BITS Pilani, K.K Birla Goa Campus 2019-2020

Name of the student : Pranav R Kalikate

ID : 2018H1410076G

Title :Market Basket Optimisation using ARL

Project : Market Basket Optimisation Using ARL

Association rule learning (ARL)

Association rule learning finds interesting associations and relationships among large sets of data items. This rule shows how frequently an item-set occurs in a transaction. A typical example is Market Based Analysis.

Market Based Analysis is one of the key techniques used by large relations to show associations between items.It allows retailers to identify relationships between the items that people buy together frequently.Given a set of transactions, we can find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.

Example suppose a person wants to buy a specific product.Let's call it product A.And this product A can be associated very well to another product B and the person who wants to buy-the product A might think of this good association between the product A and the product B.Well if you place the product A and the product B be next to each other.Well the association can suddenly pop up in the buyer's mind. And buyer will consider this as good association.And so again even if the buyer was originally meant to buy only the product A, he will end up buying product B as well.So that's the idea of how we can create added value for retail stores or grocery stores.

You can think of an online store.You know these recommendations people who bought this also about that.Well these recommendations are based on association rules as well

Dataset : Market_Basket_Optimisation.csv

Steps :

Step 1:- Importing the Libraries

```python

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

```

FIG. 1

Step 2:- Importing The Dataset

Dataset contains 7500 transactions of a store for 20 products in a store.
So based on all these 7500 transactions our machine leaning model or apriori model is going to learn the different associations it can make to actually understand the rules. Such as if customers buy this product then they're likely to buy this other set of products.

```python
# Data Preprocessing
dataset = pd.read_csv('Market_Basket_Optimisation.csv',header=None)
transactions = []
for i in range(0, 7501):
    transactions.append([str(dataset.values[i,j]) for j in range(0, 20)])

```

FIG 2

Apriori expects the input to be list of list. That is a list containing the different transactions each one put in a list.

Step 3:- Training Apriori model on the dataset

```python
# Training Apriori on the dataset
from apyori import apriori
rules = apriori(transactions, min_support = 0.003, min_confidence = 0.2, min_lift = 3, min_length = 2)

```

FIG 3

The support:

The support of a set of items I is equal to the number of transactions contained in this set of items I divided by the total number of transactions. Support argument I have put here is the min_support .That means that the items that are going to appear in your rules will have a higher support than this minimum support here.
Let's find out the support value considering that the minimum three products purchased in a transactions. (we want our rules to consider only the products that are in transactions at least three times a day.). That means 3*7 days in a week =21 times a products will appear in the transactions in a week. Divide this value with total number of transactions in a dataset to get the min_support.
min_support= (3*7)/7500=0.003
So only those transactions which have a support value higher than the min_support will be considered for getting a rules.So all the products of our rules will have a higher support than their min_support here that I've obtained above.

The minimum confidence:

Minimum confidence of 0.2 or 20 percent associated to a minimum support of 0.03 is actually a good combination and gave us relevant rules.

min_length : min no. of product we want for association in basket.

These values depends on your business problem or the number of observations in your data set.That is the number of your transactions.

Step 4:- Visualise the results

From the above association rules, consider top rules for getting better idea regarding the association between the different transactions & based on these associations, place the items nearer/away from each other in the store so as to improve the items purchase.

```
In [2]: results
Out[2]:
[RelationRecord(items=frozenset({'light cream', 'chicken'}), support=0.004532728969470737,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'light cream'}), items_add=frozenset({'chicken'}),
confidence=0.2905982905982905 7, lift=4.84395061728395)]),
 RelationRecord(items=frozenset({'mushroom cream sauce', 'escalope'}), support=0.005732568990801226,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'mushroom cream sauce'}), items_add=frozenset({'escalope'}),
confidence=0.3006993006993007, lift=3.790832696715049)]),
 RelationRecord(items=frozenset({'pasta', 'escalope'}), support=0.005865884548726837,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'pasta'}), items_add=frozenset({'escalope'}),
confidence=0.3728813559322034, lift=4.700811850163794)]),
 RelationRecord(items=frozenset({'fromage blanc', 'honey'}), support=0.003332888948140248,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'fromage blanc'}), items_add=frozenset({'honey'}),
confidence=0.2450980392156863, lift=5.164270764485569)]),
 RelationRecord(items=frozenset({'ground beef', 'herb & pepper'}), support=0.015997866951073192,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'herb & pepper'}), items_add=frozenset({'ground beef'}),
confidence=0.3234501347708895, lift=3.2919938411349285)]),
 RelationRecord(items=frozenset({'tomato sauce', 'ground beef'}), support=0.005332622317024397,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'tomato sauce'}), items_add=frozenset({'ground beef'}),
confidence=0.3773584905660377, lift=3.840659481324083)]),
 RelationRecord(items=frozenset({'olive oil', 'light cream'}), support=0.003199573390214638,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'light cream'}), items_add=frozenset({'olive oil'}),
confidence=0.2051282051282051 5, lift=3.1147098515519573)]),
 RelationRecord(items=frozenset({'whole wheat pasta', 'olive oil'}), support=0.007998933475536596,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'whole wheat pasta'}), items_add=frozenset({'olive oil'}),
confidence=0.2714932126696833, lift=4.122410097642296)]),
 RelationRecord(items=frozenset({'pasta', 'shrimp'}), support=0.005065991201173177,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'pasta'}), items_add=frozenset({'shrimp'}), confidence=0.3220338983050847,
lift=4.506672147735896)]),
 RelationRecord(items=frozenset({'milk', 'avocado', 'spaghetti'}), support=0.003332888948140248,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'avocado', 'spaghetti'}), items_add=frozenset({'milk'}),
confidence=0.41666666666666663, lift=3.215449245541838)]),
```

FIG 4. ASSOCIATION

Conclusion:

E.g., Consider the first rule.

It says that the person who buys ‘Light cream’ are also likely to buy the ‘chicken’ . Using this observation we can place these two items either closer to each other so that customer will find it easily or the best strategy is to keep them at a certain distance so that while reaching the either of the product , the customer will have to walk from one place in the store to the other place & while walking he may purchase any other product. This way you can make a customer to buy more products than he was actually intended to buy & improve the stores profit.

References:
1-udemy.com
2-geeksforgeeks.com
3- Hands on Machine Learning with Scikit-Learn and Tensorflow - Aurelien Greon
4- Kaggle.com