ASSOCIATION RULES
MARKET BASKET ANALYSIS
USING PYTHON

DATA SCIENCE INSTITUTE

Contents

Introduction to Market Basket Analysis

- The most widely used area of application for association rules is **Market Basket Analysis**

> Market Basket Analysis (Association Analysis) is a **mathematical modeling technique based upon the theory that if you buy a certain group of items, you are likely to buy another group of items**

- It is used to **analyze the customer purchasing behavior and helps in increasing the sales and maintain inventory** by focusing on the point of sale transaction data

Market Basket Analysis – Uses

Product Building	• Develop combo offers based on products bought together
Optimisation	• Organise and place associated products/categories nearby inside a store
Advertising and Marketing	• Determine the layout of the catalog of an ecommerce site
Inventory Management	• Control inventory based on product demands and what products sell together

Definitions and Terminology

Term	Definition
Transactions	A set of items (Item set)
Support	Ratio of **number of times two or more items occur together** to **the total number of transactions** Support can be thought of as P(A and B)
Confidence	Conditional probability that **a randomly selected transaction will include Item B given Item A** **P(B|A)** (written as A => B)
Lift	Ratio of the **probability of Items A and B occurring together (Joint probability)** to **the product of P(A) and P(B)**

Rule Evaluation – Support

Transaction No.	**Item 1**	**Item 2**	**Item 3**	**…**
100	**Beer**	**Diaper**	Chocolate	
101	Milk	Chocolate	Shampoo	
102	Beer	Wine	Vodka	
103	**Beer**	Cheese	**Diaper**	
104	Ice Cream	**Diaper**	**Beer**	

A B

Support of {Diaper, Beer}

$$\text{Support} = \frac{\text{No.of transactions containing both A and B}}{\text{Total no.of transactions}} = \frac{3}{5} = 60\%$$

Support of {Diaper, Beer} is 3/5

Rule Evaluation – Confidence

Transaction No.	Item 1	Item 2	Item 3	…
100	**Beer**	**Diaper**	Chocolate	
101	Milk	Chocolate	Shampoo	
102	**Beer**	Wine	Vodka	
103	**Beer**	Cheese	**Diaper**	
104	Ice Cream	**Diaper**	**Beer**	

$$\text{Confidence for } \{A\} \Rightarrow \{B\} = \frac{\text{No. of transactions containing both A and B}}{\text{No. of transactions containing A}}$$

$$\textbf{Confidence for } \{\textbf{Diaper}\} \Rightarrow \{\textbf{Beer}\} \textbf{ is } 3/3$$

When Diaper is purchased, the likelihood of Beer purchase is 100%

$$\textbf{Confidence for } \{\textbf{Beer}\} \Rightarrow \{\textbf{Diaper}\} \textbf{ is } 3/4$$

When Beer is purchased, the likelihood of Diaper purchase is 75%

$\{$Diaper$\} \Rightarrow \{$Beer$\}$ is a more important rule according to Confidence

Rule Evaluation – Lift

Transaction No.	Item 1	Item 2	Item 3	Item 4
100	Beer	Diaper	**Chocolate**	
101	**Milk**	**Chocolate**	Shampoo	
102	Beer	**Milk**	Vodka	**Chocolate**
103	Beer	**Milk**	Diaper	**Chocolate**
104	**Milk**	Diaper	Beer	

$$\text{Consider } \overset{\textbf{A}}{\{\text{Chocolate}\}} \Rightarrow \overset{\textbf{B}}{\{\text{Milk}\}}$$

$$\text{Lift} = \frac{P(A \cap B)}{P(A)P(B)} = \frac{3/5}{\left(4/5\right)\left(4/5\right)} = 0.9375$$

Lift < 1 indicates Chocolate is decreasing the chance of Milk purchase

Case Study – Online Retail Data

Background

- A typical retail transactional data from a UK retailer from 2010-11

Objective

- To mine association rules and information about item sets

Available Information

- **Total number of transactions is 541910**
- **Items are aggregated to 392 categories**
- **Data is collected for 1 year (365 days)**

Data Snapshot

ONLINE RETAIL

Variables

Observations

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	2.75	17850	United Kingdom

Column	Description	Type	Measurement	Possible Values
InvoiceNo	Invoice Number	Numeric	-	-
StockCode	Stock Code	Categorical	-	-
Description	Product Description	Character	WHITE HANGING HEART T-LIGHT HOLDER etc	-
Quantity	Quantity	Continuous	-	Positive and Negative value
InvoiceDate	Date of Invoice	Date	dd-mm-yyyy hh:mm	01/12/2010 8:26 to 09/12/2011 12:50
UnitPrice	Price per unit of product	Continuous	-	Positive and Negative value
CustomerID	Customer ID	Continuous	-	-

Market Basket Analysis in Python

```
#Market Basket Analysis Using Apriori Recommendation
```

```
pip install mlxtend
```

```
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
df = pd.read_excel('Online Retail.xlsx')
df.head()
```

- We will be using library “**mlxtend**” for performing Market Basket Analysis in Python.
- Library **“mlxtend”** is used for extracting frequent itemsets with applications in association rule learning

* Data Source : Dr Daqing Chen, Director: Public Analytics group. chend '@' lsbu.ac.uk, School of Engineering, London South Bank University, London SE1 0AA, UK, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science

Market Basket Analysis in Python

Output:

Index	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 8.26	2.55	17850	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 8.26	3.39	17850	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 8.26	2.75	17850	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 8.26	3.39	17850	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 8.26	3.39	17850	United Kingdom

Visualise Item Frequency

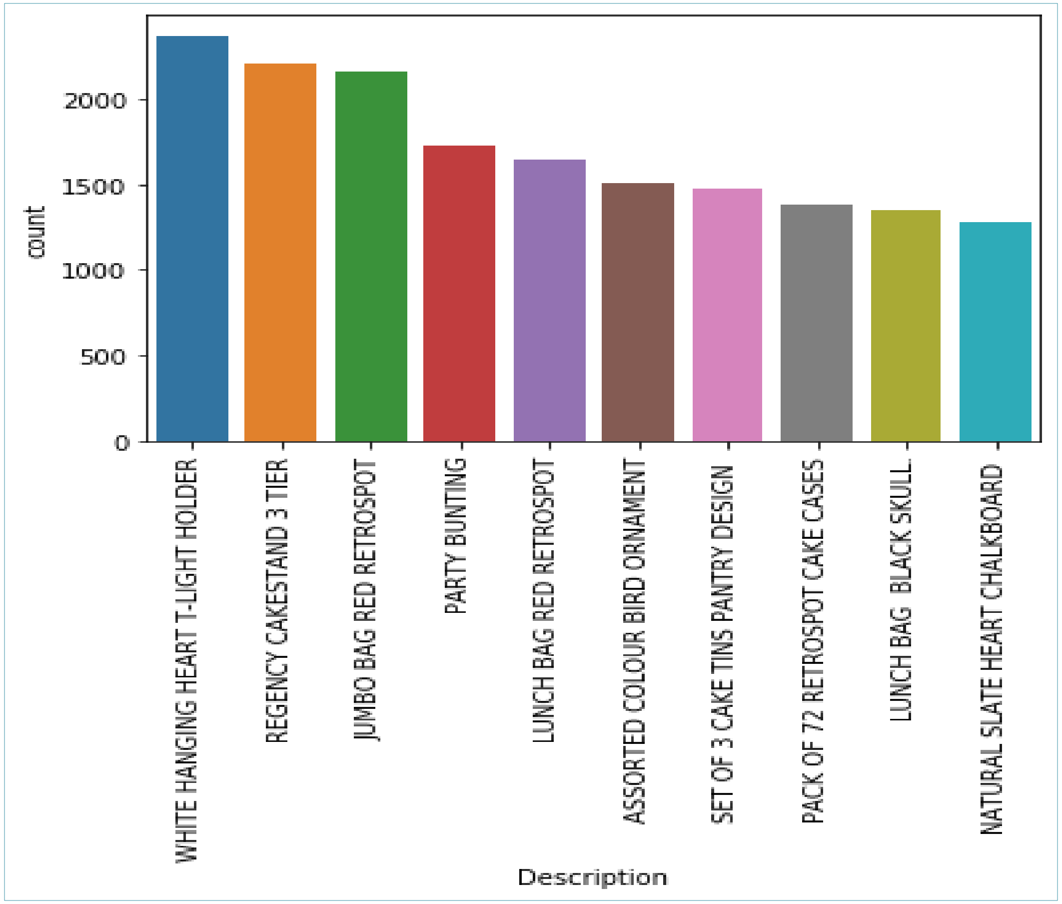
```
#Item Frequency Plot
import seaborn as sns
import matplotlib.pyplot as plt

sns.countplot(x = 'Description', data = df, order =
df['Description'].value_counts().iloc[:10].index)
plt.xticks(rotation=90)
```

- **sns.countplot()** calculates item frequency and returns a barplot.
- **order =** used to plot the categorical levels in specific order

Item Frequency Plot

```
# Output
```



Interpretation:
The plot shows items by frequency
in a descending order.

Basic Data Cleanup

```
# Data Cleaning and Consolidation
df['Description'] = df['Description'].str.strip()
df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
df['InvoiceNo'] = df['InvoiceNo'].astype('str')
df = df[~df['InvoiceNo'].str.contains('C')]
```

- **strip()** returns a copy of the string with both leading and trailing characters removed .
- **dropna()** removes all the missing values and a new object is returned which does not have any NaN values present in it.
- **contains()** function is used to test if pattern or regex is contained within a string of a Series or Index. Here it is used to remove rows where ‘InvoiceNo. Contains ‘C’

```
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))
basket.head()
```

- After the cleanup, consolidation of the items into 1 transaction per row with each product is done.

Basic Data Cleanup

```
# Output:
```

Description	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 EGG HOUSE PAINTED WOOD	12 MESSAGE CARDS WITH ENVELOPES	12 PENCIL SMALL TUBE WOODLAND	12 PENCILS SMALL TUBE RED RETROSPOT	12 PENCILS SMALL TUBE SKULL	12 PENCILS TALL TUBE POSY
InvoiceNo								
536370	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
536852	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
536974	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
537065	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
537463	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Consolidation of items

```
# Data consolidation
def encode_units(x):
    if x <= 0:
        return 0
    if x >= 1:
        return 1

basket_sets = basket.applymap(encode_units)
basket_sets.drop('POSTAGE', inplace=True, axis=1)
```

- **applymap()** method applies a function that accepts and returns a scalar to every element of a DataFrame.
- This way, we generated a data frame that shows us whether a particular items is bought or not.

```
frequent_itemsets = apriori(basket_sets, min_support=0.07, use_colnames=True)
```

- Once data is structured properly, frequent item sets that have a support of at least 7% is generated.

Get and Display the Rules

```
#Get the Rules
```

```
rules = association_rules(frequent_itemsets, metric="lift",
min_threshold=1)
```

- **association_rules()** generate the rules with their corresponding support, confidence and lift.

```
#Show Top 5 Rules
```

```
rules.head()
```

```
# Output:
```

Index	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	frozenset({'ALARM CLOCK BAKELIKE PINK'})	frozenset({'ALARM CLOCK BAKELIKE GREEN'})	0.102040816	0.096938776	0.073979592	0.725	7.478947368	0.06408788	3.283858998
1	frozenset({'ALARM CLOCK BAKELIKE GREEN'})	frozenset({'ALARM CLOCK BAKELIKE PINK'})	0.096938776	0.102040816	0.073979592	0.763157895	7.478947368	0.06408788	3.79138322
2	frozenset({'ALARM CLOCK BAKELIKE RED'})	frozenset({'ALARM CLOCK BAKELIKE GREEN'})	0.094387755	0.096938776	0.079081633	0.837837838	8.642958748	0.069931799	5.568877551
3	frozenset({'ALARM CLOCK BAKELIKE GREEN'})	frozenset({'ALARM CLOCK BAKELIKE RED'})	0.096938776	0.094387755	0.079081633	0.815789474	8.642958748	0.069931799	4.916180758
4	frozenset({'POSTAGE'})	frozenset({'ALARM CLOCK BAKELIKE GREEN'})	0.765306122	0.096938776	0.084183673	0.11	1.134736842	0.009995835	1.014675533

Manage How the Rules are Displayed

#Sort the Rules

```
rules[ (rules['lift'] >= 6) &
       (rules['confidence'] >= 0.8) ]
```

- Dataframe can be filtered using standard pandas code. In this case, rules with high lift (>6) and high confidence (>8) are displayed.

Output:

Index	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
2	frozenset({'ALARM CLOCK BAKELIKE GREEN'})	frozenset({'ALARM CLOCK BAKELIKE RED'})	0.0969388	0.0943878	0.0790816	0.8157895	8.6429587	0.0699318	4.9161808
3	frozenset({'ALARM CLOCK BAKELIKE RED'})	frozenset({'ALARM CLOCK BAKELIKE GREEN'})	0.0943878	0.0969388	0.0790816	0.8378378	8.6429587	0.0699318	5.5688776
17	frozenset({'SET/6 RED SPOTTY PAPER PLATES'})	frozenset({'SET/20 RED RETROSPOT PAPER NAPKINS'})	0.127551	0.1326531	0.1020408	0.8	6.0307692	0.0851208	4.3367347

Interpretation:

- Green and red alarm clocks are purchased together and the red paper cups, napkins and plates are purchased together in a manner that is higher than the overall probability

Manage How the Rules are Displayed

```
basket['ALARM CLOCK BAKELIKE GREEN'].sum()
340.0
basket['ALARM CLOCK BAKELIKE RED'].sum()
316.0
```

- In order to check how much opportunity is there to use the popularity of one product to drive sales of another, their sum is calculated.
- For example, it can be seen that 340 Green Alarm clocks are sold but only 316 Red Alarm clocks are sold, hence maybe selling of Red Alarm Clock can be increased through recommendations

Combinations by country

```
basket2 = (df[df['Country'] =="Germany"]
          .groupby(['InvoiceNo', 'Description'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))
basket_sets2 = basket2.applymap(encode_units)
basket_sets2.drop('POSTAGE', inplace=True, axis=1)
frequent_itemsets2 = apriori(basket_sets2, min_support=0.05,
use_colnames=True)
rules2 = association_rules(frequent_itemsets2, metric="lift",
min_threshold=1)

rules2[ (rules2['lift'] >= 4) &
        (rules2['confidence'] >= 0.5)]
```

- ❑ It is interesting to see how the combinations vary by country of purchase.
- ❑ Here, some popular combinations in Germany are displayed

Combinations by country

Output:

Index	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
1	frozenset({'PLASTERS IN TIN CIRCUS PARADE'})	frozenset({'PLASTERS IN TIN WOODLAND ANIMALS'})	0.1159737	0.1378556	0.067833698	0.58490566	4.242887092	0.051846071	2.076984285
7	frozenset({'PLASTERS IN TIN SPACEBOY'})	frozenset({'PLASTERS IN TIN WOODLAND ANIMALS'})	0.107221	0.1378556	0.061269147	0.571428571	4.145124717	0.046488133	2.011670314
11	frozenset({'RED RETROSPOT CHARLOTTE BAG'})	frozenset({'WOODLAND CHARLOTTE BAG'})	0.0700219	0.1269147	0.059080963	0.84375	6.648168103	0.050194159	5.587746171

Interpretation :

- It can be inferred that Germans like Plasters in Tin Spaceboy and Woodland Animals.

Quick Recap

In this session, we learnt **Market Basket Analysis:**

Market Basket Analysis	• Mathematical modeling technique based upon the theory that if you buy a certain group of items, you are likely to buy another group of items • Transactions, Support, Confidence and Lift are the key concepts used in this analysis • The analysis is performed by creating and studying rules based on different itemsets
Market Basket Analysis in Python	• Library `mlxtend` is used for undertaking MBA in Python • `sns.countplot()` plots frequency • `apriori()` builds frequent items • `association_rules()` builds the rules

THANK YOU!