Link Prediction in Sparse Matrix Using Data Mining

Project report submission in partial fulfilment of the requirements for the degree of
BACHELOR OF TECHNOLOGY
In
Department of Computer Science and Engineering
By

Ashish Jha	Roll No- 12616001048
Rakesh Kumar	Roll No- 12616001124
Jagrit Drolia	Roll No- 12616001077
Pratyush Kumar Singh	Roll No- 12616007082

Under the guidance of
Prof. Lopamudra Dey,
Assistant professor,
Department of Computer Science and Engineering,
Heritage Institute of Technology, Kolkata

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety.

I have duly acknowledged all the sources of information which have been used in this thesis.

Ashish Jha	Roll No- 12616001048
Rakesh Kumar	Roll No- 12616001124
Jagrit Drolia	Roll No- 12616001077
Pratyush Kumar Singh	Roll No- 12616007082

ACKNOWLEDGEMENT

I would take this opportunity to thank **Prof. Pranay Chaudhuri**, Director Heritage Institute of Technology for allowing us to form a group of four people and for providing us with all the necessary facilities to make our project work and of worth.

We would be thankful to **Prof. Lopamudra Dey**, our project coordinator as well as our project mentor for constantly supporting and guiding us by giving her invaluable insights. Her guidance encourages us to achieve our goal and impetus to excel.

We thank our teachers, Faculty members and laboratory assistants at the Heritage Institute of Technology for playing a pivotal and decisive role during the development of our project. Last but not the least we thank all friends for their cooperation and encouragement that they bestowed on us.

Ashish Jha

Rakesh Kumar

Jagrit Drolia

Pratyush Kumar Singh

HERITAGE INSTITUTE OF TECHNOLOGY, KOLKATA

WEST BENGAL UNIVERSITY OF TECHNOLOGY

BONAFIDE CERTIFICATE

Certified that this report "........**Link Prediction in Sparse Matrix using Data Mining** ....." is the bonafide work of"........ **Ashish Jha**, **Rakesh Kumar**, **Jagrit Drolia** and **Pratyush Kumar Singh** ...." who carried out the project under my supervision.

SIGNATURE

Prof. Subhasis Majumdar

HEAD OF THE DEPARTMENT

Computer Science and Engineering

Heritage Institute of Technology

SIGNATURE

Prof. Lopamudra Dey

PROJECT GUIDE, Asst Professor

Computer Science and Engineering

Heritage Institute of Technology

SIGNATURE

EXAMINER

Content

Abstract

We propose to solve the link prediction problem in sparse matrix using clustering with association rule mining.The model learns latent features from the sparse matrix, and makes better predictions.

To present the effect of clustering the data onto the association rules.Hence, we have compared the results of two different approaches: Finding association rules without consumer segmentation, and with consumer segmentation. The data analysis framework is applied to the data of Online Retail. By extracting the most important information from Online Retail data, we claim that this framework offers the right product/advertisement to the right consumer.

Predicting from a sparse matrix using algorithms like ARM are quite expensive and it takes a lot of time and space but using Clustering algorithms like K-means and DBSCAN with Apriori we can predict in linear time complexity.

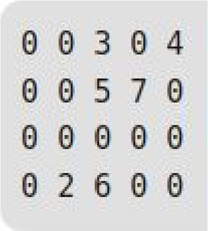
We compare the time complexity of Apriori, K-means with Apriori and DBSCAN with Apriori and their precision, And we got very positive output.

INTRODUCTION

The main goal of data mining is to define a process for discovering significant patterns or anomalies in a large volume of data. It has been applied to decision support problems in diverse areas such as medical diagnosis, targeted marketing, bioinformatics, sociology, networking, and information security, making data mining one of the most widely studied topics in intelligent systems. Data mining incorporates theory and practical developments from many older research areas such as databases, machine learning, artificial intelligence, distributed computing, information retrieval, and statistics, and lends an integrative perspective to these research areas. Due to the breadth of both applications and foundational theory in data mining research, it is often divided along methodological lines, into tasks such as classification, clustering, association, decision support, and visualization. Association rule mining, Apriori, K-Means are subtopics which have been explored by many research groups. It addresses the problem of discovering relationships between instances that originate from dependence or interaction

Sparse Matrix:

In numerical analysis and scientific computing, a sparse matrix or sparse array is a matrix in which most of the elements are zero. By contrast, if most of the elements are nonzero, then the matrix is considered dense. The number of zero-valued elements divided by the total number of elements (e.g., m × n for an m × n matrix) is called the sparsity of the matrix (which is equal to 1 minus the density of the matrix). Using those definitions, a matrix will be sparse when its sparsity is greater than 0.5.



```
0 0 3 0 4
0 0 5 7 0
0 0 0 0 0
0 2 6 0 0
```

Link Prediction :

Currently with the rapid development, online social networks have been a part of people’s life and it all can be represented in the form of a sparse matrix. A lot of sociology, biology, and information systems can use the sparse to describe, in which zero represents an individual is inactive and non-zero represents the individual is active. Link prediction not only can be used in the field of social network but can also be applied in other fields. As in bioinformatics, link prediction can be used to discover interactions between proteins, in the field of electronic commerce, link prediction can be used to create the recommendation system and in the security field, link prediction can help to find the hidden terrorist criminal gangs. Link prediction is closely related to many areas.

Link prediction in social network :

A social network is a collection of associations between individuals (e.g., people) or organizations that can be graphically represented. Links in this graph are based on one or more specific types of interdependence, such as values, visions, ideas, financial exchange, friendship, kinship, dislike, conflict or trade.

The term was first coined by 24 Professor J. A. Barnes in the 1950s (in: Class and Committees in a Norwegian Island Parish, “Human Relations”). The information that a social network provides about each individual can be used to build prediction models for a recommended link or missing link for example. Social network services such as MySpace and Facebook allow users to create a profile which contains many aspects related to the user (e.g. lists of interests, communities, schools, and links to friends). Some services, such as Google’s OrKut, are community-centric; others, such as the video blogging service YouTube and the photo service Flickr, are related to social media. In other kinds of services such as Six Apart’s LiveJournal and Vox, they are organized around text-and image weblogs. Some studies such as Hsu, et al. (2007) use a friend’s network of LiveJournal to predict friendship based on graph features.

Other studies such as those by LibenNowell and Kleinberg (2003) and Popescul and Ungar (2003), define certain linkage measures to estimate the existence probability of a potential future link. In order to use the co-occurrent property through association rules, Schmitz, et al. (2006) propose a method of using association rules in Folksonomies 0F 1 as a recommended system (such as tags, users, or resources). In my research, I use association measures (based on users co-occurrence) of some users’ properties as link prediction features (Aljandal, et al. (2008)

APPLICATION DOMAINS

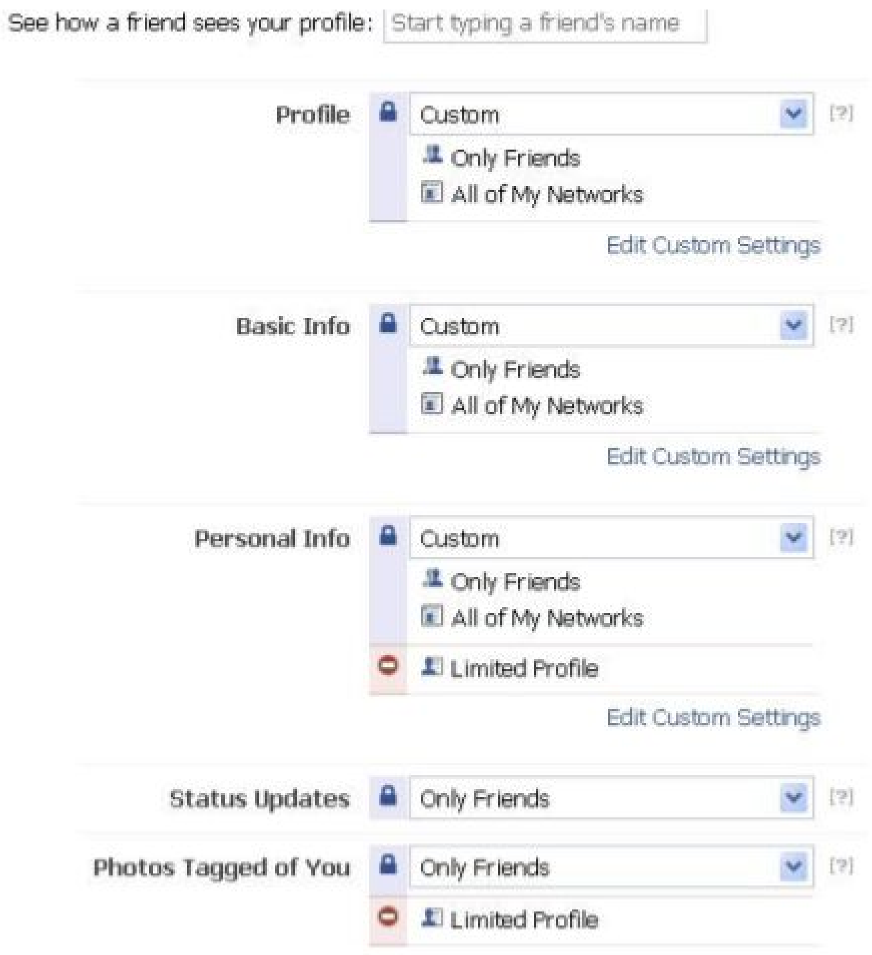
Social Networks :

Most social networking services include friend-listing mechanisms that allow users to link to others, indicating friends and associates. Friendship networks do not necessarily entail that these users know one another, but are means of expressing and controlling trust, particularly accessing private content.

In blogging services such as SUP’s LiveJournal or Xanga, this content centers on text but comprises several media, including: interactive quizzes, voice posts, embedded images, and video hosted by other services such as YouTube.

In personal photograph-centric social networks such as News Corporation’s MySpace, Facebook, Google’s Orkut, and Yahoo’s Flickr, links can be annotated (“How do you know this person?”) and friends can be prioritized (“top friends” lists) or granted privileges as shown in Figure 1 below.

Some vertical social networks such as LinkedIn, Classmates.com, and MyFamily.com specialize in certain types of links, such as those between colleagues, previous employers and employees, classmates, and relatives. As in vertical search 45 and vertical portal applications, this specialization determines many aspects of the data model, data integration, and user knowledge elicitation tasks.



See how a friend sees your profile: Start typing a friend's name

Profile: Custom [?]
Only Friends
All of My Networks
Edit Custom Settings

Basic Info: Custom [?]
Only Friends
All of My Networks
Edit Custom Settings

Personal Info: Custom [?]
Only Friends
All of My Networks
Limited Profile
Edit Custom Settings

Status Updates: Only Friends [?]

Photos Tagged of You: Only Friends [?]
Limited Profile

Figure 1 Facebook's access control lists for user profile components. © 2008 Facebook, Inc.

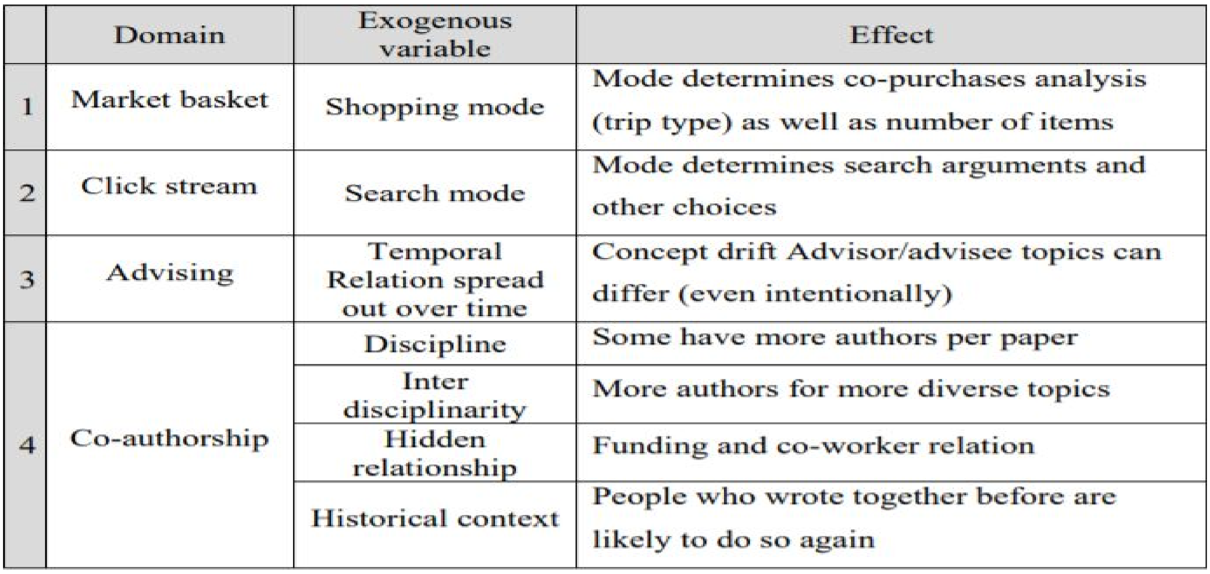
For example, LinkedIn's friend invitation process requires users to specify their relationship to the invited friend, an optional or post-hoc step in many other social networks. Friendship links can be undirected, as in Facebook and LinkedIn (requiring reciprocation, also known as confirmation, to confer access privileges) or directed, as in LiveJournal (not necessarily requiring reciprocation). In my research I use LiveJournal dataset where the links present a friendship relation between users. For the itemsets

I use two user’s properties: users’ interests and communities’ membership.

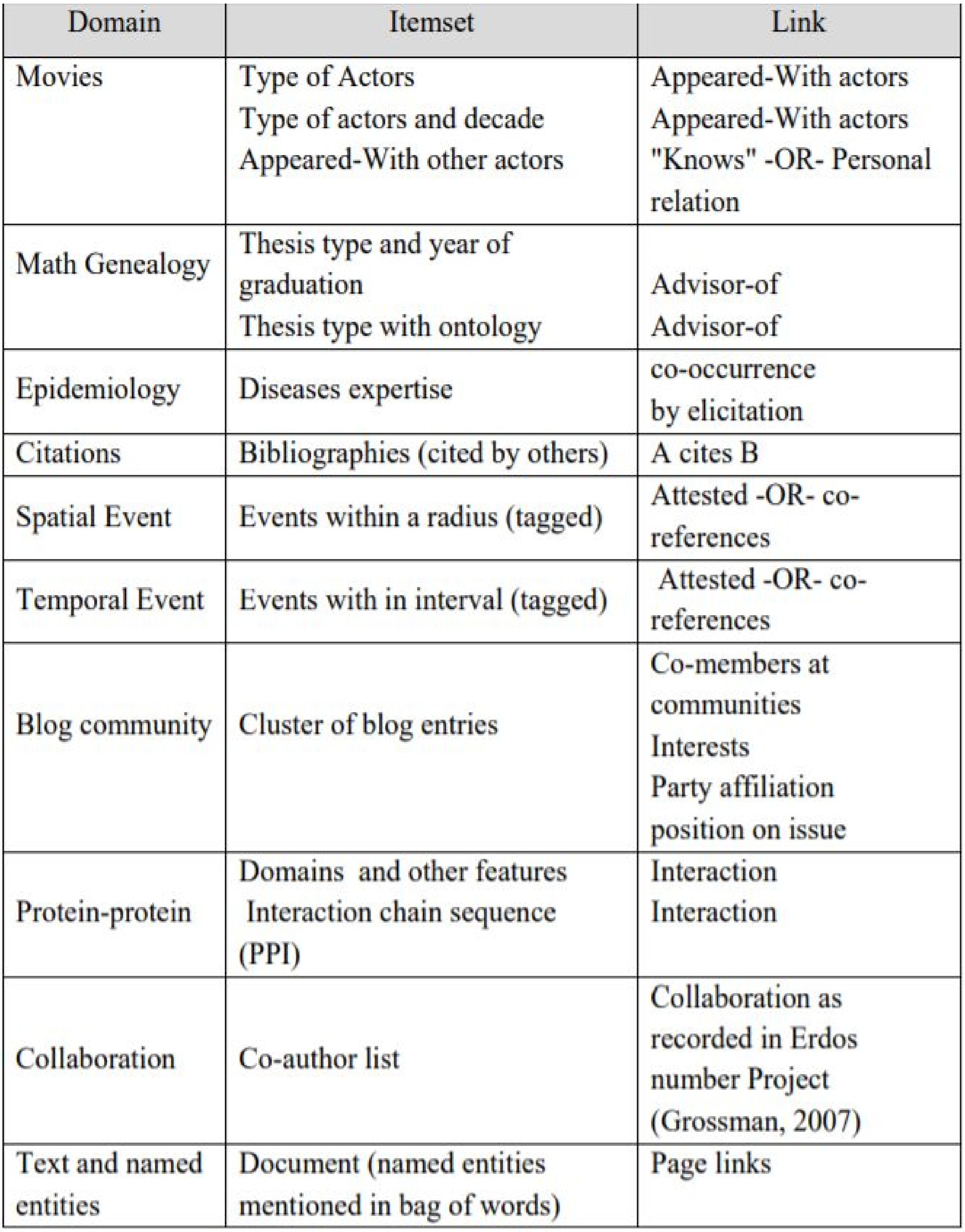
Protein-Protein Interaction :

There are different kinds of information related to protein that can be taken into account in studying the interaction between proteins. However, the information about protein-protein interactions are important for many biological functions and diseases. For example, the number of features can be collected from interactions between yeasts and features that characterize each protein involved in the interaction. In a 46 research study by Oyama, et al. (2000) produced more than two thousand features from six different types of protein features. In addition, a protein interaction network can be a source of information to build a prediction module for unknown interactions. For example, the paper by Schwikowski (2000) and others related to the PPI Networks in Rice Blast Fungus (He, Zhang, Chen, Zhang, & Peng, 2008) are useful for investigating the cellular functions of genes.

Other domains surveyed :



	Domain	Exogenous variable	Effect
1	Market basket	Shopping mode	Mode determines co-purchases analysis (trip type) as well as number of items
2	Click stream	Search mode	Mode determines search arguments and other choices
3	Advising	Temporal Relation spread out over time	Concept drift Advisor/advisee topics can differ (even intentionally)
4	Co-authorship	Discipline	Some have more authors per paper
		Inter disciplinarity	More authors for more diverse topics
		Hidden relationship	Funding and co-worker relation
		Historical context	People who wrote together before are likely to do so again



Domain	Itemset	Link
Movies	Type of Actors Type of actors and decade Appeared-With other actors	Appeared-With actors Appeared-With actors "Knows" -OR- Personal relation
Math Genealogy	Thesis type and year of graduation Thesis type with ontology	Advisor-of Advisor-of
Epidemiology	Diseases expertise	co-occurrence by elicitation
Citations	Bibliographies (cited by others)	A cites B
Spatial Event	Events within a radius (tagged)	Attested -OR- co-references
Temporal Event	Events with in interval (tagged)	Attested -OR- co-references
Blog community	Cluster of blog entries	Co-members at communities Interests Party affiliation position on issue
Protein-protein	Domains and other features Interaction chain sequence (PPI)	Interaction Interaction
Collaboration	Co-author list	Collaboration as recorded in Erdos number Project (Grossman, 2007)
Text and named entities	Document (named entities mentioned in bag of words)	Page links

Proposed Work

There are many data mining learning algorithms which are used for link prediction in sparse matrices. The main objective of our project is to compare the complexity of algorithms like apriori with the combination of K-Means and Apriori.In our project we have used K++ Means.It is possible to use different types of algorithms to extract the most important information from a database. Ten most popular data mining algorithms identified by the IEEE International Conference on Data Mining (ICDM) are C4.5, K-Means, SVM, Apriori, EM, PageRank, AdaBoost, kNN, Naive Bayes, and CART. These top ten algorithms are among the most influential data mining algorithms in the research community,but we have used here K Means because our data set is a sparse data set and K means is one the most suitable algorithm for the sparse dataset.The data analysis framework in this paper is based on Apriori data mining algorithm. Data on Online Retail are collected from Kaggle that is essential for this purpose. We first apply Apriori algorithm to this data and obtain related association rules.An association rule expresses an association between items or sets of items. We utilize two metrics, confidence and lift, for evaluating these association rules. As a second approach, we again apply the Apriori algorithm, but after clustering bought items. The segmentation of the items is generated using the K-means algorithm. The application is carried out on the Jupyter Notebook python3. We compare the final results of these two approaches in order to reveal the impact of consumer clustering on the results of the Apriori algorithm.

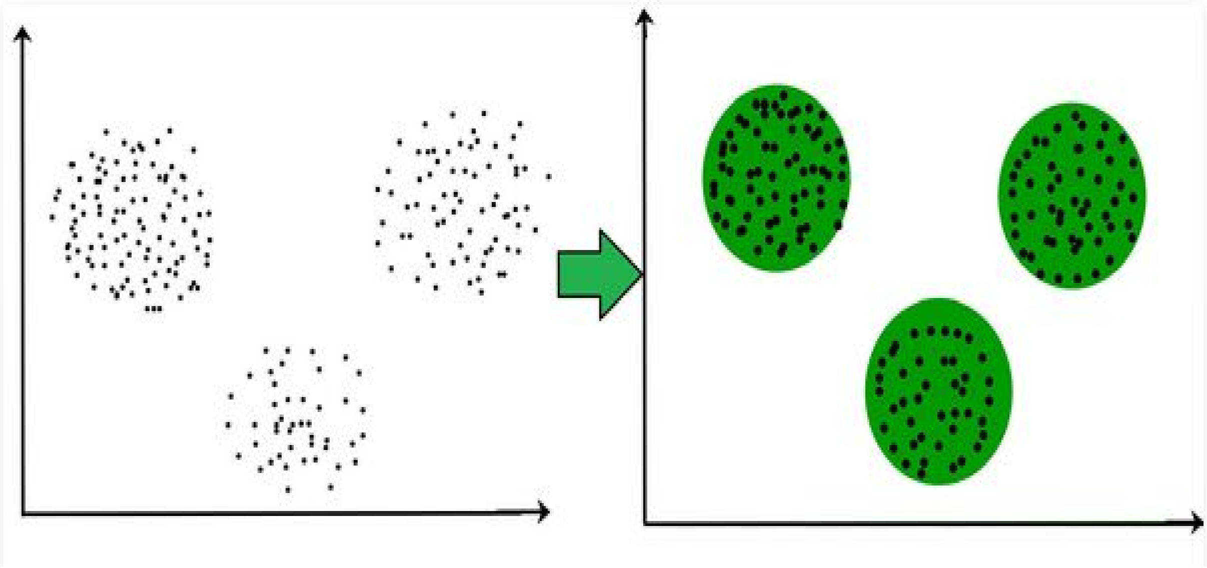
Literature Review

Data mining has a lot of potential and it is described as "a process of nontrivial extraction of implicit, previously unknown and potentially useful information from the data stored in a database".

Each of the data mining techniques that are used have one of the two objectives :

- classification/clustering and prediction.

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them and the prediction models are used to predict the continuous-valued functions.



Association Rule Mining(ARM)

Association rule mining is a method for discovering the relationships or correlation between items based on measures that are defined over observed items and proposed relationships. One of the most important aspects of association rule mining is ranking rules by their significance, according to some quantitative measure that expresses their interestingness with respect to a decision support or associative reasoning task.

[3] The concept of association rules was first introduced in a 1993 article (Agrawal, Imielinski, & Swami, 1993) in which the Apriori algorithm was also presented. Since then, association rule mining has become one of the most highly used and studied techniques in data mining.

The main principle of this technique involves discovering the efficient relationship and co-occurrence between items in the data. In other words, it discovers and measures quantitative evidence for relationships expressed in the database.

Association rules are expressed in an IF-THEN propositional rule-based format. A classic example of this method is market basket analysis. Consider a simple example: 2 “customers who buy product A often also buy product B”.[8] A decision maker such as a shopper or a marketer can access a large volume of historical data from which such rules have been extracted, to more confidently draw conclusions and make decisions that are well-supported by the data.

Formal definition:

Let L = {I1, I2, …, Im} be a set of m distinct attributes (items). Let D be a database, where each record (itemset) T has a unique identifier, and contains a set of items such that T ⊆ L. An association rule is an implication of the form X→Y, where both X,

$Y \subset L$, are sets of items called itemsets, and ($X \cap Y = \varphi$) where X and Y are two disjoint sets of items. Here, X is called the antecedent, and Y the consequent.[9] The rule can be described as when we find all items in X within a transaction it is likely the transaction also contains the items in Y (Agrawal, Imielinski, & Swami, 1993).

The first step in generating the rules is applying frequent itemset algorithms over all possible rules. The rules will then be selected based on thresholds and measures of significance and interestingness.

Measurement of Association Rules Generating association rules from a certain dataset will lead to a large number of rules if we do not specify a threshold for each specific measure. In this introduction, I present the two most fundamental association rule interestingness measures, support and confidence, which are the basis of the Apriori algorithm.

Support Support is a basic measure related to probability and set theory.[1] It is defined as the fraction of transactions in the database which contain all items in a specific rule (Agrawal, Imielinski, & Swami, 1993). This can be written as: Supp($X \rightarrow Y$) = Supp($X \cup Y$) = | *xxxx* | |*DD*| Where |xy| is the number transactions (itemset) which contain both X and Y – i.e., the probability of (x, y) – and |D| represents the total number of transactions (itemset) in the database. Minimum support thresholds are usually specified in generating the association rules which select only the most frequent items in the database.

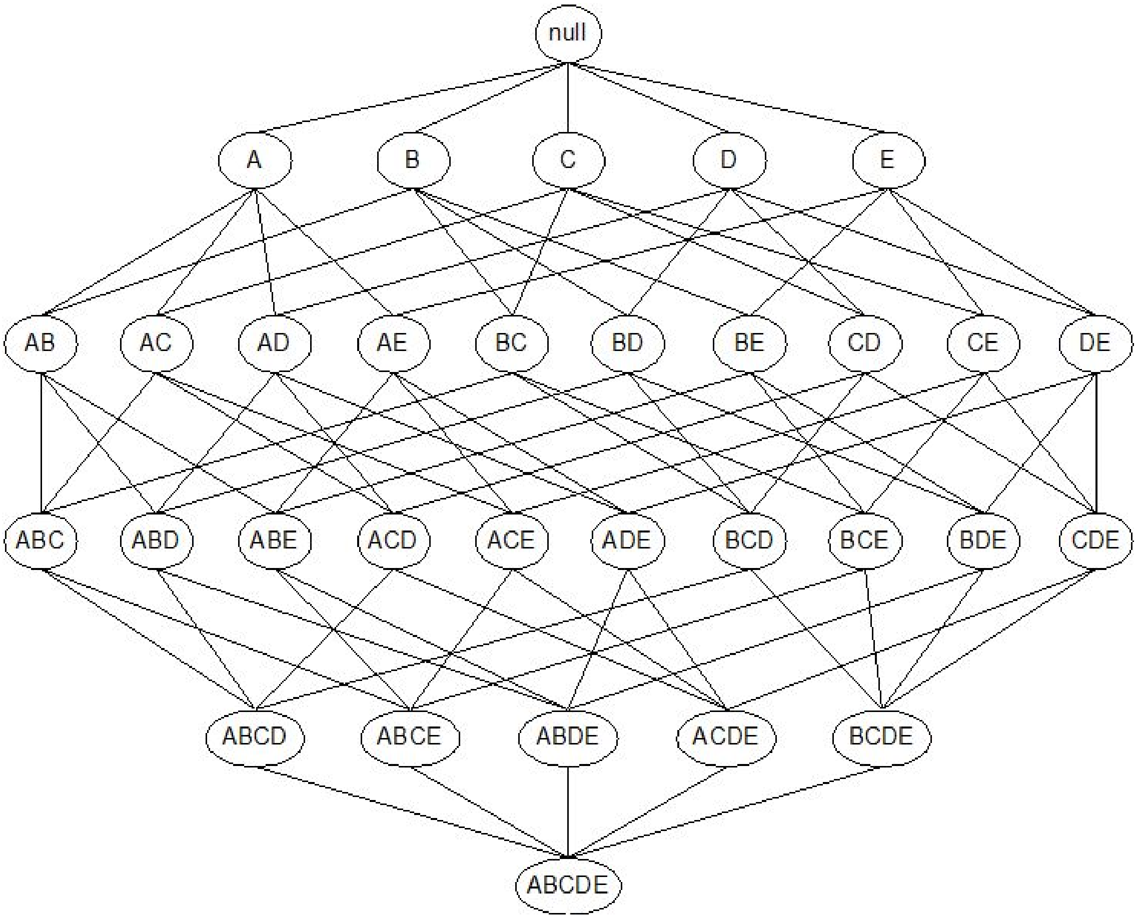
Confidence Another measure of the association rules is confidence. This is the strength of the implication of a rule and can be represented as a ratio between the transaction numbers, including X and Y and those including X, which can be written as: Conf($X \rightarrow Y$) = Supp ($X \rightarrow Y$) Supp (X) = | *xxxx* | | *xx*| Where |x| is the number of transactions (itemset) containing X. Example of

Association rules: market basket analysis, which is analyzing customer buying habits by finding associations between items that customers place in their “shopping baskets”

MARKET BASKET ANALYSIS :

Market Basket Analysis is one of the key techniques used by large retailers to uncover associations between items. It works by looking for combinations of items that occur together frequently in transactions. To put it another way, it allows retailers to identify relationships between the items that people buy.

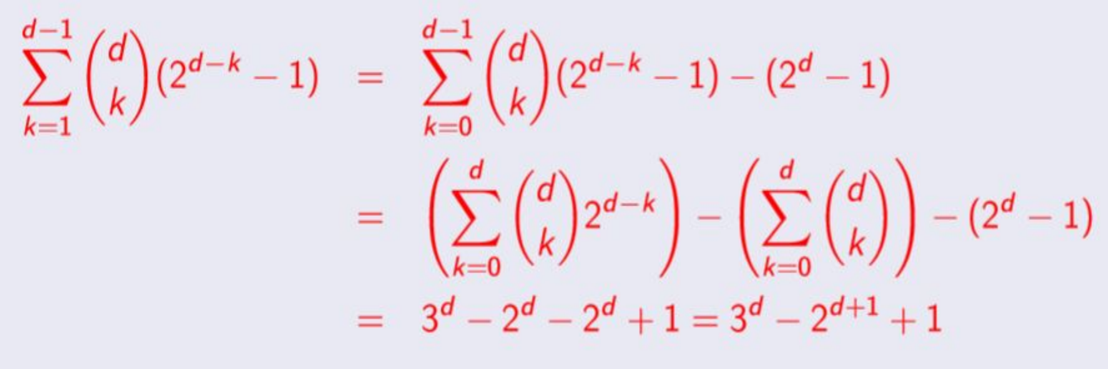
Frequent Itemset Generation



Given d items, there are **2^{d}** possible candidate itemsets

The number of distinct non-empty itemsets over I is **2^{d-1}**, while the number of distinct association rules is **$3^{d} - 2^{d+1} + 1$**.

As for the association rules, we count separately those whose LHS has k items, for 1 ≤ k < d. There are d**C**k possible itemsets of size k, and each of these, say X ,can form a rule with *2^{d-k}* −1 distinct non-empty itemsets, disjoint from X. Thus, the total number of rules is:



$$\begin{aligned}\sum_{k=1}^{d-1}\binom{d}{k}(2^{d-k}-1) &= \sum_{k=0}^{d-1}\binom{d}{k}(2^{d-k}-1)-(2^{d}-1)\\ &= \left(\sum_{k=0}^{d}\binom{d}{k}2^{d-k}\right)-\left(\sum_{k=0}^{d}\binom{d}{k}\right)-(2^{d}-1)\\ &= 3^{d}-2^{d}-2^{d}+1=3^{d}-2^{d+1}+1\end{aligned}$$

Role of association rules in link mining :

The main objective of the association rule is discovering the associations between instances which can be measured and filtered later on. Capturing a relation is similar to finding a link between instances. Association measures are descriptive statistics computed over rules of the form u → v .

This allows us to apply algorithms for association rule mining based on calculation of frequent itemsets (co-occurrence), which, by analogy with market basket analysis, denote sets of instances who share a specific property.[6] Some research focuses on such property, such as Ganiz, et al. (2006) and Shanfeng, et al. (2005), to improve link prediction.

Using association rule mining concepts and measures in link mining can be summarized as follow: if two instances X, Y are co-occurring in many cases (P(X, Y) is high enough to consider), we can predict that there is a link between X and Y and the significance of this link can be measured by using some association measures. Getoor et al. (2001) observed that there are often correlations between the attributes 0.51 0.515 0.52 0.525 0.53 0.535 0.54 1 10 19 28 37 46 55 64 73 82 91 100 109 118 AUC Parameter m Unnormalized Normalized 41 of entities and the relations in which they participate in.

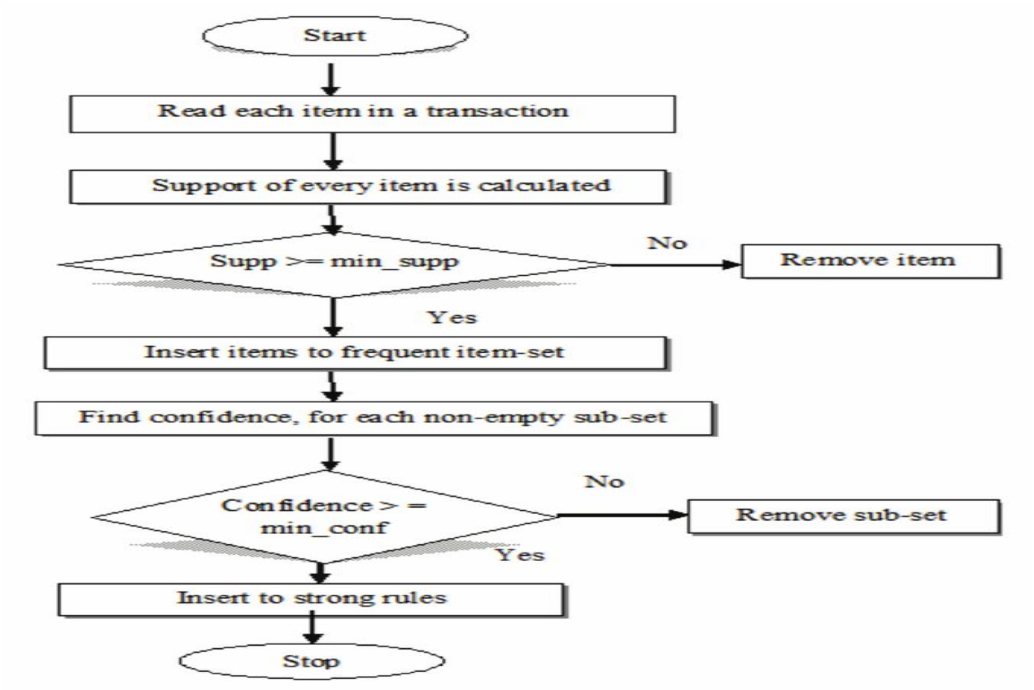
For example, in a social network, people who have the same hobbies are more likely to be friends. Therefore, we can use association rule measures as numerical features for building a link prediction module. In addition, each association rule measure captures one or more desiderata of a data mining system: novelty (surprisingness), validity (precision, recall, and accuracy), expected utility, and comprehensibility (semantic value).

APRIORI

The Apriori algorithm is probably the most well-known algorithm in the area of frequent item discovery (Agrawal, Imielinski, & Swami, 1993).[4] The algorithm takes advantage of the property that any subset of a frequent item set must be a frequent item set.

If we have (N+1)-item set then we use the (N)-item set (N is the number of items in the set) to discover it. Thus, the discovered frequent itemsets of the first pass are used to generate the candidate sets of the second pass. Once the candidate 1-item sets are found their supports are counted to discover the frequent 2-itemsets by scanning the database.

In the third pass, the frequent 2-item sets are used to generate candidate 3-item sets. Termination conditions, where there are no more new frequent item sets, is found in Figure.


Start
Read each item in a transaction
Support of every item is calculated
Supp >= min_supp
No
Remove item
Yes
Insert items to frequent item-set
Find confidence, for each non-empty sub-set
Confidence > = min_conf
No
Remove sub-set
Yes
Insert to strong rules
Stop

The algorithm contains two steps:

1. Join step: 14 The first step is to join all frequent items of size k-1 ((k-1)-item set) with themselves to generate candidate K-item sets. As a result the new list of k-item sets has been produced.

2. Prune step: This step comes from the Apriori property which states if an item set is not frequent, then all its supersets are absolutely not a frequent set.

Therefore we can prune all Candidate k- itemsets by checking whether all its (k-l)-item sets subsets are frequent or not. If we find any member of (k-1)-itemsets is not, we can prune its superset from a new list.

Method: apriori_gen() [(Agrawal & Srikant, 1994)]

Input: set of all large (k-1)-item sets L_{k-1}

Output: A superset of the set of all large k-item sets

```
// Join step
 Ii = Items i
        insert into $C_k$
        Select p.I1, p.I2, ……. , $p.I_{k-1,}$ $q .I_{k-1}$
        From $L_{k-1}$ is p, $L_{k-1}$ is q
        Where $p.I_1 = q.I_1$ and …… and $p.I_{k-2} = q.I_{k-2}$ and $p.I_{k-1} < q.I_{k-1}$.
        // Pruning step
```

```
    For all item sets $c \in C_k$ do
        For all (k-1)-subsets s of c do
            If $(s \notin L_{k-1})$ then
                delete c from $C_k$
```

------------------------Apriori Algorithm Pseudo code ----------------------------

The main disadvantage of the Apriori algorithm is running time, because the algorithm needs to scan the database for every processed level.[5] The performance of the algorithm will be unacceptable when the database size is large, however, there are many algorithms that have been proposed to solve this problem and improve the performance of the processing time of finding the frequent items.

“If an itemset is frequent, then all of its subsets must also be frequent.”

Apriori principle holds due to the following property of the support

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

measure

Support of an itemset never exceeds the support of its subsets

This is known as the anti-monotone property of support

$$Support\ (X,Y) = P(X,Y) = \frac{\#\{\text{Customers who bought } X \text{ and } Y\}}{\#\{\text{Customers}\}} \quad (1)$$

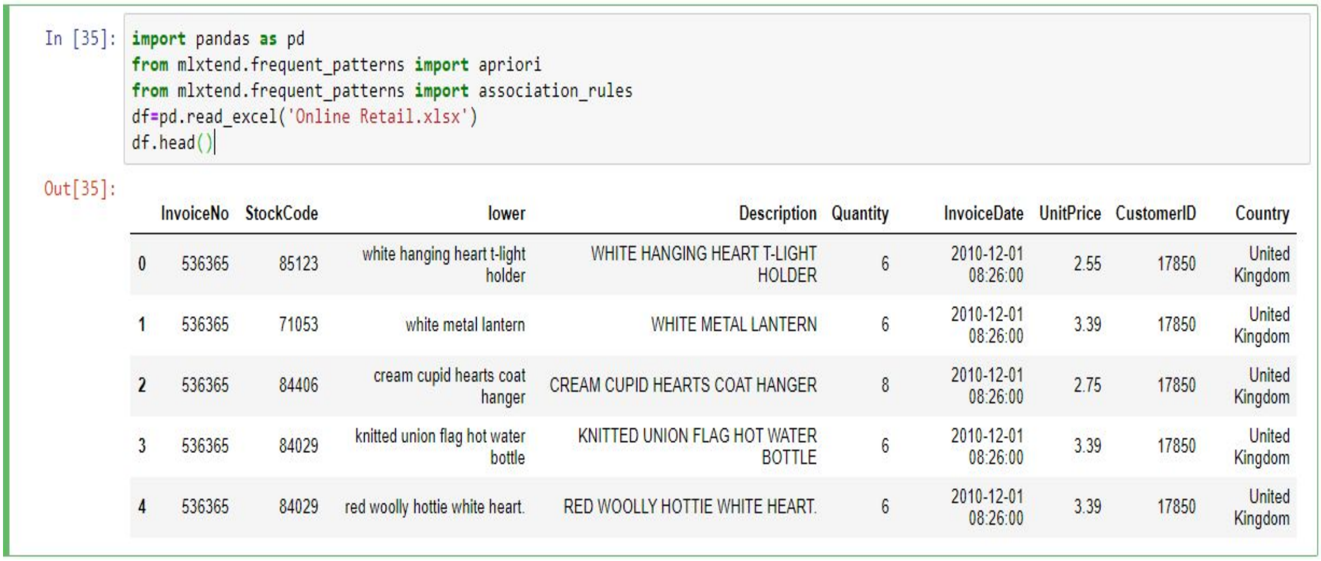
Confidence of association rule $X \rightarrow Y$：

$$Confidence\ (X \rightarrow Y) \equiv P(Y|X) = \frac{P(X,Y)}{P(X)} = \frac{\#\{\text{Customers who bought } X \text{ and } Y\}}{\#\{\text{Customers who bought } X\}} \quad (2)$$

Lift, also known as interest of association rule $X \rightarrow Y$：

$$Lift\ (X \rightarrow Y) = \frac{P(X,Y)}{P(X)\ P(Y)} = \frac{P(Y|X)}{P(XY)} \quad (3)$$

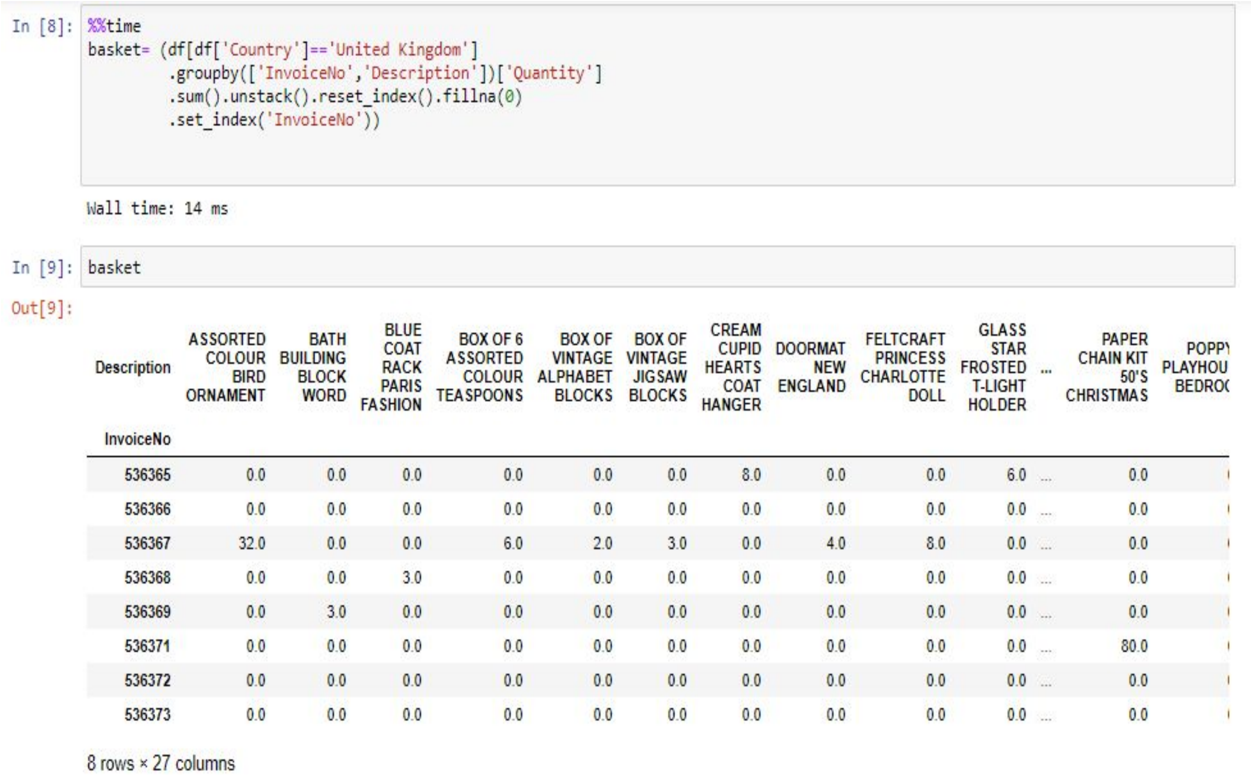
APPLICATION: APRIORI



In [35]:
```
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()
```

Out[35]:

	InvoiceNo	StockCode	lower	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123	white hanging heart t-light holder	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom
1	536365	71053	white metal lantern	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
2	536365	84406	cream cupid hearts coat hanger	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom
3	536365	84029	knitted union flag hot water bottle	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
4	536365	84029	red woolly hottie white heart.	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom



In [8]:
```
%%time
basket= (df[df['Country']=='United Kingdom']
          .groupby(['InvoiceNo','Description'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))
```

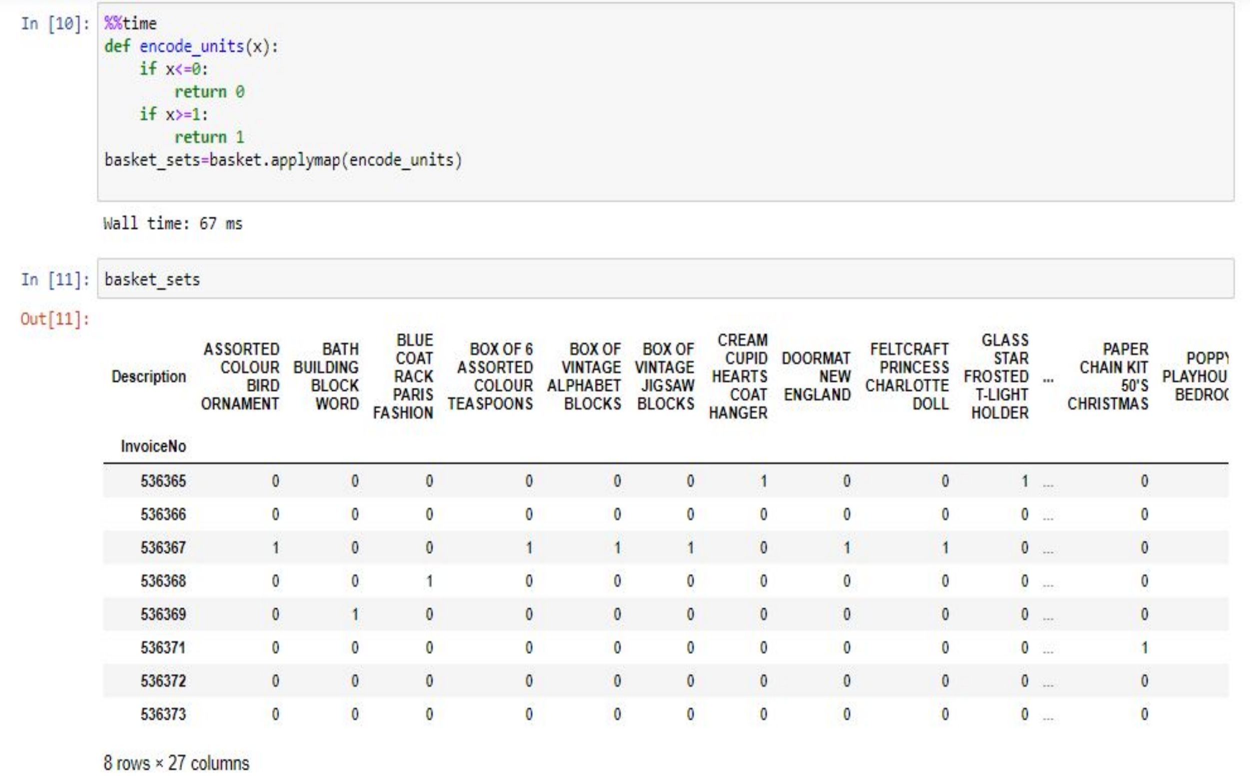
```
Wall time: 14 ms
```

In [9]:
```
basket
```

Out[9]:

Description	ASSORTED COLOUR BIRD ORNAMENT	BATH BUILDING BLOCK WORD	BLUE COAT RACK PARIS FASHION	BOX OF 6 ASSORTED COLOUR TEASPOONS	BOX OF VINTAGE ALPHABET BLOCKS	BOX OF VINTAGE JIGSAW BLOCKS	CREAM CUPID HEARTS COAT HANGER	DOORMAT NEW ENGLAND	FELTCRAFT PRINCESS CHARLOTTE DOLL	GLASS STAR FROSTED T-LIGHT HOLDER	...	PAPER CHAIN KIT 50'S CHRISTMAS	POPPY PLAYHOU BEDRO(
InvoiceNo												
536365	0.0	0.0	0.0	0.0	0.0	0.0	8.0	0.0	0.0	6.0	...	0.0	
536366	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	
536367	32.0	0.0	0.0	6.0	2.0	3.0	0.0	4.0	8.0	0.0	...	0.0	
536368	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	
536369	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	
536371	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	80.0	
536372	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	
536373	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	

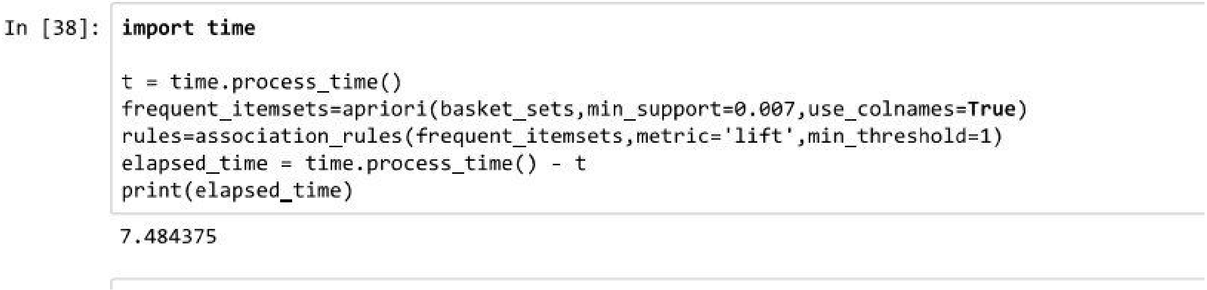
8 rows × 27 columns



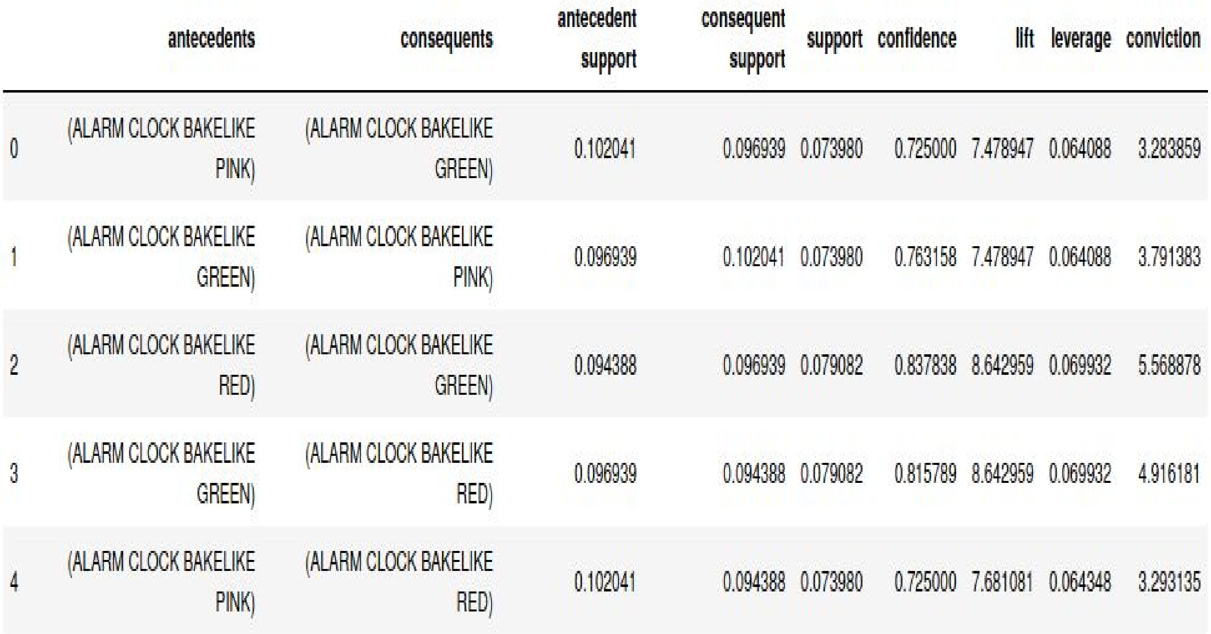
```
In [10]: %%time
         def encode_units(x):
             if x<=0:
                 return 0
             if x>=1:
                 return 1
         basket_sets=basket.applymap(encode_units)

         Wall time: 67 ms

In [11]: basket_sets

Out[11]:
```

Description	ASSORTED COLOUR BIRD ORNAMENT	BATH BUILDING BLOCK WORD	BLUE COAT RACK PARIS FASHION	BOX OF 6 ASSORTED COLOUR TEASPOONS	BOX OF VINTAGE ALPHABET BLOCKS	BOX OF VINTAGE JIGSAW BLOCKS	CREAM CUPID HEARTS COAT HANGER	DOORMAT NEW ENGLAND	FELTCRAFT PRINCESS CHARLOTTE DOLL	GLASS STAR FROSTED T-LIGHT HOLDER	...	PAPER CHAIN KIT 50'S CHRISTMAS	POPPY PLAYHOU BEDROO
InvoiceNo													
536365	0	0	0	0	0	0	1	0	0	1	...	0	
536366	0	0	0	0	0	0	0	0	0	0	...	0	
536367	1	0	0	1	1	1	0	1	1	0	...	0	
536368	0	0	1	0	0	0	0	0	0	0	...	0	
536369	0	1	0	0	0	0	0	0	0	0	...	0	
536371	0	0	0	0	0	0	0	0	0	0	...	1	
536372	0	0	0	0	0	0	0	0	0	0	...	0	
536373	0	0	0	0	0	0	0	0	0	0	...	0	

8 rows × 27 columns



```
In [38]: import time

         t = time.process_time()
         frequent_itemsets=apriori(basket_sets,min_support=0.007,use_colnames=True)
         rules=association_rules(frequent_itemsets,metric='lift',min_threshold=1)
         elapsed_time = time.process_time() - t
         print(elapsed_time)

         7.484375
```



	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(ALARM CLOCK BAKELIKE PINK)	(ALARM CLOCK BAKELIKE GREEN)	0.102041	0.096939	0.073980	0.725000	7.478947	0.064088	3.283859
1	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE PINK)	0.096939	0.102041	0.073980	0.763158	7.478947	0.064088	3.791383
2	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.094388	0.096939	0.079082	0.837838	8.642959	0.069932	5.568878
3	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.096939	0.094388	0.079082	0.815789	8.642959	0.069932	4.916181
4	(ALARM CLOCK BAKELIKE PINK)	(ALARM CLOCK BAKELIKE RED)	0.102041	0.094388	0.073980	0.725000	7.681081	0.064348	3.293135

K-Means

K-means clustering is a method of cluster analysis which aims to partition n observations into K clusters depending on some similarity/dissimilarity metric where the value of K may or may not be known a priori.[7] The algorithm assigns each point to the cluster whose center (also called centroid) is the nearest. The center is the average of all the points in the cluster, that is, its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster.

Algorithm:

INPUT:

K: no. of cluster

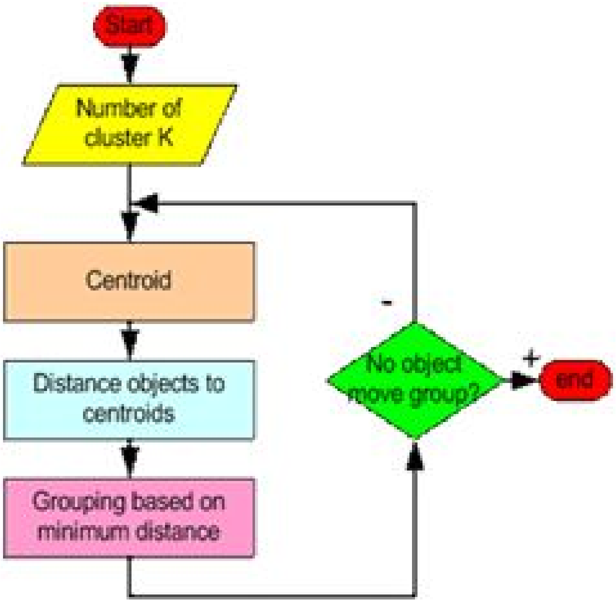
D: dataset containing n objects

METHOD:

1. Arbitrarily choose k objects in D as the initial cluster centre.
2. Calculate the distance between each data point and cluster centres.
3. Assign the data point to the cluster centre whose distance from the cluster centre is minimum of all the cluster centres.
4. When all the objects are placed recalculate the centroid k position.
5. Repeat steps 2 and 3 until the position of k is no longer moved.

Output:

A set of k clusters.


Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
-
No object move group?
+
end

Flow chart of K-means

- Initial centroids are often chosen randomly.

  Clusters produced vary from one run to another

- The centroid is (typically) the mean of the points in the cluster.
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.

Euclidean Distance: The formula for this distance between a point *X* (*X1, X2,* etc.) and a point *Y* (*Y1, Y2,* etc.) is:

$$d= \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations.

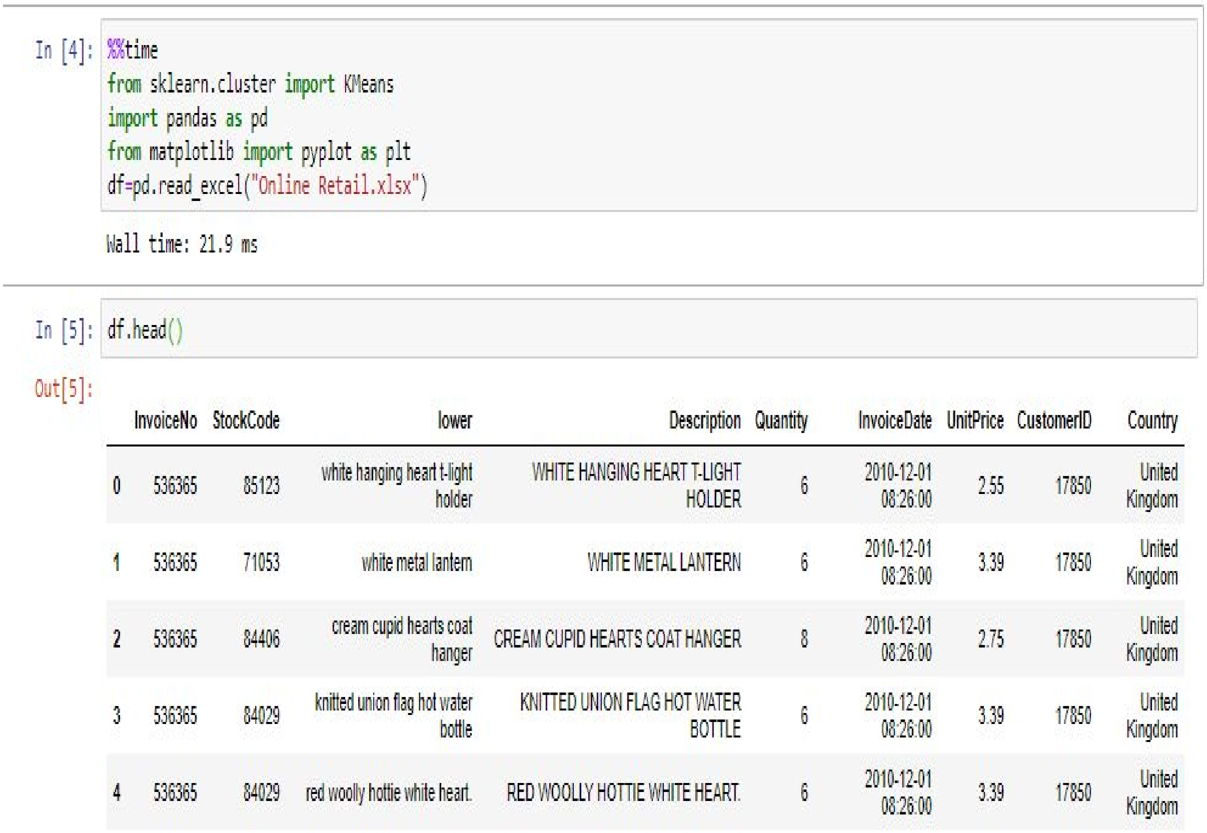
  Often the stopping condition is changed to ‘Until relatively few points change clusters’

- Complexity is O(**n * K * I * d**)

  n = number of points, K = number of clusters,

  I = number of iterations, d = number of attributes

APPLICATION: KMEANS+APRIORI

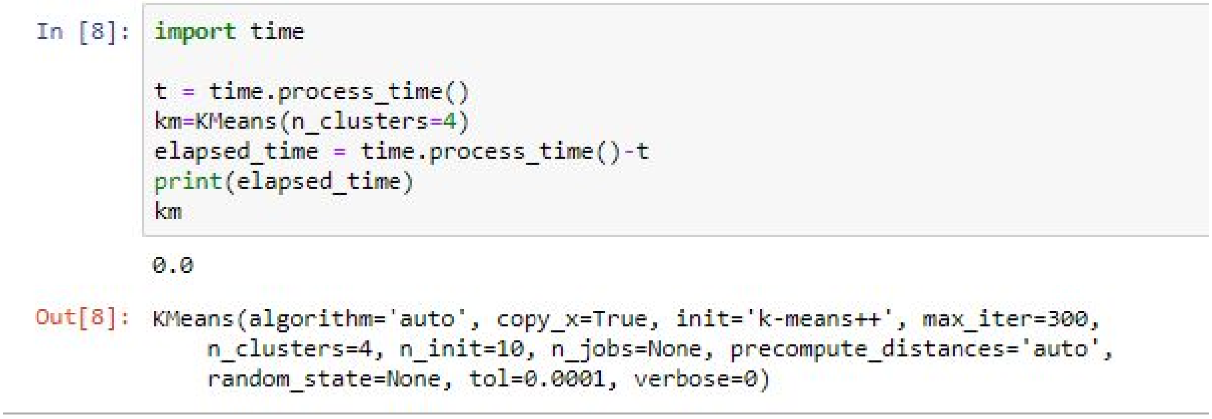


```
In [4]: %%time
        from sklearn.cluster import KMeans
        import pandas as pd
        from matplotlib import pyplot as plt
        df=pd.read_excel("Online Retail.xlsx")

        Wall time: 21.9 ms
```

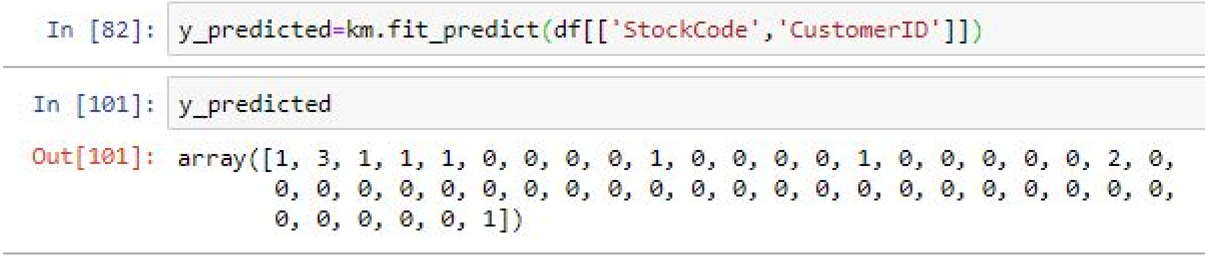
```
In [5]: df.head()
```

Out[5]:

	InvoiceNo	StockCode	lower	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123	white hanging heart t-light holder	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom
1	536365	71053	white metal lantern	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
2	536365	84406	cream cupid hearts coat hanger	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom
3	536365	84029	knitted union flag hot water bottle	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
4	536365	84029	red woolly hottie white heart.	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom



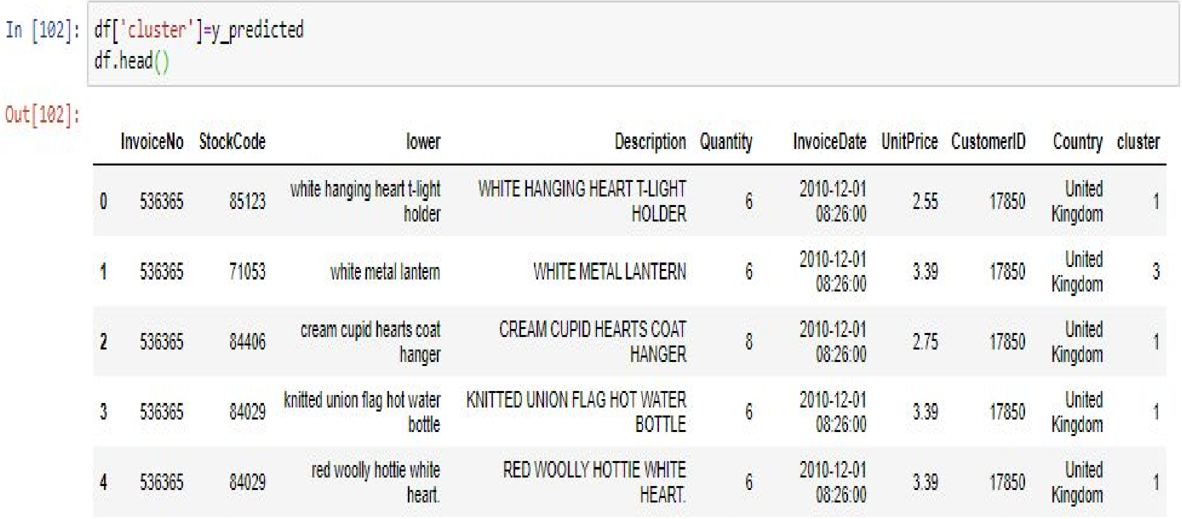
```
In [8]: import time

        t = time.process_time()
        km=KMeans(n_clusters=4)
        elapsed_time = time.process_time()-t
        print(elapsed_time)
        km

        0.0

Out[8]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
            n_clusters=4, n_init=10, n_jobs=None, precompute_distances='auto',
            random_state=None, tol=0.0001, verbose=0)
```



```
In [82]: y_predicted=km.fit_predict(df[['StockCode','CustomerID']])
```

```
In [101]: y_predicted
```

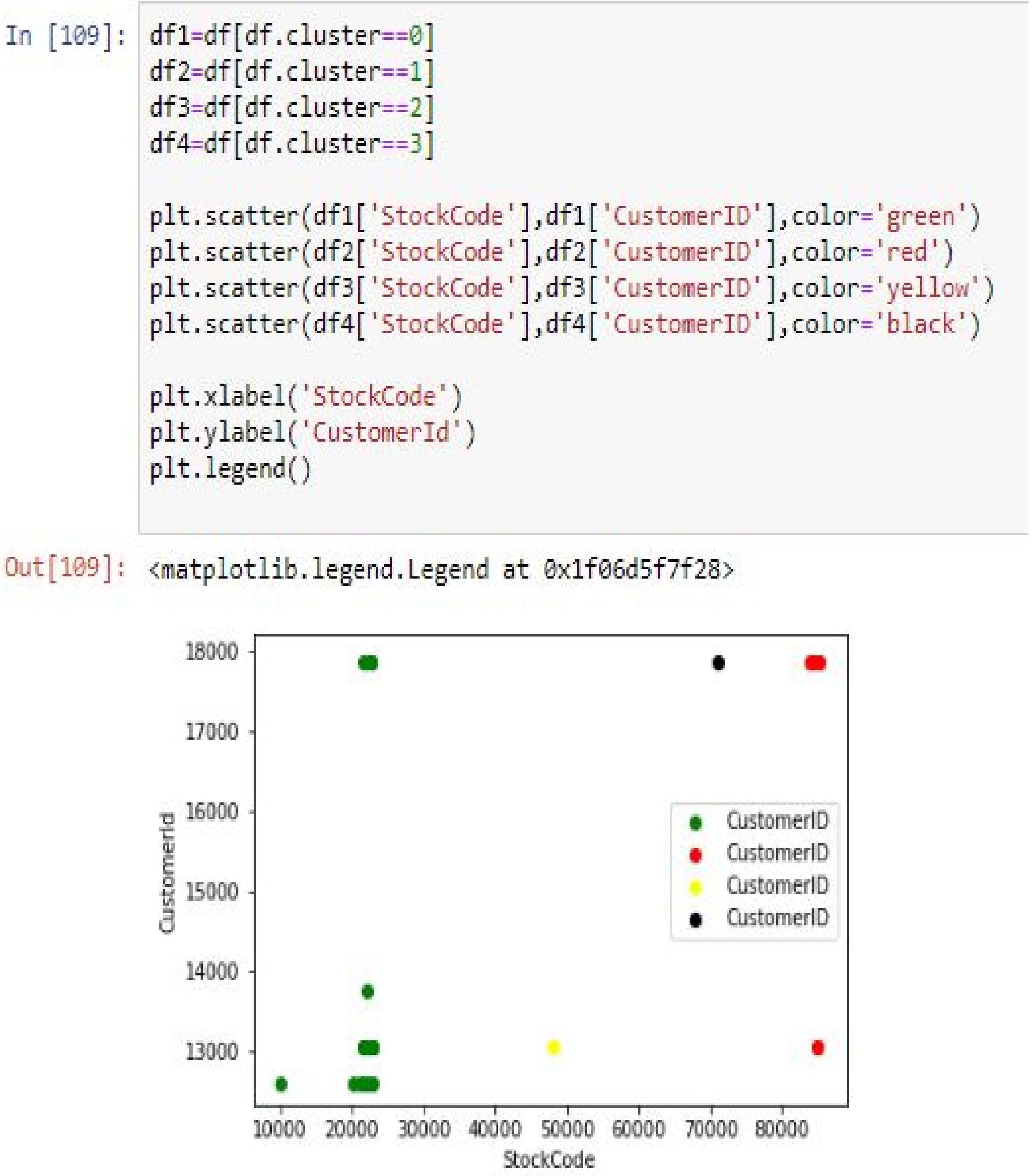
```
Out[101]: array([1, 3, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 2, 0,
                 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                 0, 0, 0, 0, 0, 1])
```



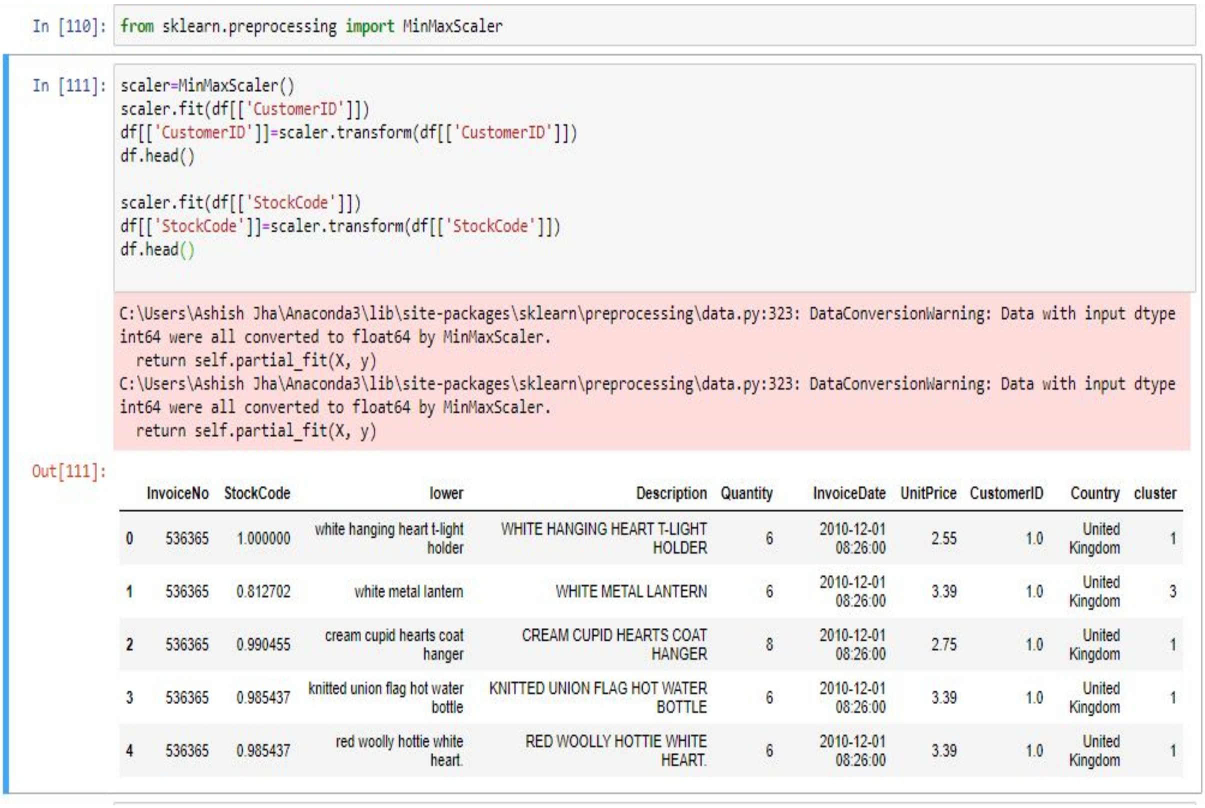
```
In [102]: df['cluster']=y_predicted
          df.head()
```

Out[102]:

	InvoiceNo	StockCode	lower	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	cluster
0	536365	85123	white hanging heart t-light holder	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom	1
1	536365	71053	white metal lantern	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	3
2	536365	84406	cream cupid hearts coat hanger	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom	1
3	536365	84029	knitted union flag hot water bottle	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	1
4	536365	84029	red woolly hottie white heart.	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	1



```
In [109]: df1=df[df.cluster==0]
          df2=df[df.cluster==1]
          df3=df[df.cluster==2]
          df4=df[df.cluster==3]

          plt.scatter(df1['StockCode'],df1['CustomerID'],color='green')
          plt.scatter(df2['StockCode'],df2['CustomerID'],color='red')
          plt.scatter(df3['StockCode'],df3['CustomerID'],color='yellow')
          plt.scatter(df4['StockCode'],df4['CustomerID'],color='black')

          plt.xlabel('StockCode')
          plt.ylabel('CustomerId')
          plt.legend()

Out[109]: <matplotlib.legend.Legend at 0x1f06d5f7f28>
```



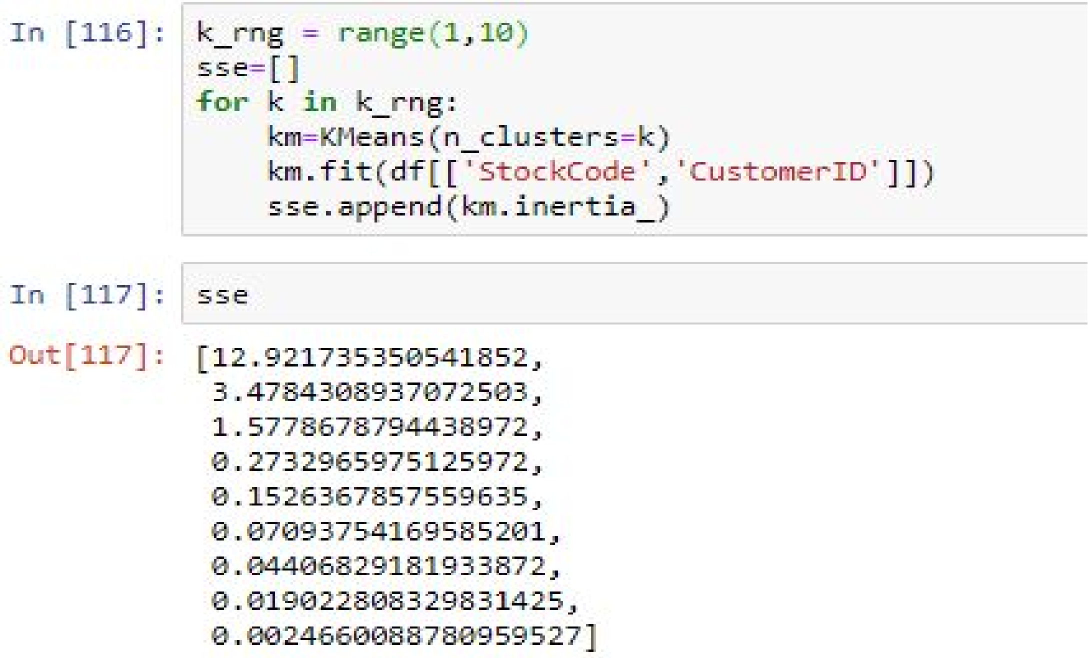
```
In [110]: from sklearn.preprocessing import MinMaxScaler
```

```
In [111]: scaler=MinMaxScaler()
          scaler.fit(df[['CustomerID']])
          df[['CustomerID']]=scaler.transform(df[['CustomerID']])
          df.head()

          scaler.fit(df[['StockCode']])
          df[['StockCode']]=scaler.transform(df[['StockCode']])
          df.head()
```

```
C:\Users\Ashish Jha\Anaconda3\lib\site-packages\sklearn\preprocessing\data.py:323: DataConversionWarning: Data with input dtype
int64 were all converted to float64 by MinMaxScaler.
  return self.partial_fit(X, y)
C:\Users\Ashish Jha\Anaconda3\lib\site-packages\sklearn\preprocessing\data.py:323: DataConversionWarning: Data with input dtype
int64 were all converted to float64 by MinMaxScaler.
  return self.partial_fit(X, y)
```

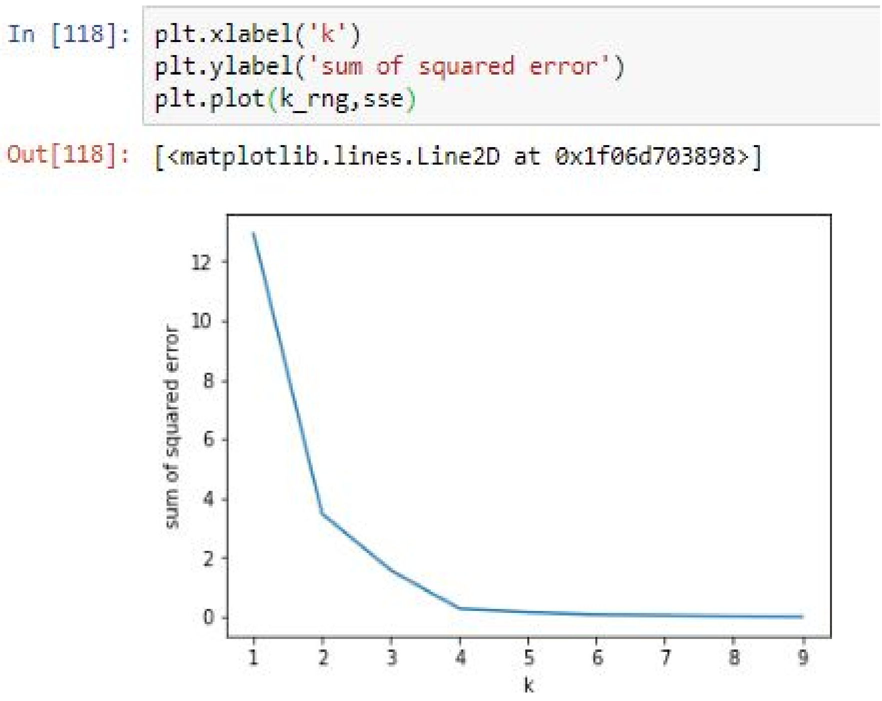
Out[111]:

	InvoiceNo	StockCode	lower	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	cluster
0	536365	1.000000	white hanging heart t-light holder	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	1.0	United Kingdom	1
1	536365	0.812702	white metal lantern	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	1.0	United Kingdom	3
2	536365	0.990455	cream cupid hearts coat hanger	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	1.0	United Kingdom	1
3	536365	0.985437	knitted union flag hot water bottle	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	1.0	United Kingdom	1
4	536365	0.985437	red woolly hottie white heart.	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	1.0	United Kingdom	1

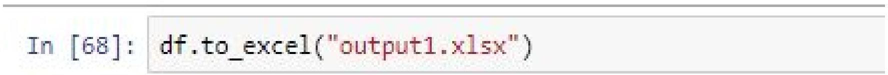


```
In [116]: k_rng = range(1,10)
          sse=[]
          for k in k_rng:
              km=KMeans(n_clusters=k)
              km.fit(df[['StockCode','CustomerID']])
              sse.append(km.inertia_)
```

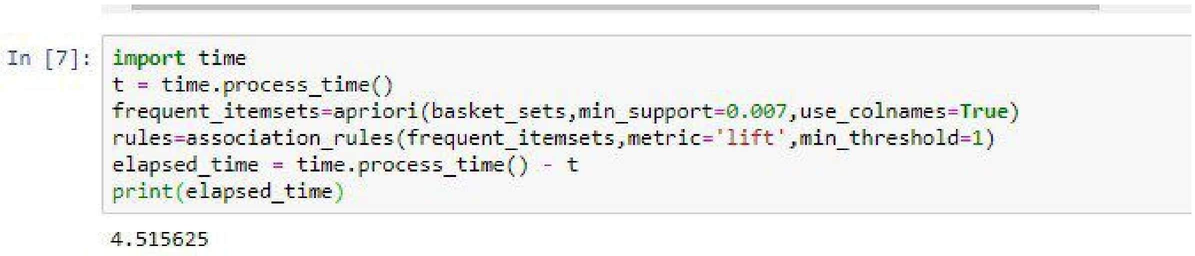
```
In [117]: sse

Out[117]: [12.921735350541852,
           3.4784308937072503,
           1.5778678794438972,
           0.2732965975125972,
           0.1526367857559635,
           0.07093754169585201,
           0.04406829181933872,
           0.019022808329831425,
           0.0024660088780959527]
```



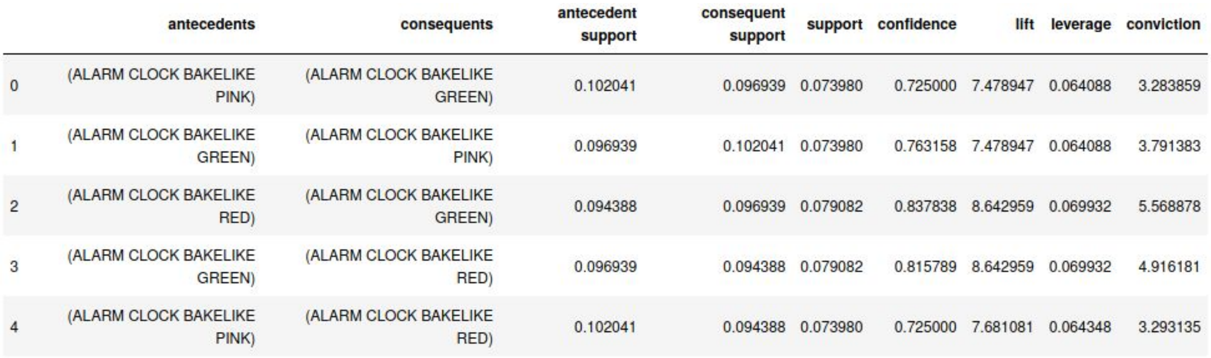
```
In [118]: plt.xlabel('k')
          plt.ylabel('sum of squared error')
          plt.plot(k_rng,sse)

Out[118]: [<matplotlib.lines.Line2D at 0x1f06d703898>]
```



```
In [68]: df.to_excel("output1.xlsx")
```

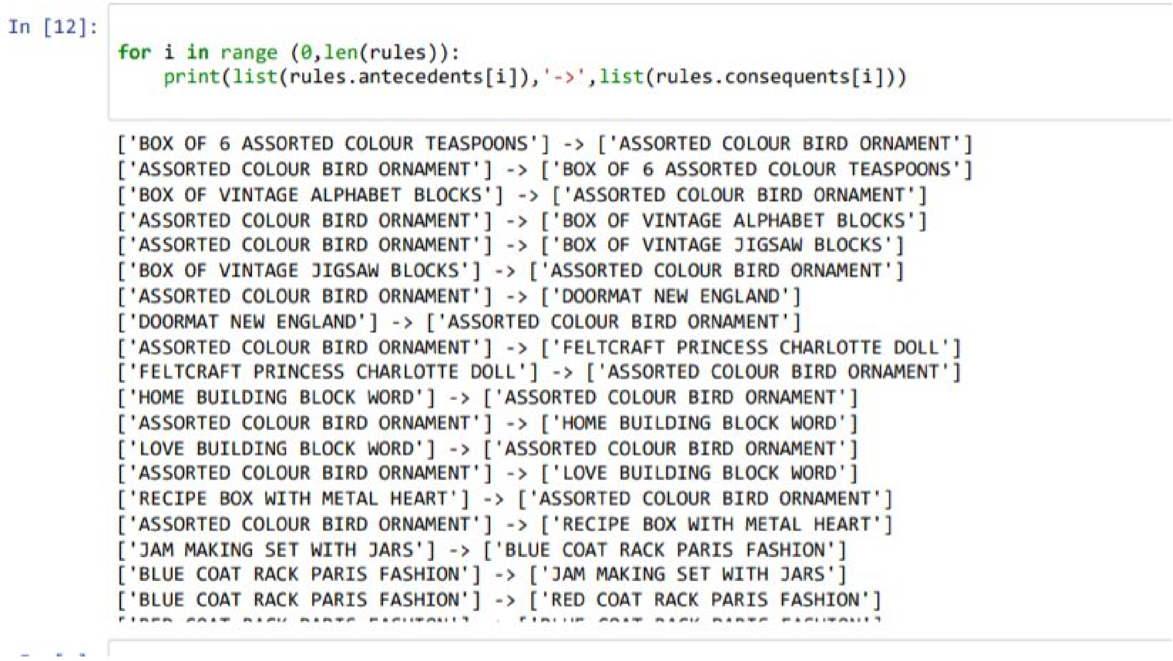


```
In [7]: import time
        t = time.process_time()
        frequent_itemsets=apriori(basket_sets,min_support=0.007,use_colnames=True)
        rules=association_rules(frequent_itemsets,metric='lift',min_threshold=1)
        elapsed_time = time.process_time() - t
        print(elapsed_time)

        4.515625
```



	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(ALARM CLOCK BAKELIKE PINK)	(ALARM CLOCK BAKELIKE GREEN)	0.102041	0.096939	0.073980	0.725000	7.478947	0.064088	3.283859
1	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE PINK)	0.096939	0.102041	0.073980	0.763158	7.478947	0.064088	3.791383
2	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.094388	0.096939	0.079082	0.837838	8.642959	0.069932	5.568878
3	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.096939	0.094388	0.079082	0.815789	8.642959	0.069932	4.916181
4	(ALARM CLOCK BAKELIKE PINK)	(ALARM CLOCK BAKELIKE RED)	0.102041	0.094388	0.073980	0.725000	7.681081	0.064348	3.293135

Link Prediction using K-Means and Apriori



```
In [12]:  for i in range (0,len(rules)):
              print(list(rules.antecedents[i]),'->',list(rules.consequents[i]))
```

```
['BOX OF 6 ASSORTED COLOUR TEASPOONS'] -> ['ASSORTED COLOUR BIRD ORNAMENT']
['ASSORTED COLOUR BIRD ORNAMENT'] -> ['BOX OF 6 ASSORTED COLOUR TEASPOONS']
['BOX OF VINTAGE ALPHABET BLOCKS'] -> ['ASSORTED COLOUR BIRD ORNAMENT']
['ASSORTED COLOUR BIRD ORNAMENT'] -> ['BOX OF VINTAGE ALPHABET BLOCKS']
['ASSORTED COLOUR BIRD ORNAMENT'] -> ['BOX OF VINTAGE JIGSAW BLOCKS']
['BOX OF VINTAGE JIGSAW BLOCKS'] -> ['ASSORTED COLOUR BIRD ORNAMENT']
['ASSORTED COLOUR BIRD ORNAMENT'] -> ['DOORMAT NEW ENGLAND']
['DOORMAT NEW ENGLAND'] -> ['ASSORTED COLOUR BIRD ORNAMENT']
['ASSORTED COLOUR BIRD ORNAMENT'] -> ['FELTCRAFT PRINCESS CHARLOTTE DOLL']
['FELTCRAFT PRINCESS CHARLOTTE DOLL'] -> ['ASSORTED COLOUR BIRD ORNAMENT']
['HOME BUILDING BLOCK WORD'] -> ['ASSORTED COLOUR BIRD ORNAMENT']
['ASSORTED COLOUR BIRD ORNAMENT'] -> ['HOME BUILDING BLOCK WORD']
['LOVE BUILDING BLOCK WORD'] -> ['ASSORTED COLOUR BIRD ORNAMENT']
['ASSORTED COLOUR BIRD ORNAMENT'] -> ['LOVE BUILDING BLOCK WORD']
['RECIPE BOX WITH METAL HEART'] -> ['ASSORTED COLOUR BIRD ORNAMENT']
['ASSORTED COLOUR BIRD ORNAMENT'] -> ['RECIPE BOX WITH METAL HEART']
['JAM MAKING SET WITH JARS'] -> ['BLUE COAT RACK PARIS FASHION']
['BLUE COAT RACK PARIS FASHION'] -> ['JAM MAKING SET WITH JARS']
['BLUE COAT RACK PARIS FASHION'] -> ['RED COAT RACK PARIS FASHION']
['RED COAT RACK PARIS FASHION'] -> ['BLUE COAT RACK PARIS FASHION']
```

DBSCAN

(Density-Based Spatial Clustering of Applications with Noise)

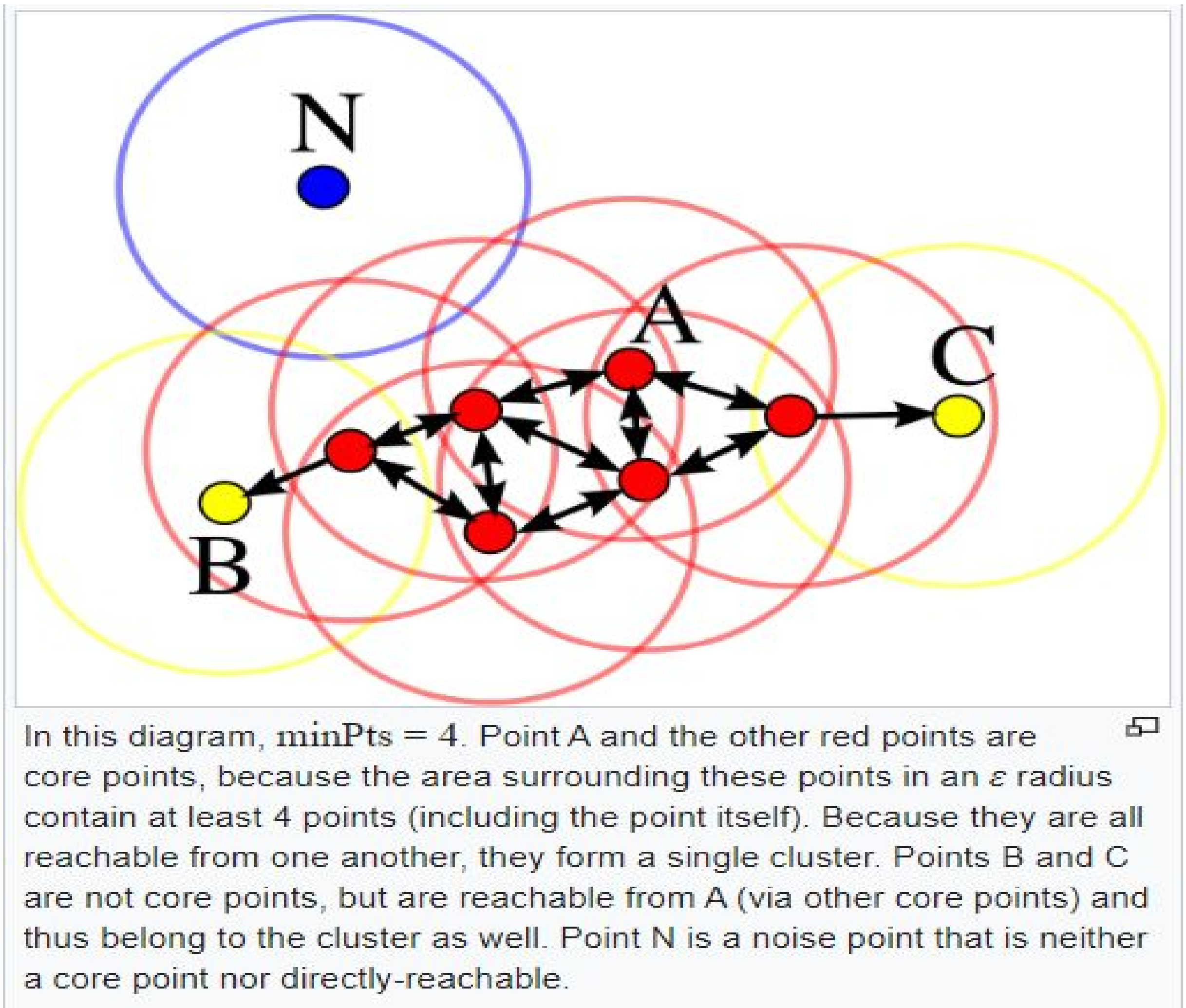
It is a popular unsupervised learning method utilized in model building and machine learning algorithms. It is a clustering method that is used in machine learning to separate clusters of high density from clusters of low density. Given that DBSCAN is a density based clustering algorithm, it does a great job of seeking areas in the data that have a high density of observations, versus areas of the data that are not very dense with observations. DBSCAN can sort data into clusters of varying shapes as well,another strong advantage.

DBSCAN works as such:

Divides the dataset into n dimensions For each point in the dataset, DBSCAN forms an n dimensional shape around that data point,and then counts how many data points fall within that shape.

DBSCAN counts this shape as a cluster. DBSCAN iteratively expands the cluster, by going through each individual point within the cluster, and counting the number of other data points nearby. Going through the aforementioned process step-by-step, DBSCAN will start by dividing the data into n dimensions.

After DBSCAN has done so, it will start at a random point (in this case let's assume it was one of the red points), and it will count how many other points are nearby. DBSCAN will continue this process until no other data points are nearby, and then it will look to form a second cluster.



In this diagram, $\mathrm{minPts} = 4$. Point A and the other red points are core points, because the area surrounding these points in an ε radius contain at least 4 points (including the point itself). Because they are all reachable from one another, they form a single cluster. Points B and C are not core points, but are reachable from A (via other core points) and thus belong to the cluster as well. Point N is a noise point that is neither a core point nor directly-reachable.

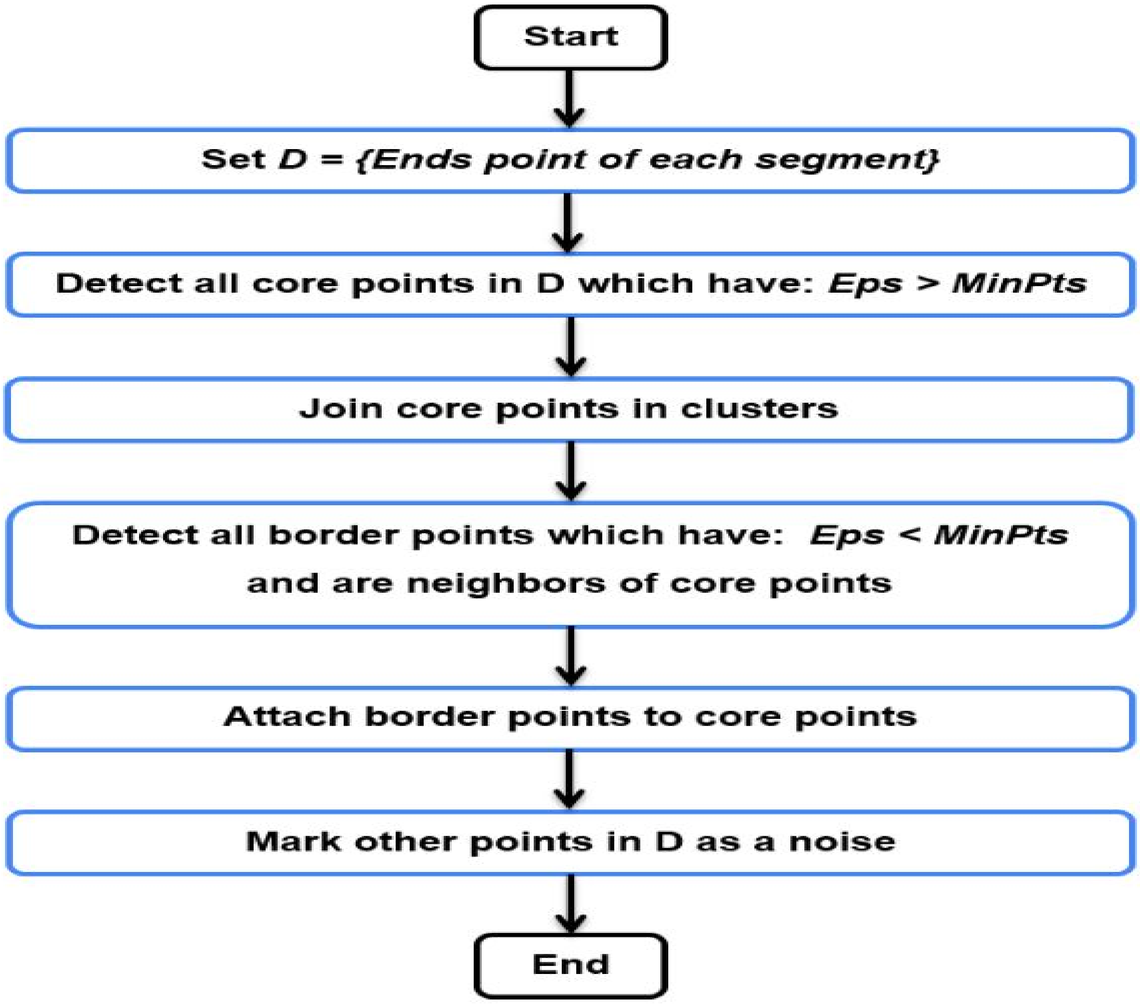
Original Query-based algorithm :

DBSCAN requires two parameters: ε (eps) and the minimum number of points required to form a dense region (minPts). It starts with an arbitrary starting point that has not been visited. This point's ε-neighborhood is retrieved, and if it contains sufficiently many points, a cluster is started. Otherwise, the point is labeled as noise. Note that this point might later be found in a sufficiently sized ε-environment of a different point and hence be made part of a cluster.

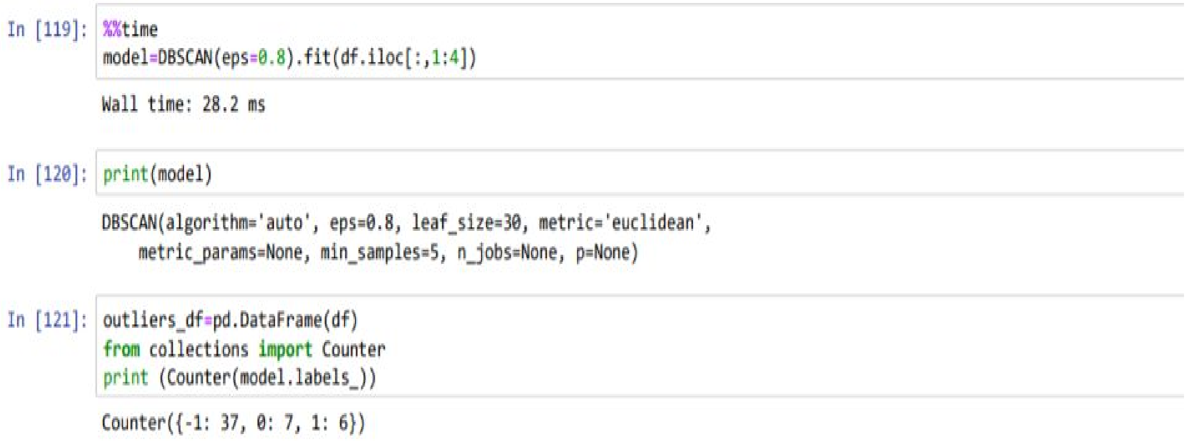
If a point is found to be a dense part of a cluster, its ε-neighborhood is also part of that cluster. Hence, all points that are found within the ε-neighborhood are added, as is their own ε-neighborhood when they are also dense. This process continues until the density-connected cluster is completely found. Then, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.

DBSCAN can be used with any distance function as well as similarity functions or other predicates. The distance function can therefore be seen as an additional parameter.

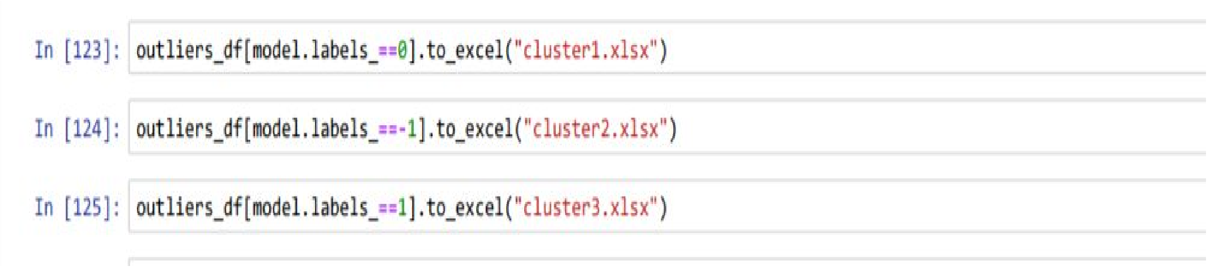
Flowchart of the DBSCAN :


Start
Set *D* = *{Ends point of each segment}*
Detect all core points in D which have: *Eps* > *MinPts*
Join core points in clusters
Detect all border points which have: *Eps* < *MinPts* and are neighbors of core points
Attach border points to core points
Mark other points in D as a noise
End

DBSCAN on the retail data :



```
In [119]: %%time
          model=DBSCAN(eps=0.8).fit(df.iloc[:,1:4])

          Wall time: 28.2 ms

In [120]: print(model)

          DBSCAN(algorithm='auto', eps=0.8, leaf_size=30, metric='euclidean',
              metric_params=None, min_samples=5, n_jobs=None, p=None)

In [121]: outliers_df=pd.DataFrame(df)
          from collections import Counter
          print (Counter(model.labels_))

          Counter({-1: 37, 0: 7, 1: 6})
```

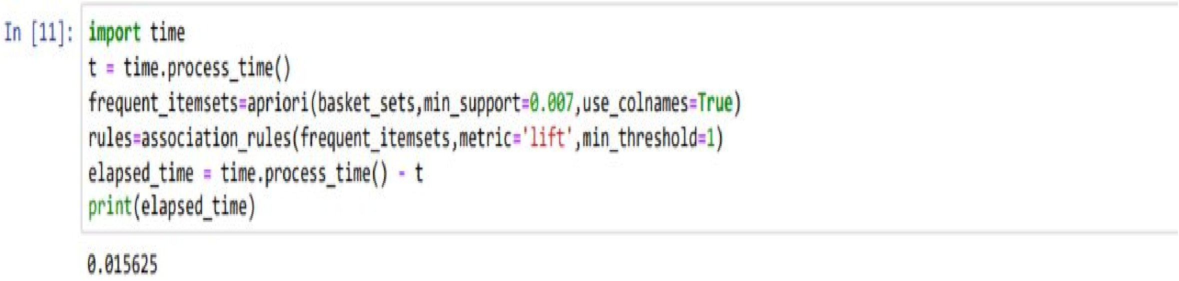


```
In [123]: outliers_df[model.labels_==0].to_excel("cluster1.xlsx")

In [124]: outliers_df[model.labels_==-1].to_excel("cluster2.xlsx")

In [125]: outliers_df[model.labels_==1].to_excel("cluster3.xlsx")
```

After applying DBSCAN we classified data into three clusters and stored it into excel file

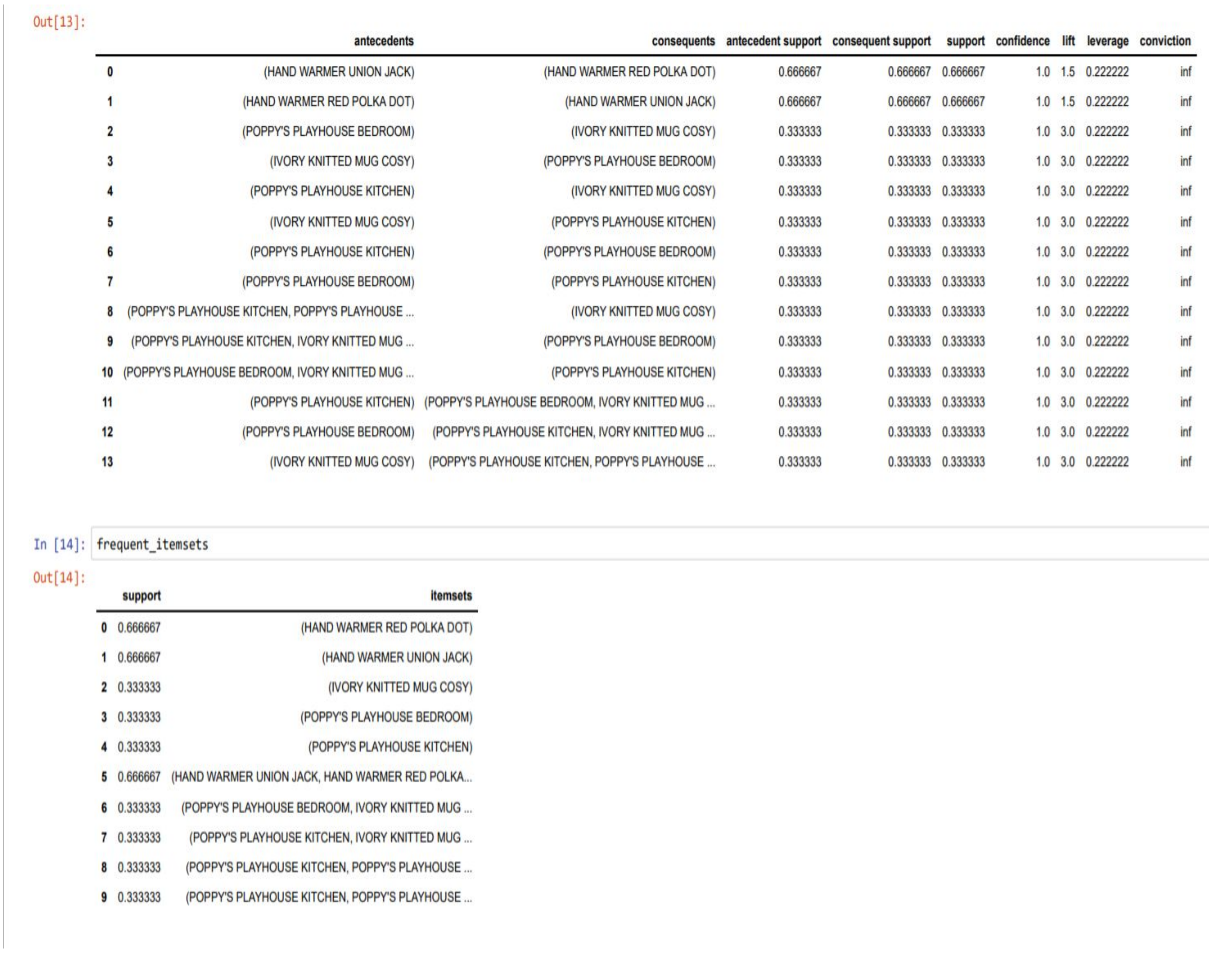
We then applied apriori algorithm for predicting link among the items and we observed the time it took to predict the link, following is our finding

Application: DBSCAN + Apriori



In [11]:
```
import time
t = time.process_time()
frequent_itemsets=apriori(basket_sets,min_support=0.007,use_colnames=True)
rules=association_rules(frequent_itemsets,metric='lift',min_threshold=1)
elapsed_time = time.process_time() - t
print(elapsed_time)
```

```
0.015625
```



Out[13]:

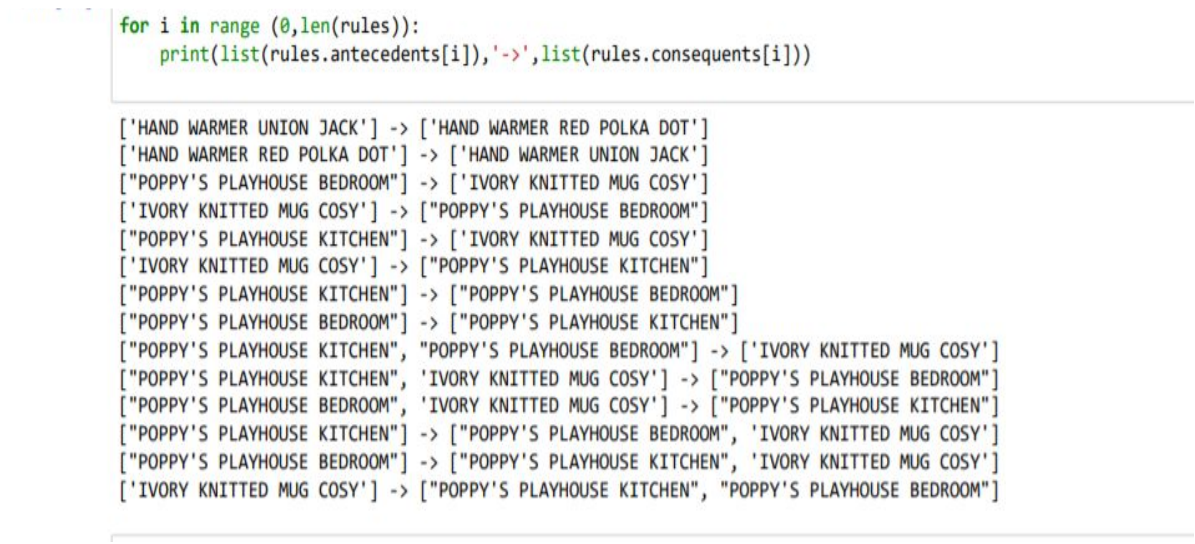
	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(HAND WARMER UNION JACK)	(HAND WARMER RED POLKA DOT)	0.666667	0.666667	0.666667	1.0	1.5	0.222222	inf
1	(HAND WARMER RED POLKA DOT)	(HAND WARMER UNION JACK)	0.666667	0.666667	0.666667	1.0	1.5	0.222222	inf
2	(POPPY'S PLAYHOUSE BEDROOM)	(IVORY KNITTED MUG COSY)	0.333333	0.333333	0.333333	1.0	3.0	0.222222	inf
3	(IVORY KNITTED MUG COSY)	(POPPY'S PLAYHOUSE BEDROOM)	0.333333	0.333333	0.333333	1.0	3.0	0.222222	inf
4	(POPPY'S PLAYHOUSE KITCHEN)	(IVORY KNITTED MUG COSY)	0.333333	0.333333	0.333333	1.0	3.0	0.222222	inf
5	(IVORY KNITTED MUG COSY)	(POPPY'S PLAYHOUSE KITCHEN)	0.333333	0.333333	0.333333	1.0	3.0	0.222222	inf
6	(POPPY'S PLAYHOUSE KITCHEN)	(POPPY'S PLAYHOUSE BEDROOM)	0.333333	0.333333	0.333333	1.0	3.0	0.222222	inf
7	(POPPY'S PLAYHOUSE BEDROOM)	(POPPY'S PLAYHOUSE KITCHEN)	0.333333	0.333333	0.333333	1.0	3.0	0.222222	inf
8	(POPPY'S PLAYHOUSE KITCHEN, POPPY'S PLAYHOUSE ...	(IVORY KNITTED MUG COSY)	0.333333	0.333333	0.333333	1.0	3.0	0.222222	inf
9	(POPPY'S PLAYHOUSE KITCHEN, IVORY KNITTED MUG ...	(POPPY'S PLAYHOUSE BEDROOM)	0.333333	0.333333	0.333333	1.0	3.0	0.222222	inf
10	(POPPY'S PLAYHOUSE BEDROOM, IVORY KNITTED MUG ...	(POPPY'S PLAYHOUSE KITCHEN)	0.333333	0.333333	0.333333	1.0	3.0	0.222222	inf
11	(POPPY'S PLAYHOUSE KITCHEN)	(POPPY'S PLAYHOUSE BEDROOM, IVORY KNITTED MUG ...	0.333333	0.333333	0.333333	1.0	3.0	0.222222	inf
12	(POPPY'S PLAYHOUSE BEDROOM)	(POPPY'S PLAYHOUSE KITCHEN, IVORY KNITTED MUG ...	0.333333	0.333333	0.333333	1.0	3.0	0.222222	inf
13	(IVORY KNITTED MUG COSY)	(POPPY'S PLAYHOUSE KITCHEN, POPPY'S PLAYHOUSE ...	0.333333	0.333333	0.333333	1.0	3.0	0.222222	inf

In [14]:
```
frequent_itemsets
```

Out[14]:

	support	itemsets
0	0.666667	(HAND WARMER RED POLKA DOT)
1	0.666667	(HAND WARMER UNION JACK)
2	0.333333	(IVORY KNITTED MUG COSY)
3	0.333333	(POPPY'S PLAYHOUSE BEDROOM)
4	0.333333	(POPPY'S PLAYHOUSE KITCHEN)
5	0.666667	(HAND WARMER UNION JACK, HAND WARMER RED POLKA...
6	0.333333	(POPPY'S PLAYHOUSE BEDROOM, IVORY KNITTED MUG ...
7	0.333333	(POPPY'S PLAYHOUSE KITCHEN, IVORY KNITTED MUG ...
8	0.333333	(POPPY'S PLAYHOUSE KITCHEN, POPPY'S PLAYHOUSE ...
9	0.333333	(POPPY'S PLAYHOUSE KITCHEN, POPPY'S PLAYHOUSE ...

Predicting Link by applying: DBSCAN + Apriori



```
for i in range (0,len(rules)):
    print(list(rules.antecedents[i]),'->',list(rules.consequents[i]))
```

```
['HAND WARMER UNION JACK'] -> ['HAND WARMER RED POLKA DOT']
['HAND WARMER RED POLKA DOT'] -> ['HAND WARMER UNION JACK']
["POPPY'S PLAYHOUSE BEDROOM"] -> ['IVORY KNITTED MUG COSY']
['IVORY KNITTED MUG COSY'] -> ["POPPY'S PLAYHOUSE BEDROOM"]
["POPPY'S PLAYHOUSE KITCHEN"] -> ['IVORY KNITTED MUG COSY']
['IVORY KNITTED MUG COSY'] -> ["POPPY'S PLAYHOUSE KITCHEN"]
["POPPY'S PLAYHOUSE KITCHEN"] -> ["POPPY'S PLAYHOUSE BEDROOM"]
["POPPY'S PLAYHOUSE BEDROOM"] -> ["POPPY'S PLAYHOUSE KITCHEN"]
["POPPY'S PLAYHOUSE KITCHEN", "POPPY'S PLAYHOUSE BEDROOM"] -> ['IVORY KNITTED MUG COSY']
["POPPY'S PLAYHOUSE KITCHEN", 'IVORY KNITTED MUG COSY'] -> ["POPPY'S PLAYHOUSE BEDROOM"]
["POPPY'S PLAYHOUSE BEDROOM", 'IVORY KNITTED MUG COSY'] -> ["POPPY'S PLAYHOUSE KITCHEN"]
["POPPY'S PLAYHOUSE KITCHEN"] -> ["POPPY'S PLAYHOUSE BEDROOM", 'IVORY KNITTED MUG COSY']
["POPPY'S PLAYHOUSE BEDROOM"] -> ["POPPY'S PLAYHOUSE KITCHEN", 'IVORY KNITTED MUG COSY']
['IVORY KNITTED MUG COSY'] -> ["POPPY'S PLAYHOUSE KITCHEN", "POPPY'S PLAYHOUSE BEDROOM"]
```

Time Comparison :

Algorithm Name	**Apriori**	**KMeans+ Apriori**	**DBSCAN+Apriori**
Time Taken	***7.48435***	***4.515625***	***0.15625***

Comparison between different approaches

Accuracy Comparison :

Algorithm Applied	**Expected Accuracy**
Apriori	**70-75%**
K-Means	**50-55%**
DB Scan	**35-40%**

Expected Accuracy Based on time of execution

Conclusion

The intense competition and increased choices available for customers have created new pressures on marketing decision-makers and there has emerged a need to manage customers in a long-term relationship. If companies make sense of customer needs and manage the relationships more intelligently, it is obvious that they will provide crucial competitive differentiation to gain market share and retain customers.

Customer retention marketing is a tactically-driven approach based on customer behavior.This study uses a research framework which can be appropriate for any sector to mine customer knowledge. The data mining is realized using two of the most known data mining algorithms: Apriori algorithm and K-means algorithm. They both help us to find association rules. We have compared the resulting association rules in two different data analysis approaches. In the first analysis, data are not clustered, whereas in the second analysis data are clustered,and the time of execution is found in both cases and mentioned above.

Both approaches then use the Apriori algorithm to extract related association rules. In this manner, we aim at determining the impact of clustering into our case study.As the case study, we have chosen the Online Retail Data. It is doubtless that our customer knowledge of lots of users is not enough to extract general strategies for Online Shopping sites and it is not reasonable to state the results as certain especially in such a tremendously changing market. We have chosen this area for demonstrating the usefulness of the approach in a simple case.

In data mining, in order to get more reliable results, it is important to implement different algorithms and find the one with the best predictions.In future we will implement more clustering algorithms and compare their results, especially for this kind of small datasets.

REFERENCES

[1].Hand, David J. "Data Mining." *Encyclopedia of Environmetrics* 2 (2006).

[2].Hand, D. J. (2006). Data Mining. *Encyclopedia of Environmetrics*, *2*.

[3].Liu, Bing, Wynne Hsu, and Yiming Ma. "Integrating classification and association rule mining." *KDD*. Vol. 98. 1998.

[4].Al-Maolegi, Mohammed, and Bassam Arkok. "An improved Apriori algorithm for association rules." *arXiv preprint arXiv:1403.3948* (2014).

[5].Yılmaz, Nergis, and Gülfem Işıklar Alptekin. "The Effect of Clustering in the Apriori Data Mining Algorithm: A Case Study." *Proceedings of the World Congress on Engineering*. Vol. 3. 2013.

[6].Yokoi, Sho, Hiroshi Kajino, and Hisashi Kashima. "Link Prediction in Sparse Networks by Incidence Matrix Factorization." *Journal of Information Processing* 25 (2017): 477-485.

[7].Bradley, Paul S., and Usama M. Fayyad. "Refining Initial Points for K-Means Clustering." *ICML*. Vol. 98. 1998.

[8]. Aljandal, Waleed, et al. "Ontology-Aware Classification and Association Rule Mining for Interest and Link Prediction in Social Networks." AAAI Spring Symposium: Social Semantic Web: Where Web 2.0 Meets Web 3.0. 2009.

[9]. Aljandal, Waleed A. Itemset size-sensitive interestingness measures for association rule mining and link prediction. Diss. Kansas State University, 2009.