

You get a client who runs a retail store. Your client gives you data for all transactions that consists of items bought in the store by several customers over a period of time and asks you to use that data to help boost their business. Your client will use your findings to not only change/update/add items in inventory but also use them to change the layout of the physical store or rather an online store.

Clients can use rules for numerous marketing strategies:

- Changing the store layout according to trends
  - Customer behavior analysis
  - Catalogue design
  - Cross marketing on online stores
  - What are the trending items customers buy
  - Customized emails with add-on sales
- 
- You will use a dataset from the [UCI Machine Learning Repository](#). The dataset is called **Online-Retail**, and you can download it from [here](#).
- 
- **Data Pre-processing**
  - Convert *dataframe* into transaction data so that all items that are bought together in one invoice are in one row.  
Hint: group data in the dataframe either by CustomerID, CustomerID, and Date or InvoiceNo and Date
  - Store (only items) transaction data into a **.csv** file.
  - Display total number of transactions and items of transaction data.
  - Get a total number of transactions with 1 item, 2 items , 3 items and 8 items.
  - Get the number of items in the longest transaction

- **Generate Associate Rules**
- Min Support as 0.001, confidence as 0.8
- Display total number of rules with 10 items as max of items in a rule.
- Print only top 10 rules in following format
- A -> B [support=s, confidence=c, lift=l]
- Get all rules with 100% confidence
- Get rules with max length of 3
- Find what customers buy before buying 'METAL'
- Find the answer to the question *Customers who bought METAL also bought....*

```
"""AssociationRule.ipynb
```

```
# Apriori
# Importing the libraries
"""
```

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

```
"""# Data Preprocessing"""
```

```
dataset = pd.read_csv('Market_Basket_Optimisation.csv', header =
None)
dataset.shape
```

```
dataset.head()
```

```
transactions = []
for i in range(0, 7501):
    transactions.append([str(dataset.values[i,j]) for j in range(0,
20)])
transactions[0]
```

```

"""# Training the Apriori model on the dataset"""

pip install apyori

from apyori import apriori
rules = apriori(transactions = transactions, min_support = 0.003,
min_confidence = 0.2, min_lift = 4, min_length = 2, max_length =
3)

"""# Visualising the results

## Displaying the first results coming directly from the output
of the apriori function
"""

results = list(rules)
print("There are {} RULES derived.".format(len(results)))

results

for i in range(0, len(results)):
    print(results[i][0])

# first index of the inner list
for item in results :
    pair=item[0]
    #print(pair)
    items=[x for x in pair]
    #print (items)
    print(items[0]+'->' +items[1])
    # second index of the inner list
    print("Support: " + str(item[1]))
    # third index of the list located at 0th
    # of the third index of the inner list

    print("Confidence: " + str(item[2][0][2]))
    print("Lift: " + str(item[2][0][3]))
    print("=====")

def inspect(results):
    lhs          = [tuple(result[2][0][0])[0] for result in
results]
    rhs          = [tuple(result[2][0][1])[0] for result in
results]
    supports     = [result[1] for result in results]
    confidences  = [result[2][0][2] for result in results]
    lifts       = [result[2][0][3] for result in results]
    return list(zip(lhs, rhs, supports, confidences, lifts))
resultsinDataFrame = pd.DataFrame(inspect(results), columns =
['Left Hand Side', 'Right Hand Side', 'Support', 'Confidence',
'Left'])

```

```

"""## Displaying the results non sorted"""

resultsinDataFrame

"""## Displaying the results sorted by descending lifts"""

resultsinDataFrame.nlargest(n = 5, columns = 'Lift')


# Apriori in R

# Data Preprocessing
# install.packages('arules')
library(arules)
dataset = read.csv('Market_Basket_Optimisation.csv', header =
FALSE)
dataset = read.transactions('Market_Basket_Optimisation.csv', sep
= ',', rm.duplicates = TRUE)
summary(dataset)
itemFrequencyPlot(dataset, topN = 10)

# Training Apriori on the dataset
rules = apriori(data = dataset, parameter = list(support = 0.004,
confidence = 0.2))

# Visualising the results
inspect(sort(rules, by = 'lift')[1:10])

```