

# DIRE DAWA UNIVERSITY INSTITUTE OF TECHNOLOGY SCHOOL OF COMPUTING

#### DEPARTMENT OF COMPUTER SCIENCE

Post Graduate Program (Msc) in Computer Science

**Course Title: Data Mining and Warehousing** 

Course Code: COSC 623
Assignment 3: Association Rule Mining

**Course Assignment** 

By

Full Name	ID
Tadesse Shefera	DDU1400693

Submitted to: Gaddisa Olani. (PhD)

## Contents

Overview of given dataset	1
Preprocessing	
Split every data item in a single transaction to evaluate individually	
Encoding	
Frequent itemset by support	2
Frequent itemset by support and confidence	2
Formatting result based on Assignment instruction	3
Summery	3

### Overview of given dataset

The given data was 1000 instance with single column. It also in text format. Even a single transaction has multiple items, all items treated as a single data value in pandas. So, we must split every data item as single value in one transaction.

```
In [7]: import pandas as pd
    df = pd.read_table('supermarketdata.txt', header=None)
    df.shape
Out[7]: (1000, 1)
```

## **Preprocessing**

Split every data item in a single transaction to evaluate individually

```
For single transaction these 5 items considered as single value.

1 11 12 21 64 87

2 16 45 55 64 66

There is space between data items and we use it for splitting purpose
```

```
In [15]: # store all 1000 transaction in all transaction
          all_transaction=[]
          for i in range (total transaction):
              single transaction=df.iloc[i][0]
              single_transaction_splited=single_transaction.split( )
              all_transaction.append(single_transaction_splited)
          all transaction
Out[15]: [['5', '15', '32', '61', '78'],
           ['11', '12', '21', '64', '87'],
           ['16', '45', '55', '64', '66'],
['20', '51', '55', '68', '74'],
            '8', '19', '24', '31', '95'],
             '25', '40', '49', '58', '97'],
            '16', '42', '56', '71',
                                      '73'],
            '14', '38', '50', '68', '82'],
                 '14',
                       '37',
            '17', '39', '50', '75', '79'],
           ['24', '28', '56', '60', '62'],
```

Now we have list of all transaction with list

#### **Encoding**

We try to encode list of transaction using "TransactionEncoder" and convert it to data frame

```
In [24]: #import necessary library and encode
    from mlxtend.preprocessing import TransactionEncoder
    from mlxtend.frequent_patterns import apriori
    transaction_encoder = TransactionEncoder()
    transaction_array = transaction_encoder.fit(all_transaction).transform(all_transaction)
    df = pd.DataFrame(transaction_array, columns=transaction_encoder.columns_)
```

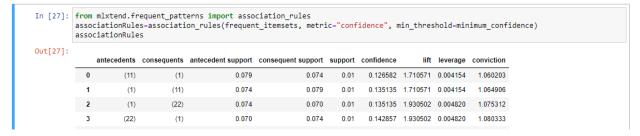
## Frequent itemset by support

Using apriori algorithm we try to find the association which satisfy the minimum support that accepted from user.



## Frequent itemset by support and confidence

By using the above support result we are going to find association that satisfy minimum confidence that accepted from user



## Formatting result based on Assignment instruction

For report purpose we need 4 attributes from above result (antecedents, consequents, support and confidence)

```
In [9]:
         =>from above dataframe We need only 4 variables [antecedents, consequents, suppo
         =>Then we are going to filiter it
         ass_rule=associationRules[['antecedents','consequents','support','confidence']]
         ass_rule.head(5)
Out[9]:
            antecedents consequents support confidence
         0
                   (72)
                                      0.013
                                              0.213115
                   (16)
                               (98)
                                              0.217391
          1
                                      0.015
                   (26)
                                      0.017
                                              0.217949
          3
                   (41)
                               (26)
                                      0.017
                                              0.202381
```

To access frozenset it should be convert to list and we try to display in the following format.

```
In [29]: #to access frozenset we try convert to list
      ass_rule=ass_rule.values.tolist()#run this line only one time
      print("#######################")
                 \t\tConfidence\t\tSupport")
      print("Rule
      print("#################################")
      for i in range (len(ass rule)):
         print(f"{list(ass_rule[i][0])[0]} ==> {list(ass_rule[i][1])[0]}",
              f"\t\t({round(ass_rule[i][3]*100,1)}%)",
              f"\t\t({round(ass_rule[i][2]*100,1)}%)")
      Rule
                        Confidence
                                         Support
      72 ==> 1
                        (21.3\%)
                                         (1.3\%)
      16 ==> 98
                        (21.7\%)
                                         (1.5\%)
      26 ==> 41
                        (21.8\%)
                                         (1.7\%)
      41 ==> 26
                        (20.2\%)
                                         (1.7\%)
```

## **Summery**

In this assignment we apply different data preprocessing and we try to find association rule for given supermarket dataset by accepting minimum support and minimum confidence from user.

From given dataset we observe that the existence of data item in many transactions is less. And we decided the strong rule is rules is "26->41" with 1.7% support and 21.8% confidence.