTTIE WHITE HEART.      | 6        | 12/1/2010 8:26 | 3.39      | 17850.0    | United Kingdom |

```
# Exploring the columns of the data
data.columns

Index(['InvoiceNo', 'StockCode', 'Description', 'Quantity', 'InvoiceDate',
      'UnitPrice', 'CustomerID', 'Country'],
      dtype='object')
```

```
# Exploring the different regions of transactions
data.Country.unique()

array(['United Kingdom', 'France', 'Australia', 'Netherlands', 'Germany',
      'Norway', 'EIRE', 'Switzerland', 'Spain', 'Poland', 'Portugal',
      'Italy', 'Belgium', 'Lithuania', 'Japan', 'Iceland',
      'Channel Islands', 'Denmark', 'Cyprus', 'Sweden', 'Austria',
      'Israel', 'Finland', 'Bahrain', 'Greece', 'Hong Kong', 'Singapore',
      'Lebanon', 'United Arab Emirates', 'Saudi Arabia',
      'Czech Republic', 'Canada', 'Unspecified', 'Brazil', 'USA',
      'European Community', 'Malta', 'RSA'], dtype=object)
```

### Cleaning the Data

```
# Stripping extra spaces in the description
data['Description'] = data['Description'].str.strip()

# Dropping the rows without any invoice number
data.dropna(axis = 0, subset = ['InvoiceNo'], inplace = True)
data['InvoiceNo'] = data['InvoiceNo'].astype('str')

# Dropping all transactions which were done on credit
data = data[~data['InvoiceNo'].str.contains('C')]
```

### Splitting the data according to the region of transaction

```
# Transactions done in France
basket_France = (data[data['Country'] == "France"]
                 .groupby(['InvoiceNo', 'Description'])['Quantity']
                 .sum().unstack().reset_index().fillna(0)
                 .set_index('InvoiceNo'))

# Transactions done in the United Kingdom
basket_UK = (data[data['Country'] == "United Kingdom"]
             .groupby(['InvoiceNo', 'Description'])['Quantity']
             .sum().unstack().reset_index().fillna(0)
             .set_index('InvoiceNo'))

# Transactions done in Portugal
```

```

basket_Por = (data[data['Country'] == "Portugal"]
              .groupby(['InvoiceNo', 'Description'])['Quantity']
              .sum().unstack().reset_index().fillna(0)
              .set_index('InvoiceNo'))

basket_Sweden = (data[data['Country'] == "Sweden"]
                 .groupby(['InvoiceNo', 'Description'])['Quantity']
                 .sum().unstack().reset_index().fillna(0)
                 .set_index('InvoiceNo'))

```

## Hot encoding the Data

```

# Defining the hot encoding function to make the data suitable
# for the concerned libraries
def hot_encode(x):
    if(x<= 0):
        return 0
    if(x>= 1):
        return 1

```

```

# Encoding the datasets
basket_encoded = basket_France.applymap(hot_encode)
basket_France = basket_encoded

```

```

basket_encoded = basket_UK.applymap(hot_encode)
basket_UK = basket_encoded

```

```

basket_encoded = basket_Por.applymap(hot_encode)
basket_Por = basket_encoded

```

```

basket_encoded = basket_Sweden.applymap(hot_encode)
basket_Sweden = basket_encoded

```

## Building the models and analyzing the results

a) France:

```

# Building the model
frq_items = apriori(basket_France, min_support = 0.05, use_colnames = True)

```

```

# Collecting the inferred rules in a dataframe
rules = association_rules(frq_items, metric = "lift", min_threshold = 1)
rules = rules.sort_values(['confidence', 'lift'], ascending = [False, False])
rules.head()

```

c:\Users\Gaurav\AppData\Local\Programs\Python\Python310\lib\site-packages\mlxtend\frequent\_patterns\fpcommon.py:111: DeprecationWarning: warnings.warn()

|     | antecedents  | consequents                     | antecedent support | consequent support | support  | confidence | lift     | leverage | conviction |
|-----|--|---------------------------------|--------------------|--------------------|----------|------------|----------|----------|------------|
| 45  | (JUMBO BAG WOODLAND ANIMALS)                       | (POSTAGE)                       | 0.076531           | 0.765306           | 0.076531 | 1.000      | 1.306667 | 0.017961 | inf        |
| 259 | (PLASTERS IN TIN CIRCUS PARADE, RED TOADSTOOL ...) | (POSTAGE)                       | 0.051020           | 0.765306           | 0.051020 | 1.000      | 1.306667 | 0.011974 | inf        |
| 272 | (PLASTERS IN TIN WOODLAND ANIMALS, RED TOADSTO...) | (POSTAGE)                       | 0.053571           | 0.765306           | 0.053571 | 1.000      | 1.306667 | 0.012573 | inf        |
| 301 | (SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED     | (SET/6 RED SPOTTY PAPER PLATES) | 0.102041           | 0.127551           | 0.099490 | 0.975      | 7.644000 | 0.086474 | 34.897959  |

## Conclusion A:

From the above output, it can be seen that paper cups and paper and plates are bought together in France. This is because the French have a culture of having a get-together with their friends and family atleast once a week. Also, since the French government has banned the use of plastic in the country, the people have to purchase the paper-based alternatives.

b) Portugal:

```

frq_items = apriori(basket_Por, min_support = 0.05, use_colnames = True)
rules = association_rules(frq_items, metric = "lift", min_threshold = 1)
rules = rules.sort_values(['confidence', 'lift'], ascending = [False, False])
rules.head()

```

c:\Users\Gaurav\AppData\Local\Programs\Python\Python310\lib\site-packages\mlxtend\frequent\_patterns\fpcommon.py:111: DeprecationWarning: warnings.warn(

|      | antecedents                        | consequents                        | antecedent support | consequent support | support  | confidence | lift      | leverage | conviction |
|------|------------------------------------|------------------------------------|--------------------|--------------------|----------|------------|-----------|----------|------------|
| 1170 | (SET 12 COLOUR PENCILS SPACEBOY)   | (SET 12 COLOUR PENCILS DOLLY GIRL) | 0.051724           | 0.051724           | 0.051724 | 1.0        | 19.333333 | 0.049049 | inf        |
| 1171 | (SET 12 COLOUR PENCILS DOLLY GIRL) | (SET 12 COLOUR PENCILS SPACEBOY)   | 0.051724           | 0.051724           | 0.051724 | 1.0        | 19.333333 | 0.049049 | inf        |
| 1172 | (SET OF 4 KNICK KNACK TINS LONDON) | (SET 12 COLOUR PENCILS DOLLY GIRL) | 0.051724           | 0.051724           | 0.051724 | 1.0        | 19.333333 | 0.049049 | inf        |
| 1173 | (SET 12 COLOUR PENCILS DOLLY GIRL) | (SET OF 4 KNICK KNACK TINS LONDON) | 0.051724           | 0.051724           | 0.051724 | 1.0        | 19.333333 | 0.049049 | inf        |

## Conclusion B:

On analyzing the association rules for Portuguese transactions, it is observed that Tiffin sets (Knick Knack Tins) and color pencils. These two products typically belong to a primary school going kid. These two products are required by children in school to carry their lunch and for creative work respectively and hence are logically make sense to be paired together.

c) Sweden:

```

frq_items = apriori(basket_Sweden, min_support = 0.05, use_colnames = True)
rules = association_rules(frq_items, metric = "lift", min_threshold = 1)
rules = rules.sort_values(['confidence', 'lift'], ascending = [False, False])
rules.head()

```

c:\Users\Gaurav\AppData\Local\Programs\Python\Python310\lib\site-packages\mlxtend\frequent\_patterns\fpcommon.py:111: DeprecationWarning: warnings.warn(

|     | antecedents                   | consequents                   | antecedent support | consequent support | support  | confidence | lift | leverage | conviction |
|-----|-------------------------------|-------------------------------|--------------------|--------------------|----------|------------|------|----------|------------|
| 0   | (PACK OF 72 SKULL CAKE CASES) | (12 PENCILS SMALL TUBE SKULL) | 0.055556           | 0.055556           | 0.055556 | 1.0        | 18.0 | 0.052469 | inf        |
| 1   | (12 PENCILS SMALL TUBE SKULL) | (PACK OF 72 SKULL CAKE CASES) | 0.055556           | 0.055556           | 0.055556 | 1.0        | 18.0 | 0.052469 | inf        |
| 4   | (ASSORTED BOTTLE TOP MAGNETS) | (36 DOILIES DOLLY GIRL)       | 0.055556           | 0.055556           | 0.055556 | 1.0        | 18.0 | 0.052469 | inf        |
| 5   | (36 DOILIES DOLLY GIRL)       | (ASSORTED BOTTLE TOP MAGNETS) | 0.055556           | 0.055556           | 0.055556 | 1.0        | 18.0 | 0.052469 | inf        |
| 100 | (CHILDRENS CUTLERY)           | (CHILDRENS CUTLERY)           | 0.055556           | 0.055556           | 0.055556 | 1.0        | 18.0 | 0.052469 | inf        |

## Conclusion C:

On analyzing the above rules, it is found that boys' and girls' cutlery are paired together. This makes practical sense because when a parent goes shopping for cutlery for his/her children, he/she would want the product to be a little customized according to the kid's wishes.

## Inference:

There are three major components of the Apriori algorithm which are as follows.

1. Support
2. Confidence
3. Lift

The Apriori algorithm advantages are as follows:

1. The resulting rules are intuitive and easy to communicate to an end-user

2. It doesn't require labeled data as it is fully unsupervised; as a result, you can use it in many different situations because unlabeled data is often more accessible
3. Many extensions were proposed for different use cases based on this implementation—for example, there are association learning algorithms that take into account the ordering of items, their number, and associated timestamps
4. The algorithm is exhaustive, so it finds all the rules with the specified support and confidence.

What are the disadvantages of Apriori Algorithm?

One of the biggest limitations of the Apriori Algorithm is that it is slow. This is so because of the bare decided by the:

1. A large number of itemsets in the Apriori algorithm dataset.
2. Low minimum support in the data set for the Apriori algorithm.
3. The time needed to hold a large number of candidate sets with many frequent itemsets.
4. Thus it is inefficient when used with large volumes of datasets.

Many methods are available for improving the efficiency of the algorithm.

1. Hash-Based Technique.
2. Transaction Reduction.
3. Partitioning.
4. Sampling.
5. Dynamic Itemset Counting.