

Market Basket Analysis using Python

May 2018

Rakesh Auplish

Introduction

- This presentation is about a project for implementation of Market Basket Analysis using Python
- The contents of this presentation are
 - What is Market Basket Analysis?
 - What data is used?
 - How it is implemented in Python?

What is Market Basket Analysis (MBA)



Data Mining technique to understand customer buying patterns and the associations between item they buy.

Apply **association analysis** to identify associations and infer marketing business rules like; product placement, catalog design and product recommendation

Market Basket Analysis schematic



Frequent Item sets in Python

Python Library **mlxtend** provides useful tools for deriving association rules from data mining

- Apriori algorithm is very popular for getting frequent itemsets.
- Mlxtend library **mlxtend.frequent_patterns** **apriori** generates frequent itemsets from datasets stored in dataframes

Association Rules in Python

- Mlxtend library
`mlxtend.frequent_patterns`
`association_rules`
generates association rules from
frequent itemsets
- The output from the library is shown
on the next slide

Association Rules in Python

Association Rules are displayed with their metrics as shown in the following table.

antecedants	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
(SET/6 RED SPOTTY PAPER CUPS)	(SET/6 RED SPOTTY PAPER PLATES)	0.05892	0.05949	0.05269	0.89423	15.0316	0.04919	8.89209
(SET/6 RED SPOTTY PAPER PLATES)	(SET/6 RED SPOTTY PAPER CUPS)	0.05949	0.05892	0.05269	0.88571	15.0316	0.04919	8.23442

Association Rules explained

Some terms used in Association Analysis are important to understand. Brief descriptions are provided here. For details visit

https://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/

- An association between Shirt and Trouser is written as {Shirt} -> {Trouser} and is termed as a *rule*
- {Shirt} is termed as *Antecedent*
- {Trouser} is termed as *Consequent*
- Both can have multiple items eg: {Shirt, Tie} -> {Trouser, Belt}

Association Rules explained

- *Support* metric is defined for item sets, not association rules. It is the relative frequency this rule occurs in the item set
 - {Shirt} -> {Trouser} occurs 270 times in 500 txns
 - Support = $270/500 = 0.54$ or 54%
- *Confidence* is a measure of reliability of the rule. It is 1 (maximal) for a rule $A \rightarrow C$ if the consequent and antecedent always occur together.
 - {Shirt} -> {Trouser} occurs 270 times
 - {Shirt} occurs 360 times
 - Confidence = $270/360 = 0.75$ or 75%
- *Lift* is confidence over probability of the outcome. If A and C are independent, the Lift score will be exactly 1.
 - Confidence = 0.75
 - {Trouser} occurs 290 times in 500 txns = 0.58
 - Lift = $0.75 / 0.58 = 1.29$

Association Rules explained

- *Leverage* is the reverse of lift. A leverage value of 0 indicates independence.
- *Conviction*. A high conviction value means that the consequent is highly depending on the antecedent. Similar to lift, if items are independent, the conviction is 1

Python code for analysis

Process for implementing Frequent Item sets and Association Rules in Python is as follows:

- Import the libraries

```
from mlxtend.preprocessing import TransactionEncoder  
from mlxtend.frequent_patterns import apriori  
from mlxtend.frequent_patterns import association_rules
```
- Binarise data using TransactionEncoder
- Generate frequent itemsets using apriori
- Generate association rules using association_rules
- See accompanying Python notebook for code examples.