

UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING AND INFORMATICS

**A MARKET BASKET ANALYSIS MODEL TO ADDRESS VISITOR COLD START
PREDICTION USING ASSOCIATION RULES**

Name: Daniel Kamau Kimani

Registration No: P58/75966/2012

Course: MSc. Computer Science

Supervisor: Prof. Elisha T. Opiyo Omulo

Introduction

Background

E-commerce is an online transaction which normally takes place over the internet network and uses digital technology. These transactions include buying and selling of goods or services on the internet.

Many e-commerce companies like Amazon, Netflix, booking.com and Jumia build their websites with recommender systems that provide customers with personalized recommendations.

These systems face challenges and limitations that reduce their performance, e.g. recommendations' overspecialization, cold start, and difficulties when items with unequal probability distribution appear

MBA helps us identify items likely to be purchased together, and association rule mining finds correlations between items in a set of transactions. Association rule mining (ARM) identifies the association or relationship between a large set of data items and forms the base for market basket analysis. ARM has been used in different industries for example supermarkets, mail order, telemarketing, production, fraud detection of credit card and e-commerce.

Problem Statement

E-commerce websites use recommender techniques that recommend items based on user ratings and items similarities during customer interactions. These techniques give results from information retrieval system that is interpreted as a match to the user's query. Getting to know your visitors is crucial in creating a great user experience for them. However, when a recommender system meets a returning user or a user for the first time or where there is no user history about that user, the system doesn't know the personal preferences of the user. This causes a visitor cold start which makes it impossible to provide personalized recommendations.

Objectives

1. Investigate effectiveness of machine learning algorithms used in solving cold start problem in recommender systems.
2. Formulate a model to find relationship among products to be used for recommendations in a cold start problem
3. Build a prototype for (2) above
4. Evaluate the prototype and measure its results

Significance of the Study

- 1. Recommendation engine and content placement** – this will enhance recommendation of items to be purchased hence saving time for navigation and searching through an online market.
- 2. Cost estimation** - where you are successfully able to automatically complete a customer's shopping cart, you will equally be able to estimate the amount one is likely to spend.
- 3. Marketing** – this will help retailers' market with their customers electronics depending with seasons and their preferences.

Literature Review

Empirical (Related Work)

Meng Chen, Cheng Yang, Jiechao Chen², Peng Yi (2013), propose a collaborative filtering recommendation system to recommend items to new users' hence solving a user cold start problem. Social sub-community division and ontology decision models for CF are used to build relationships between user static information and dynamic preferences by learning. They further recommend improvements on the proposal to be able to recommend items to both new users and ordinary users.

Hridya Sobhanam, A.K.Mariappan (2013) proposes a solution to cold start, sparsity and overspecialization in recommender systems by combining association rules with clustering to expand a new user's profile. This achieves improvement on accuracy with 36% compared to other techniques.

Theoretical Models

Apriori Algorithm

- Was proposed by Agrawal and Srikant in 1994 and is one of the data mining algorithms that are used for MBA and mining potential association rules.
- It effectively looks at the likelihood of different elements occurring together.
- It is the algorithm behind MBA and seeks to find association rules among variables.
- It assumes that any subset of a frequent item set must be frequent E.g. For a transaction containing {Mango, Grapes, Apples} also contains {Grapes, Mango}
- According to Apriori, if {mango, Apple, Grapes} is frequent then {grapes, Mango} must also be frequent.

The algorithm produces rules and ranks them into the following metrics;

- **Support** – defines how frequent a rule occurs and is the default popularity of an item.
- **Confidence** – the degree to which a conclusion is right. E.g. likelihood that a customer who bought both item A and B. It divides the number of transactions involving A and B by the number of transactions involving B.
- **Lift** – measure of how much more likely a customer will purchase an item now that he/she intends to purchase another related item compared to the customer not purchasing the intended item. E.g. it will assume an increase in sale of item A when you sell item B.

Collaborative Filtering (CF) algorithms

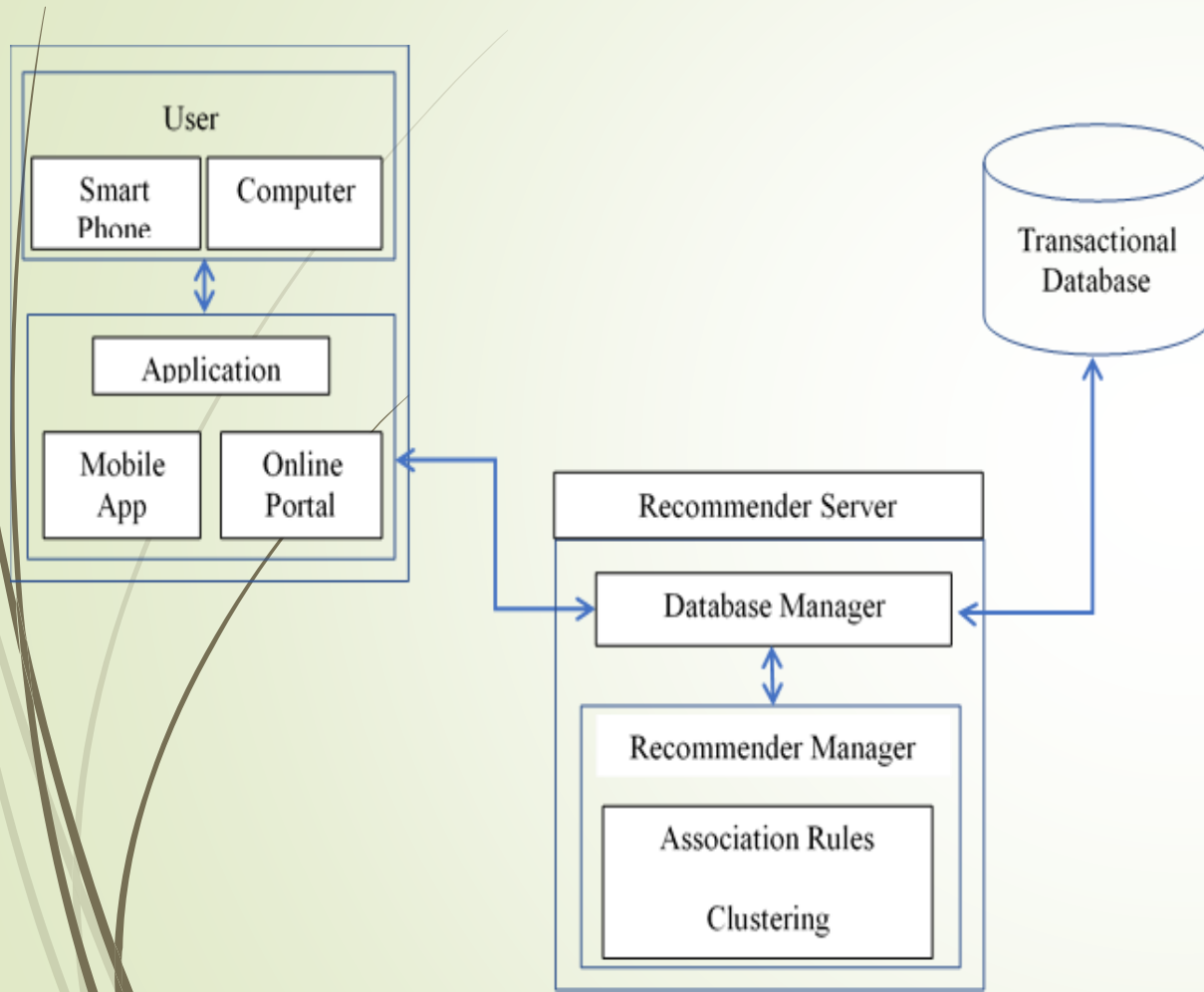
- The underlying assumption of these methods is captured by the principle of like mindedness: users who are measurably similar in their historical preferences are likely to also share similar tastes in the future.
- Assume that, in order to recommend content of any kind to users, information can be drawn from what they and other similar users have liked in the past.
- k-Nearest Neighbor (kNN) algorithm was the most favored approach to CF, since it transparently captured this assumption of like-mindedness
- operates by finding, for each user (or item), a number of similar users (items) whose profiles can then be used to directly compute recommendations

Proposed Architecture

Our approach uses association rules technique. Association rules between products that interest users can be derived from co-occurrence patterns.

This proposed solution focuses on visiting users where the system has no prior history on the users purchasing habits but can still recommend products to them through product association rules. Association in our case will be looking for products that co-occur. This is derived from past purchase history of other users from the available datasets.

Proposed Architecture



The process starts where a user selects an item/product, a backend engine generates itemsets and then association rules as per the set support threshold from historical purchases, if customers item matches an antecedent in the rules generated, the items on the consequent of the rules and with the highest lift measure are picked for recommendations to the customer.

CHAPTER 3: METHODOLOGY

We will follow the software development lifecycle (SDLC) and follow the waterfall model of the SDLC. Waterfall model defines the development process into a linear flow with a specified sequence to let the users understand that further level is made progressive on completion of the previous one. It involves a six-step process namely; requirements gathering and analysis, System design, implementation, testing, deployment of the system and maintenance.



Requirements gathering and analysis

Data collection and Datasets

We will use implicit data from secondary shopping transactions data.

We will source the data from secondary data repository in .csv format for processing on python. The data required will contain transactional data that can be analyzed to identify patterns that can help in developing the recommender. This data will for part of our dataset once its transcribed and stored in single columns per transaction.



Clean up the dataset

This will involve formatting the data also called encoding e.g. replacing word numbers with numeric and removing any unnecessary information that is not required in the process. Finally we will identify the data types we have and build data frames. We will use python inbuilt libraries and packages to clean up the data for easy consumption by the system.

Design

Generate Rules

We will use association rules since it is an unsupervised learning tool that is able to look for hidden patterns and therefore, there is limited need for data prep and feature engineering. It is a good start for certain cases of data exploration and can point the way for a deeper dive into the data using other approaches.

We will write our rules like; like this: {Milk} => {Bread}

This means that there is a great relationship between customers who buy milk and customers who buy bread in the same transaction. In this case Milk will be an antecedent and {Bread} will be a consequent.


Implementation

Apriori algorithm

Association rule mining finds interesting associations and/or correlation relationships among large set of data items. Association rules shows attribute value conditions that occur frequently together in a given dataset.

From the dataset that is available;

- we will generate the frequent patterns using Apriori algorithm.
- The association rules between products or items that interest a user can be derived from the patterns generated and these rules allow us to discover the products or items that frequently appear together.

- 
- List of all combination possible from group of products is generated.
 - Each combination represents a possible antecedent of an association rule.
For each combination, search the set of association rules for any rules that have a matching antecedent.
 - If a matching rule is found, take the products/items in its consequent and add them to the recommended list and the products can be recommended to the users even if he/she is a new user.



Clustering

Working with a very large dataset may likely generate very large amount of rules which may not be feasible for our experiments, therefore, we will then apply clustering to reduce the number of rules we have to consider and in turn increase efficiency on the recommendation. We will cluster our rules with the region the transactions were performed.

Testing and Evaluation

The prototype will be evaluated through offline experiments that do not require real users

We will evaluate the prototype by measuring the three metrics below;

Support is the relative frequency that the rules show up. We will mostly look for high support in order to make sure it is a useful relationship.

Confidence this will measure reliability of the rules.

Lift is the ratio of the observed support to that expected if the two rules were independent.

CHAPTER 4: SYSTEM ANALYSIS, DESIGN AND IMPLEMENTATION

SYSTEM ANALYSIS

- i. Data is collected from secondary sources mostly online repositories containing supermarket datasets
- ii. Dataset cleanup or processing to remove unnecessary information that is not required in the process is performed through inbuilt packages on python
- iii. Generate rules by use of association rules to identify hidden patterns in our data
- iv. Output or recommendations from the rules generated above to be able to inform any customer even the new ones without a history of purchases.

Data collection and preprocessing

- We obtained our data implicitly from an online repository.
- The dataset contains 541909 records with 25900 unique orders and approximately 4070 unique products and the files are in .csv format.
- We used Python packages to read, scan and temporarily store the data in data frames where analysis is executed on.
- The data is inconsistent and noisy and thus we applied python inbuilt packages to clean up and organize the data.

Dataset cleanup

The data available is inconsistent and noisy thus required clean up to make it less noisy. We will use python inbuilt libraries and packages to clean up the data for easy consumption by the system. The libraries that are required by python for data cleaning are pandas and numpy

Generate itemsets and association rules

A common strategy adopted by many association rule mining algorithms and which we adopted in our approach is to mold a problem into two major subtasks:

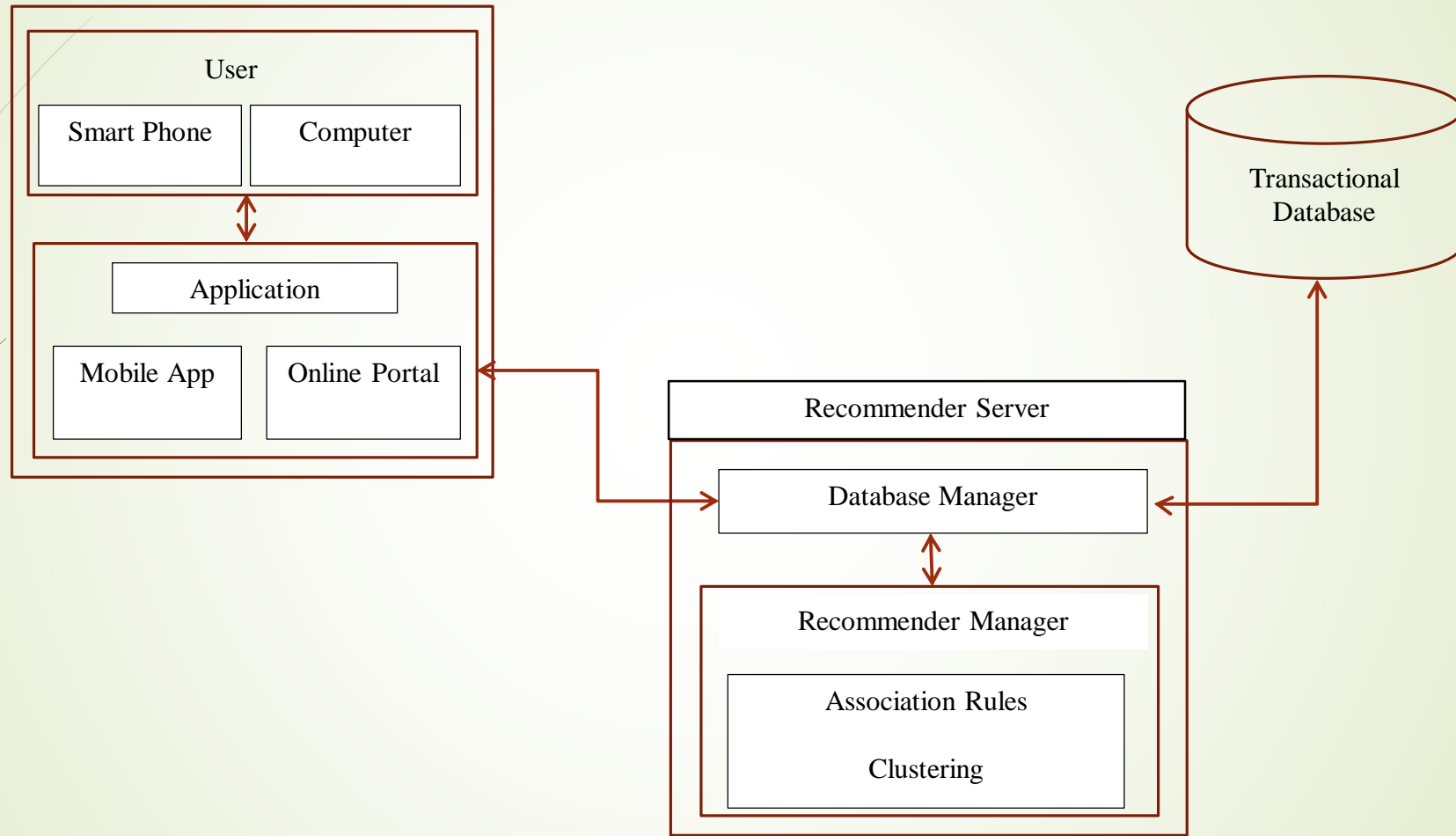
Generate Frequent Itemset, whose objective is to find all the itemsets that satisfy the minimum support threshold and they are called frequent itemsets.

Generation Rules that satisfy our thresholds, whose objective is to extract all the high-confidence rules from the frequent itemsets found in the previous step.

Generate itemsets and association rules

- From the dataset that is available, frequent patterns are generated using Apriori algorithm.
- The association rules between products or items that interest a user can be derived from the patterns generated and these rules allow us to discover the products or items that frequently appear together. List of all combination possible from group of products is generated.
- Each combination represents a possible antecedent of an association rule.
- For each combination, search the set of association rules for any rules that have a matching antecedent.
- If a matching rule is found, take the products/items in its consequent and add them to the recommended list and the products can be recommended to the users even if he/she is a new user.

Architectural Design of the recommender system



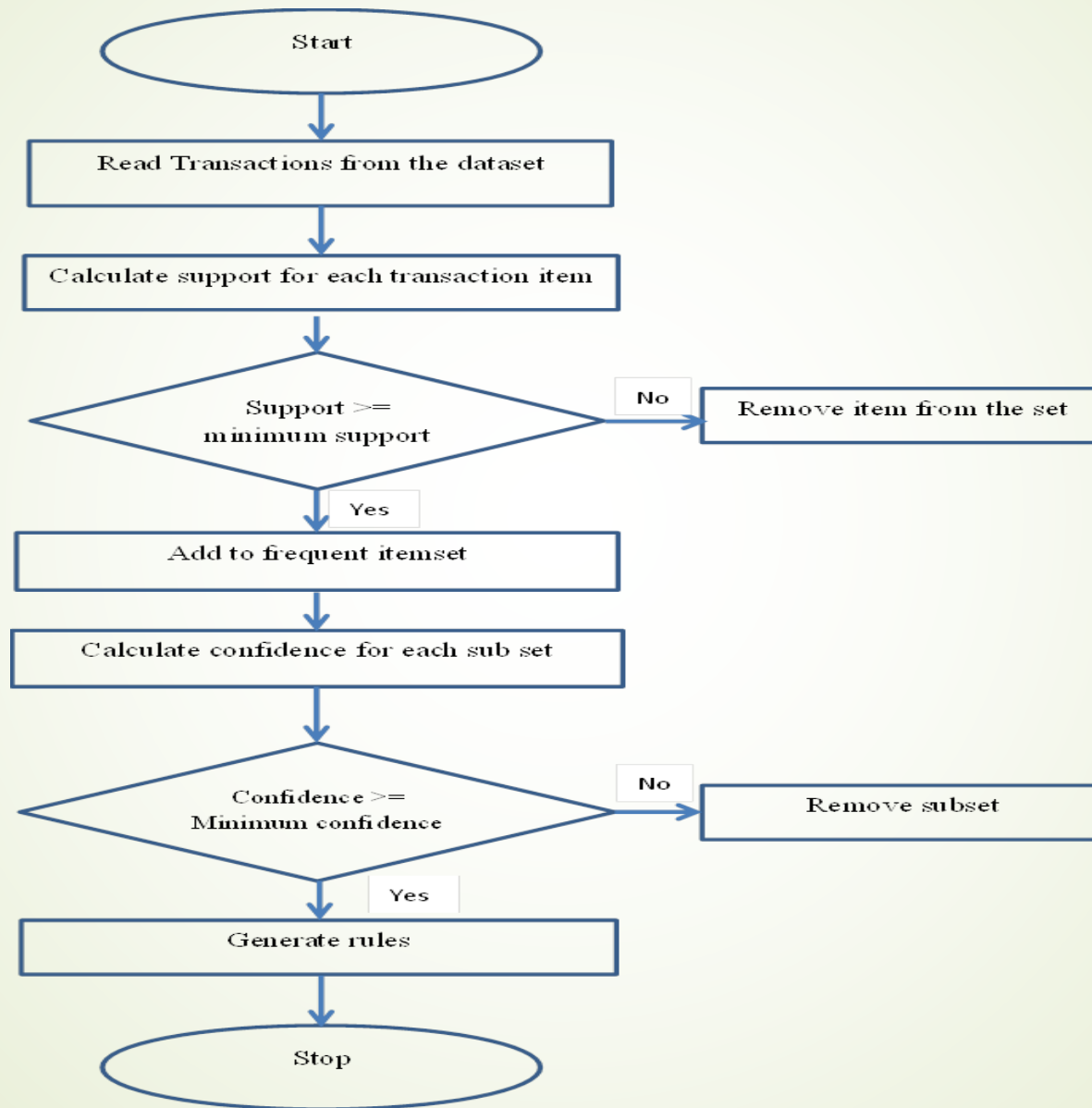
The recommender system explained

- The recommendation system is built on python where customer product choice is received for pattern matching and recommendation for next item generated
- It processes requests and generates a recommendation list
- The database manager handles database input/output processing and works with data such as customer data and product data.
- The recommendation manager works with recommendation algorithms that use association rules with apriori algorithm to identify patterns in in product purchases relationship

Application of association rule technique

- From the dataset that is available, we generate the frequent patterns using Apriori algorithm.
- List of all combination possible from group of products is generated.
- Each combination represents a possible antecedent of an association rule and For each combination we search the set of association rules for any rules that have a matching antecedent.
- If a matching rule is found, take the products/items in its consequent and add them to the recommended list
- the products can be recommended to the users even if he/she is a new user

System flow diagram



Implementation

The recommendation system is built on python.

We only implemented the recommender feature of the system which is the backend engine that generates rules and recommendations.

The system is broken into two processes;


- Relationship Patterns identification – this is where we generate rules and analyse the to get hidden patterns
- Clustering – to reduce the number of rules that we have to deal with by clustering results

This system draws recommendations as per product co-occurrence in historical purchases by other customers since the assumption is that the user is new and has no history on the system.

CHAPTER 5: RESULTS ANALYSIS AND EVALUATION

- We did a simulation with python 3 and a dataset containing transactional data from France, United Kingdom and Germany. Below are the results we achieve in terms of rules that we can be able to deduce meaning from.
- We used association rules to generate rules and then clustered the results further with region i.e. country where we picked France.
- For the transactions selected we have support in all of them above 0.06 and confidence above 60%. This means that the approach would be perfectly applicable for recommending items to both new users and ordinary users.

- In our experiment we do not consider customer historical profile but we are able to build a recommendations list from item co-occurrence having the users first choice of product as the antecedent.
- Support is an important measure in association rules because a rule that has very low support may occur simply by chance. A low support rule is also likely to be uninteresting from a business perspective because it may not be profitable to promote items that customers seldom buy.
- A common strategy adopted by many association rule mining algorithms and which we adopted in our approach is to mold a problem into two major subtasks:


- 
- I. Generate Frequent Itemset, whose objective is to find all the itemsets that satisfy the minimum support threshold and they are called frequent itemsets.
 - II. Generation Rules that satisfy our thresholds, whose objective is to extract all the high-confidence rules from the frequent itemsets found in the previous step.

Results visualization

Association rules table

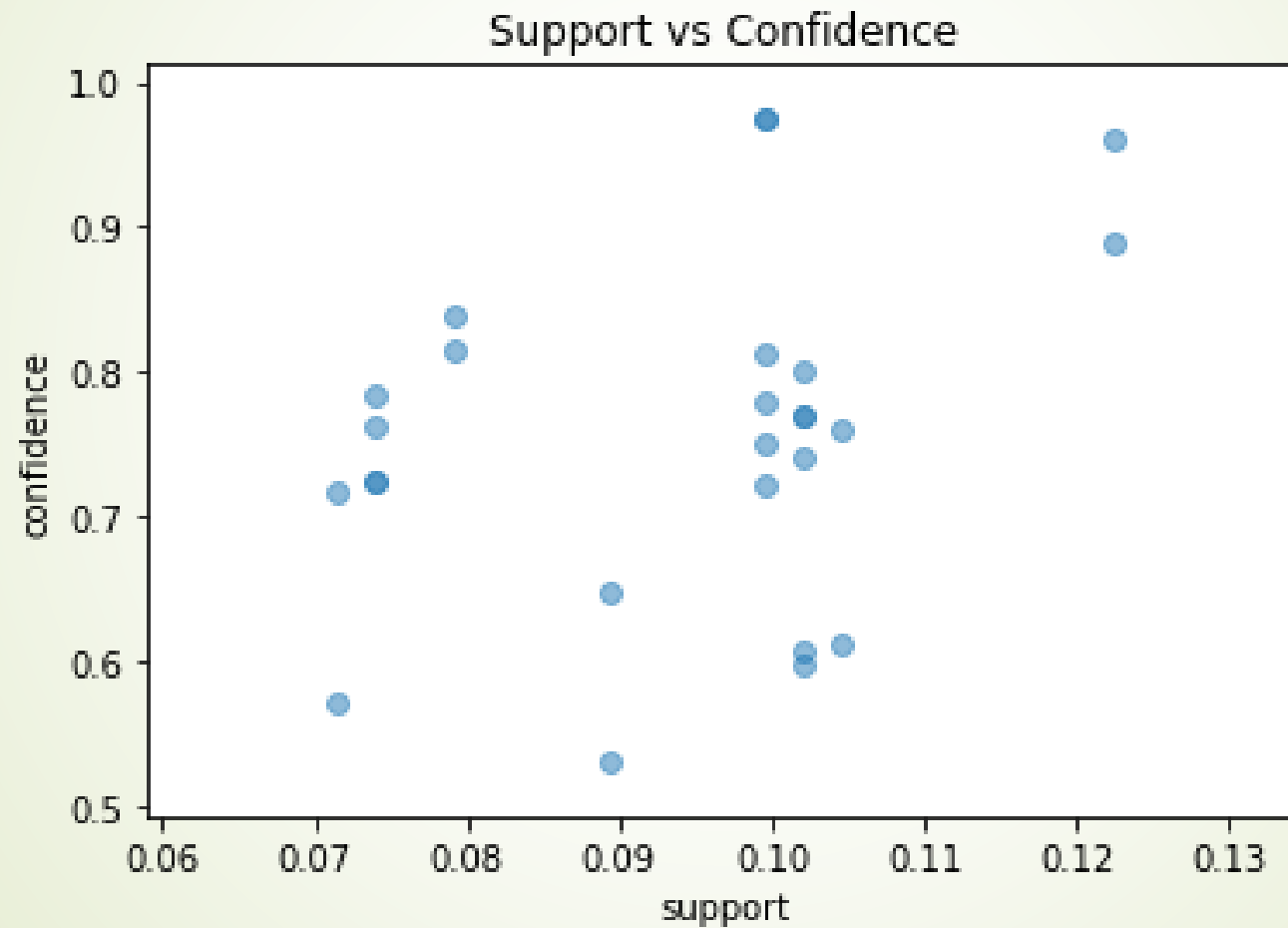
	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE PINK)	0.096939	0.102041	0.07398	0.763158	7.4789	0.064088	3.791383
1	(ALARM CLOCK BAKELIKE PINK)	(ALARM CLOCK BAKELIKE GREEN)	0.102041	0.096939	0.07398	0.725	7.4789	0.064088	3.283859
2	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.096939	0.094388	0.079082	0.815789	8.643	0.069932	4.916181
3	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.094388	0.096939	0.079082	0.837838	8.643	0.069932	5.568878
4	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE PINK)	0.094388	0.102041	0.07398	0.783784	7.6811	0.064348	4.153061

- The information given in the tables can be used to evaluate the association rule $\{\text{antecedent}\} \Rightarrow \{\text{consequent}\}$.
- At first glance, it may appear that people who pick an item on the LHS also tend to pick an item on the RHS because the rule's support is quite high and the rule's confidence values is also reasonably high.
- This conclusion can be misleading despite its high confidence value. The danger in the confidence measure is that it does not consider the support of the itemset in the rule consequent.
- Due to the limitations in the support-confidence framework, various objective measures like lift have been used to evaluate the quality of association patterns.

- 
- Lift measures the ratio between the rule's confidence and the support of the itemset in the rule consequent and is represented as below;
 - $\text{Lift}(X \rightarrow Y) = \text{confidence}(\{X, Y\}) / \text{support}(\{Y\})$
 - This shows how likely a consequent is likely to be purchased where and antecedent has been purchased.
 - Where a rule's lift is 1, this implies that there is no association between the items. Where the lift value is greater than 1 means that item consequent is likely to be bought if item antecedent is bought (positively correlated), while a value less than 1 means that item consequent is unlikely to be bought if item antecedent is bought (negatively correlated).
 - We set a minimum threshold on lift as 1 and on our experiment with rule that have a lift of at least 6, got 26 associations.

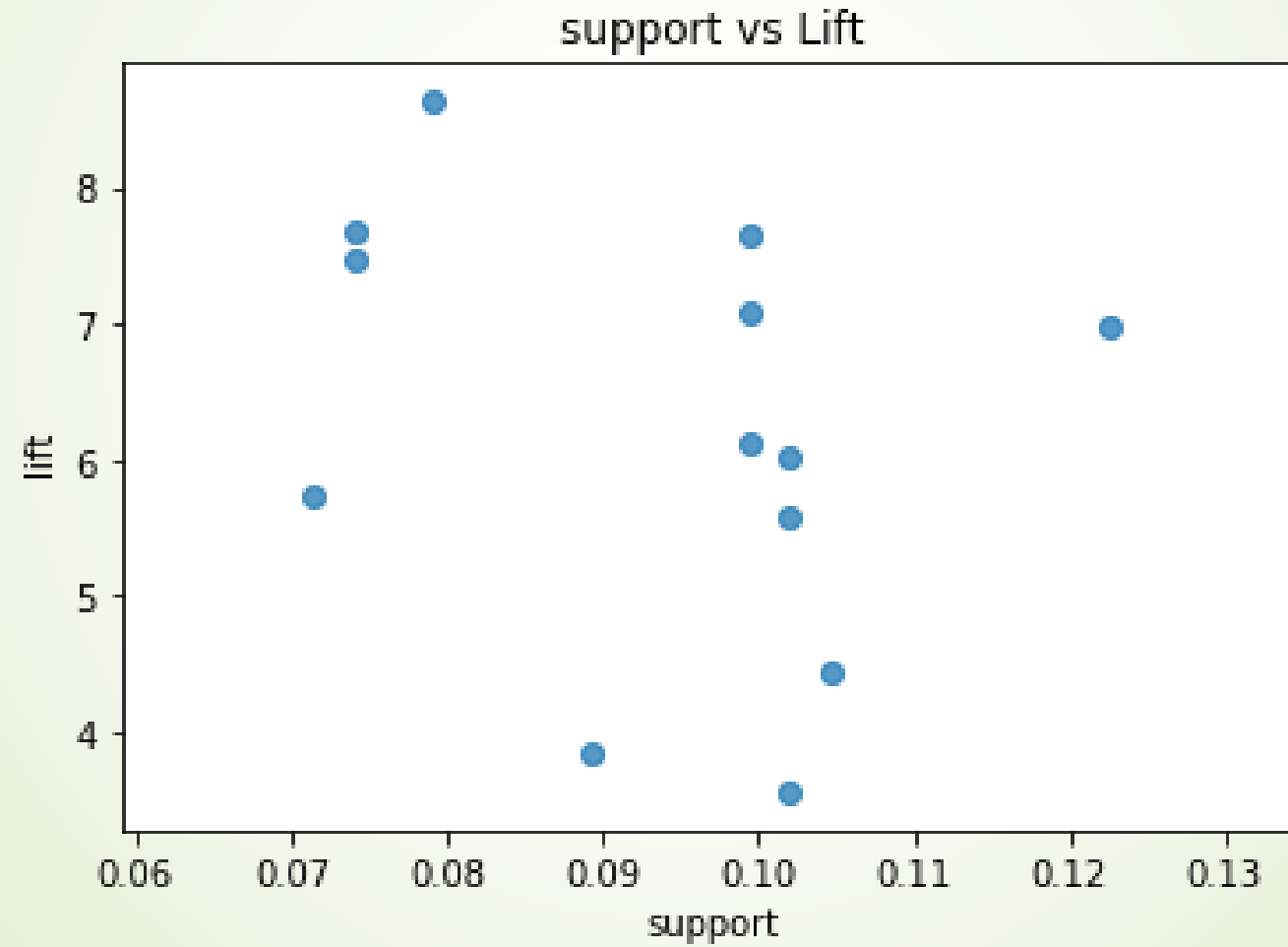
Results visualization

Graphs



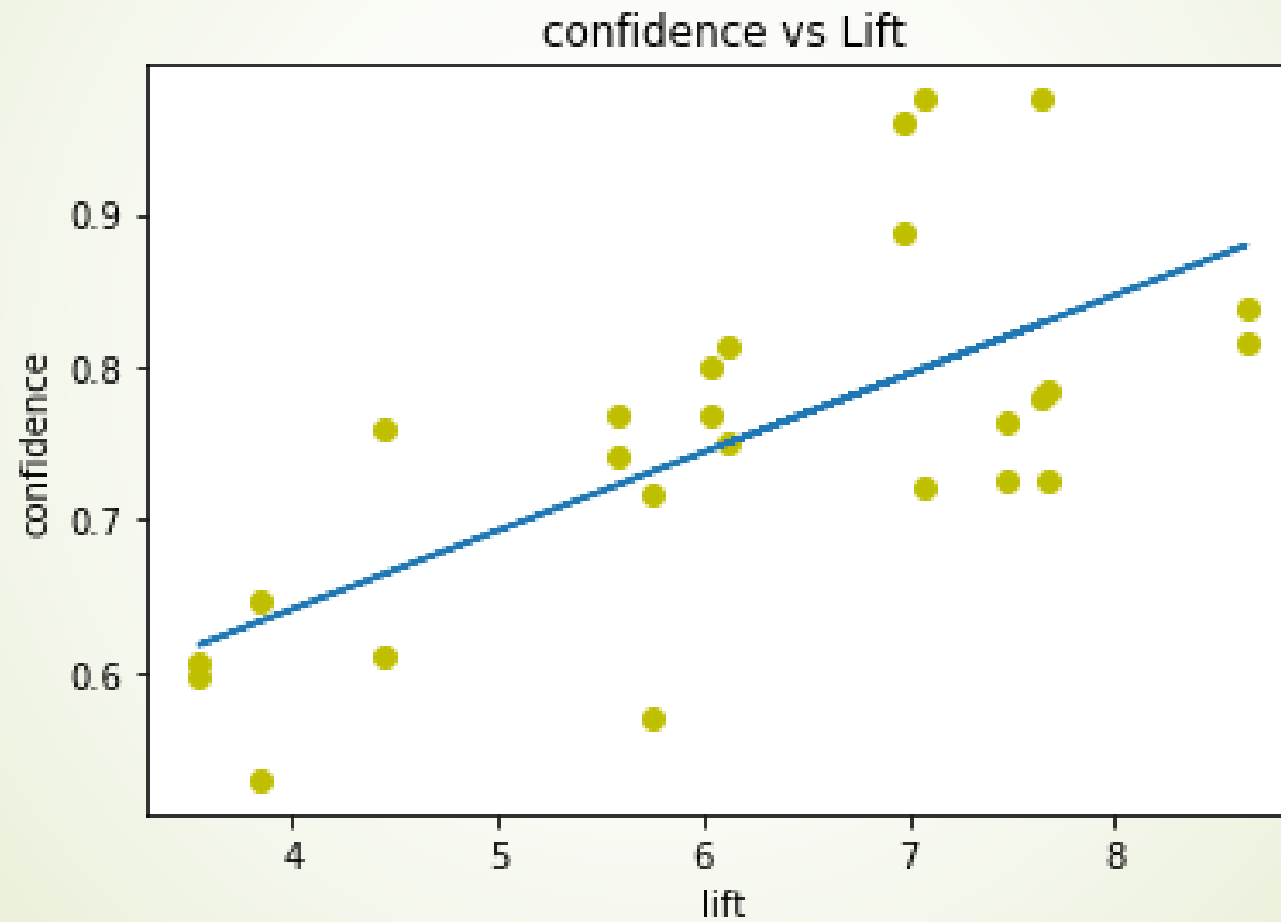
Results visualization

Graphs



Results visualization

Graphs



CHAPTER 6: CONCLUSION AND RECOMMENDATIONS

Conclusion

The basis of this research was to demonstrate how we can use AR in addressing problems that are associated with recommender systems specifically the cold start problem, in a time where there is a lot of growth in online shopping and e-commerce penetration.

We find that Association Rules is an important and effective concept of machine learning that is being used in market basket analysis.

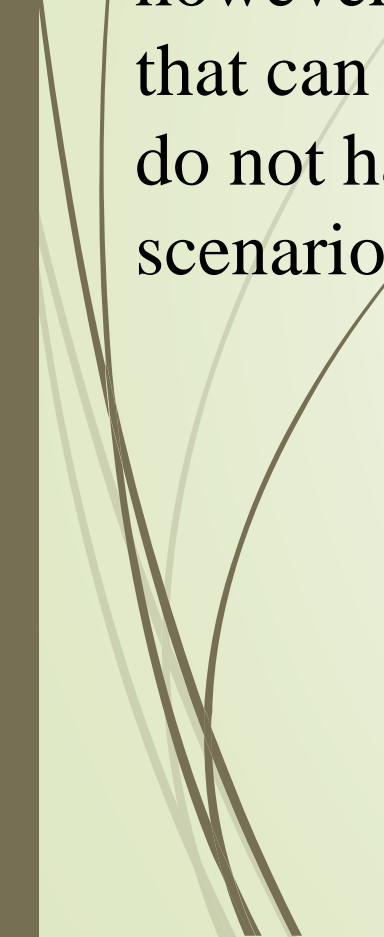
Identifying patterns and relationships in products helps in recommending to the customer what relevant items they might be interested in buying. Association rules help uncover all such relationships between items from huge databases.

Rules do not extract an individual's preference on specific products but rather find relationships between sets of products in distinct transactions and therefore providing relevant recommendations to both new users and existing ones. This aspect as per our approach on the proposal helps reduce chances of visitor cold start where a user is new to the system or no historical profiles exist about them and therefore, gives AR an advantage over CF which considers product ratings to recommend a product.



Recommendations

Association rules are a good approach in resolving visitor cold start problem, however, further research need to be done to find more evaluation techniques that can be used together with AR to be able to recommend new products that do not have associations but could be preferred by a user in a cold start scenario.



References

- Michael R. Wick and Paul J. Wagner, Department of Computer Science, University of Wisconsin-Eau Claire, “*Using Market Basket Analysis to Integrate and Motivate Topics in Discrete Structures*”
- Anna Gatzoura, Miquel Sanchez-Marre, Universitat Politècnica de Catalunya - BarcelonaTech, “*A Case-Based Recommendation Approach for Market Basket Data*”, Feb. 2015, pp. 20-27, vol. 30
- John Chege, “*A Web Content-Based Recommender System To Promote Automatic Discovery Of Learning Content For High School Students*”, November 2014.
- Joseph A. Konstan, “*Recommender systems: from algorithms to user experience*”, 10th March 2012.
- Heydary, M. and Yousefli, A. (2017), “*A new optimization model for market basket analysis with allocation considerations: A genetic algorithm solution approach*”, Management & Marketing. Challenges for the Knowledge Society, Vol. 12, No. 1, pp. 1- 11. DOI: 10.1515/mmcks-2017-0001.
- Sanjeevan Sivapalan, Alireza Sadeghian, Hossein Rahanam, “*Recommender Systems in E-Commerce*”, August 2014.

- Seren Sezen Karalök, Adnan Aktepe, Süleyman Ersöz, *“An Application in SPSS Clementine Based on the Comparison of Association Algorithms in Data Mining”*, September 2016.
- Marina Kholod, *“Market Basket Analysis of Convenience Store POS Data”*, November 2018.
- Roshan Gangurde, Dr. Binod Kumar, Dr. S. D. Gore, *“Building Prediction Model using Market Basket Analysis”*, February 2017.
- Sanjeev Mainali, *“Market Basket Analysis”*, August 2019.
- Wei J., He J., Chen K., Zhou Y., & Tang Z. (2017). *“Collaborative filtering and deep learning based recommendation system for cold start items. Expert Systems with Applications.”*
- Sadia Zeb , Dr. Irfan Ali Bhacho , Dr. Sheeraz Memon (2020) *“Addressing Cold Start Item Problem in Recommender System”*
- Hridya Sobhanam, A.K.Mariappan (2013), *“A Hybrid Approach to Solve Cold Start Problem in Recommender Systems using Association Rules and Clustering Technique”*
- Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong (2008), *“Addressing cold-start problem in recommendation systems. In Proceedings of the 2nd international conference on Ubiquitous information management and communication”*