Recommender System Using Market Basket Analysis

Nishant Srivastava, Stuti, Kanika Gupta

[1]Department of Computer Science & Engineering

[2]Jaypee Institute of Information Technology

ABSTRACT

Data Mining is a tool for retrieving novel and useful information contained in huge data stores. Conventional approaches to data mining techniques have majorly targeted the discovery of correlations among items that occur often in transactional databases. This method referred to as frequent item set mining believes that recurring item sets must be more significant to the user.

Alternatively, in this paper, we attempt to simulate an algorithm for a recent development called Utility Mining, which studies the usefulness or utility of item sets, in addition to their frequency. This High Utility Item Set Mining facilitates the recognition of item sets having utility value greater than a lower limit specified by the seller.

Our objective is to propose an ensemble algorithm which can tackle the issues of considering the purchase of an item as a binary variable and not its frequency in traditional ARM method and the shortcomings of HUIM method, wherein the Downward Closure property cannot be applied, in order to find a pruned and useful set of high utility items. The high utility items obtained through the proposed algorithm will be then deployed in an e-commerce website to accurately predict the next product bought by an online customer using a simple recommender system and compare its result with the prevalent method i.e. A-priori of association rule mining.

INTRODUCTION

Market basket analysis is a technique for finding client obtaining designs by extracting rules or co-occurrences from stores' value-based databases. Finding, for instance, that general store clients are probably going to buy dairy products like milk , bread, and cheddar together can help administrators in planning store format, sites, item blend and packaging, and other promoting techniques.

To date, the A-priori algorithm is quite a prevalent method for mining the association rules from transaction logs, which fulfill the minimum support and confidence levels as per the users' requirement. However, the practical utility of frequent item set mining is confined by the importance of the identified item sets.

An emerging area is that of Utility Mining which not only considers the frequency of the item sets but also considers the utility associated with the item sets by taking metrics like profit and sales into consideration.

Recommendation Engines can be defined as information filtering system aiming to predict the products the user might like and buy. A typical list of recommendations on the basis of the user's interest in a product is produced through a variety of methods and using association rules to generate recommendations is one such method.

PROPOSED WORK

1.Utility Mining Algorithm

TID: Transaction ID
F: frequency of each item in a transaction
C: cost of each item
E: user specified threshold
Ē: utility threshold
HU: High Utility Item Set
HTWU: High Transaction Weighted Utilisation Item Set

Input: Transactional database, Utility (cost) of each item
Output: Association Rules

```
1.∀TID
      calculate transactional
      utility using F X C
2.∀X ∊ itemset
      calculate transaction
      weighted utilisation
      (twu)
3.If twu(X)>E
      X ⊆ HTWU
4.If E = Ē → HU ⊆ HTWU
```

2.Recommendation Engine Algorithm

Q: Search Query
X: Product
AR: Association Rules

Input: Search Query (Q)
Output: Recommendation using Association Rules

```
1.If Q=true
      Result=∀P ∊ Product
      Data
2.∀Result
      If X ∊ Result is selected
        match(X, A->B) such
      that R1=A
        ∀X∊ Association Rules
      such that X is the
      antecedent
            retrieve AR(X)
3.∀AR(X)
      If support(AR(X))=max
      OR utility(AR(X))=max
        recommend the
      consequents of A->B
      where A=X i.e. B
```

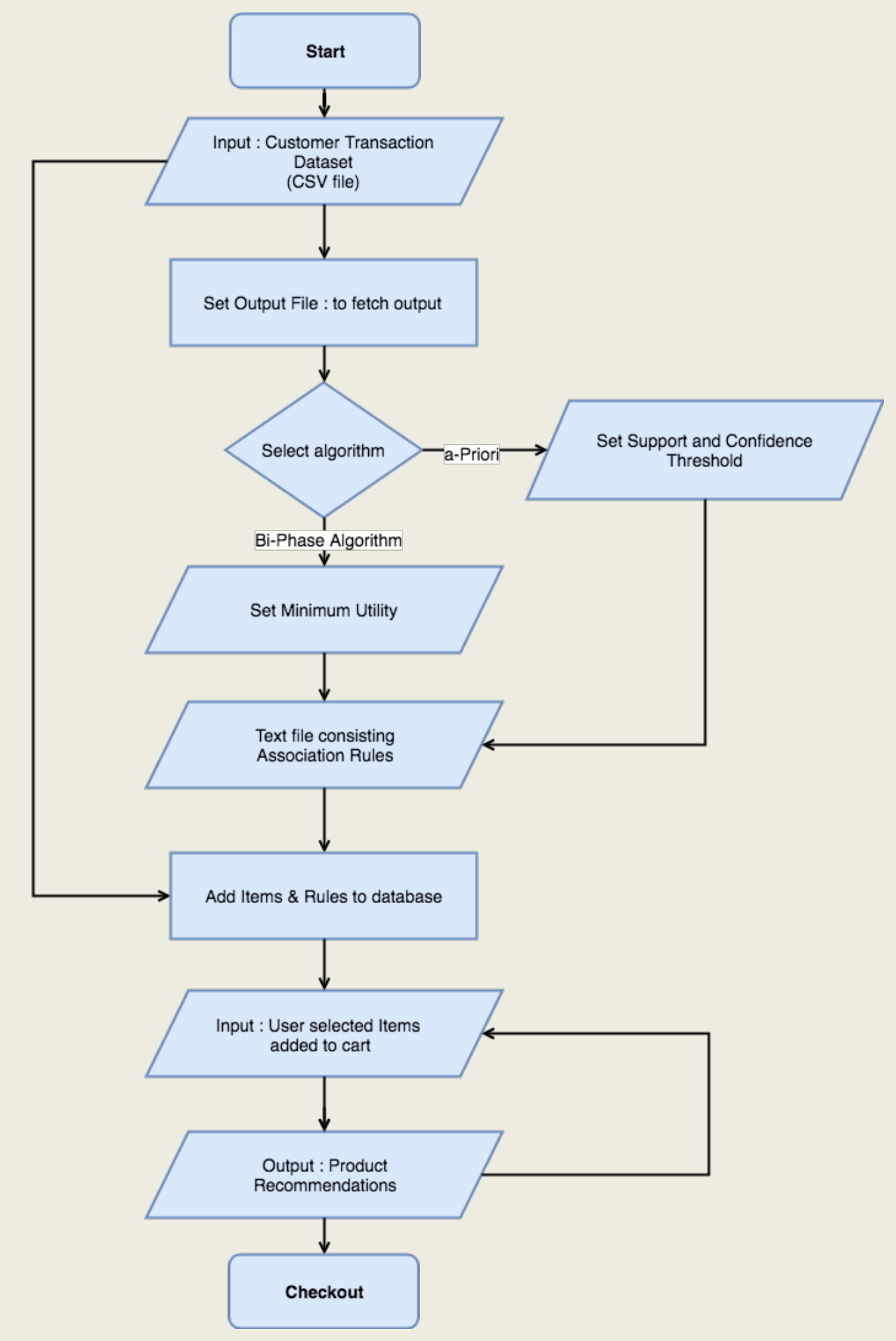

Start
Input : Customer Transaction Dataset (CSV file)
Set Output File : to fetch output
Select algorithm
a-Priori
Set Support and Confidence Threshold
Bi-Phase Algorithm
Set Minimum Utility
Text file consisting Association Rules
Add Items & Rules to database
Input : User selected Items added to cart
Output : Product Recommendations
Checkout

Figure 1. Control Flow Diagram

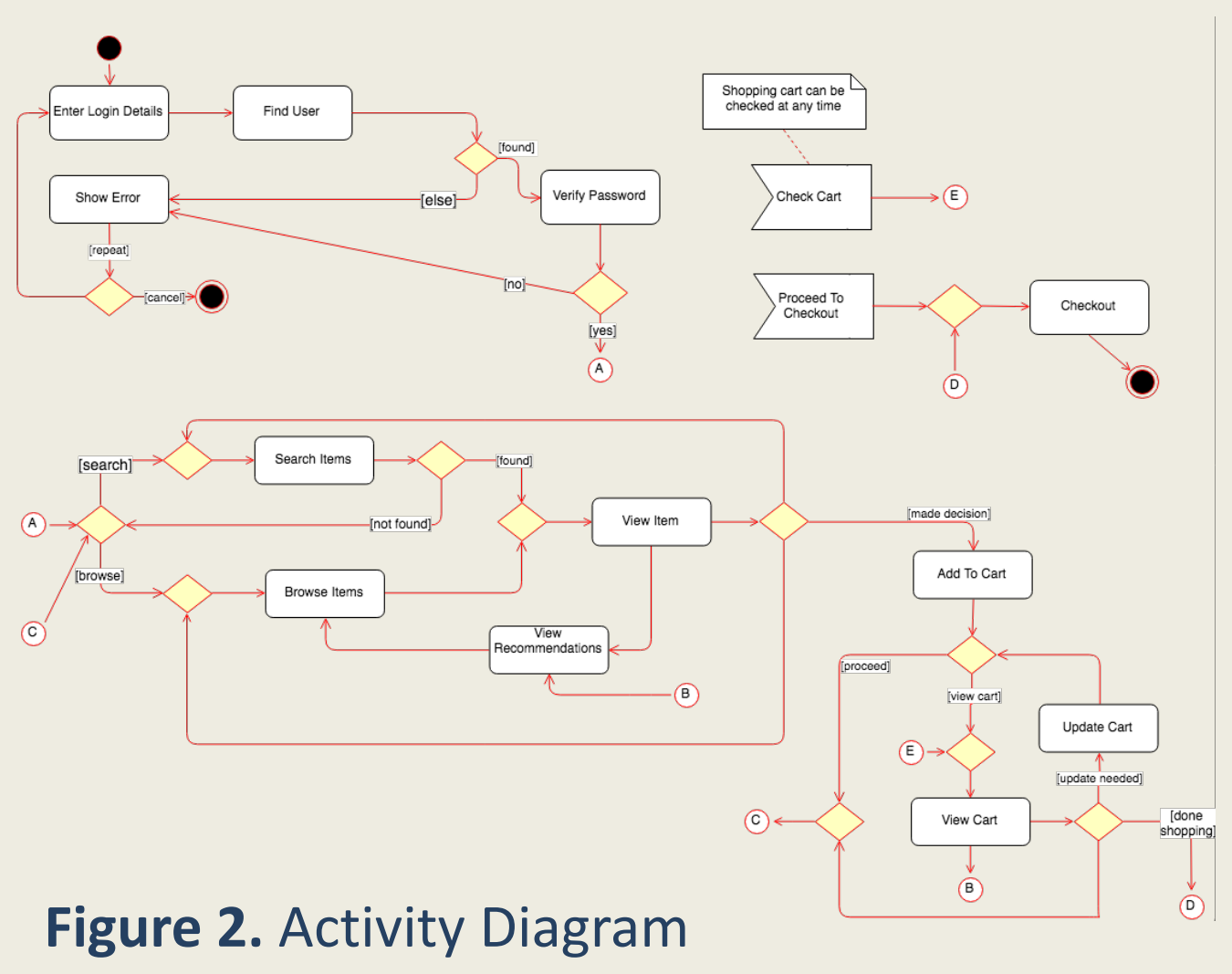


Figure 2. Activity Diagram

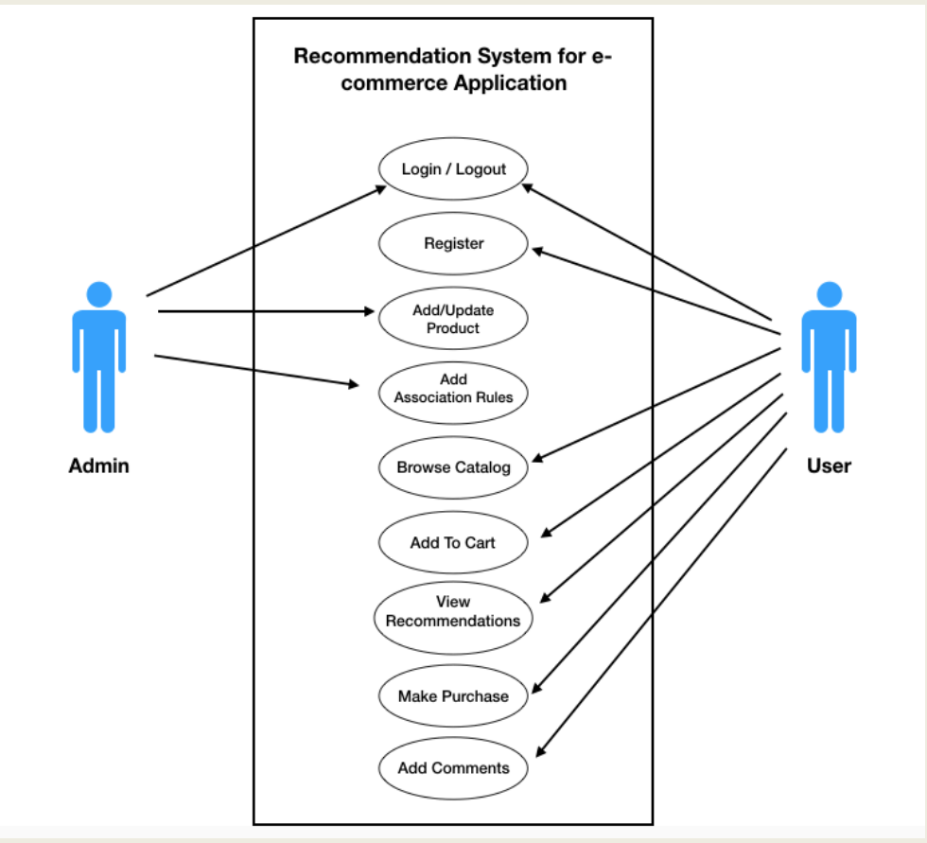
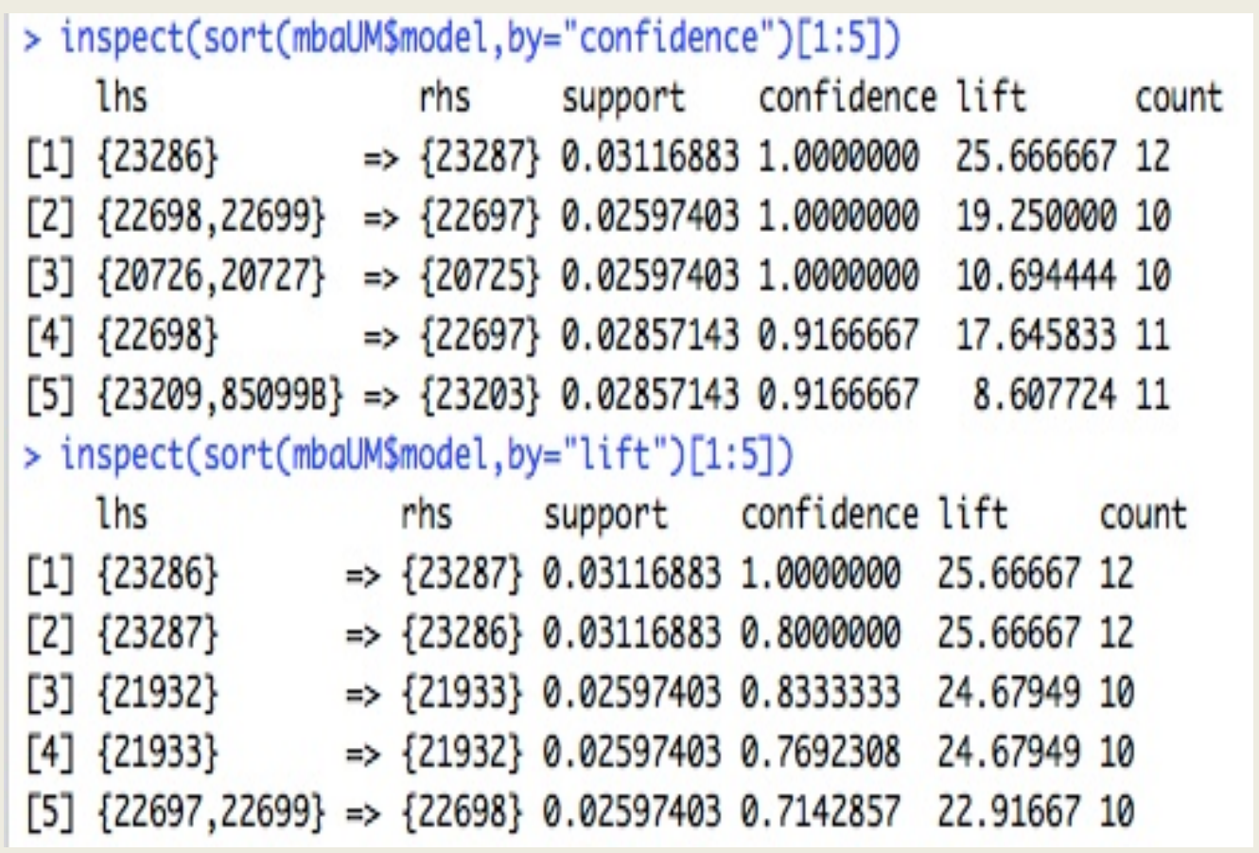


Figure 3. Use Case Diagram

FINDINGS

We have used three measures – Support, Confidence and Lift to rank the mined association rules on the basis of their importance. A lift greater than 1 signifies that the antecedents and consequents are positively correlated.



```
> inspect(sort(mbaUM$model,by="confidence")[1:5])
    lhs                rhs     support    confidence lift      count
[1] {23286}         => {23287} 0.03116883 1.0000000  25.666667 12
[2] {22698,22699}   => {22697} 0.02597403 1.0000000  19.250000 10
[3] {20726,20727}   => {20725} 0.02597403 1.0000000  10.694444 10
[4] {22698}         => {22697} 0.02857143 0.9166667  17.645833 11
[5] {23209,85099B} => {23203} 0.02857143 0.9166667   8.607724 11
> inspect(sort(mbaUM$model,by="lift")[1:5])
    lhs               rhs     support    confidence lift     count
[1] {23286}        => {23287} 0.03116883 1.0000000  25.66667 12
[2] {23287}        => {23286} 0.03116883 0.8000000  25.66667 12
[3] {21932}        => {21933} 0.02597403 0.8333333  24.67949 10
[4] {21933}        => {21932} 0.02597403 0.7692308  24.67949 10
[5] {22697,22699} => {22698} 0.02597403 0.7142857  22.91667 10
```

Figure 4. Lift, Support & Confidence Measure of Rules

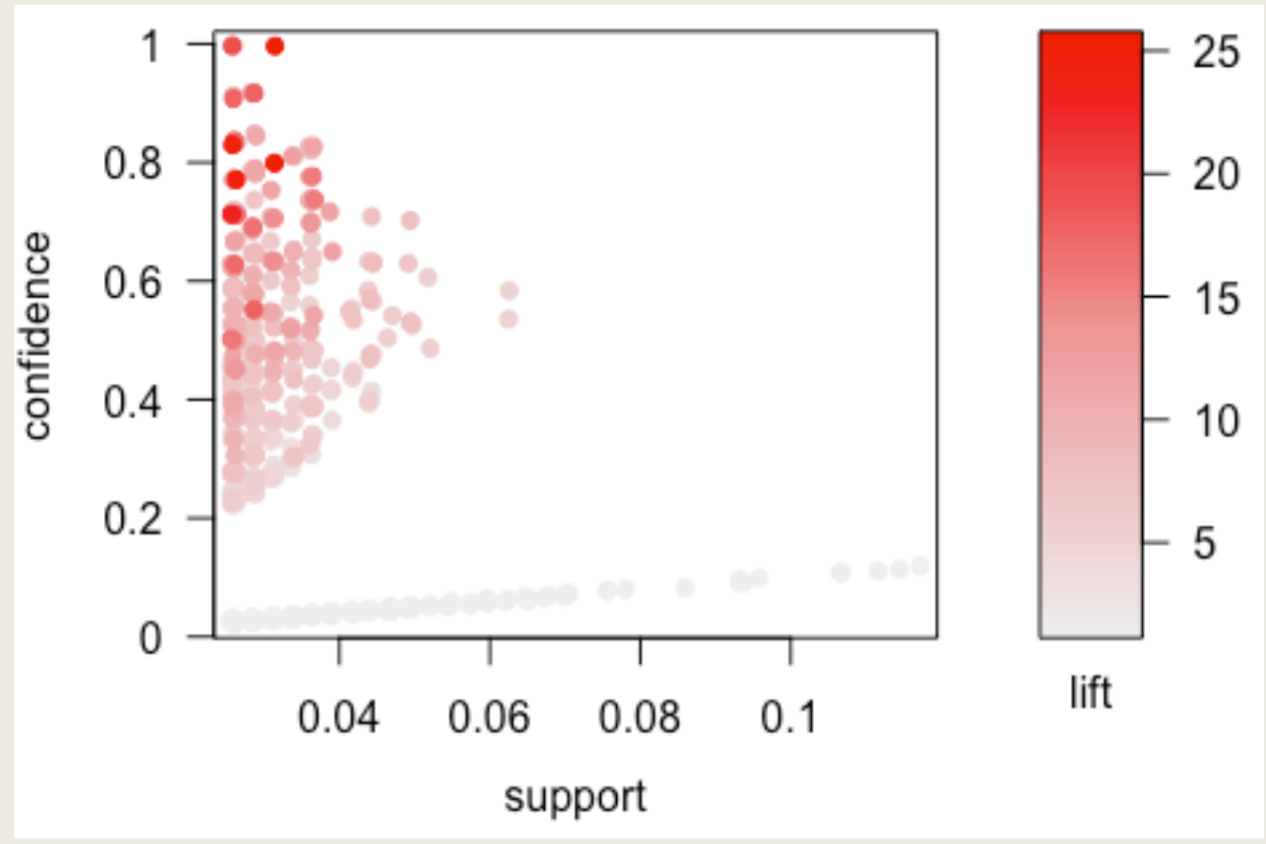


Figure 5. Scatter Plot b/w Lift, Support & Confidence

In order to compare the efficiency of A-priori and the proposed ensemble algorithm we performed multiple test cases on the test set of 10 each wherein we checked for the precision, recall and the F-score of all the test cases on both the algorithms.

Test Case ID	Precision	Recall	F-score
1	45%	78%	0.57
2	60%	82%	0.69
3	52%	74%	0.61
4	58%	76%	0.66

Table 1. Performance Measures of A-Priori.

Test Case ID	Precision	Recall	F-score
1	61%	100%	0.76
2	52%	95%	0.67
3	70%	85%	0.77
4	61%	98%	0.75

Table 2. Performance Measures of Utility Based.

Comparison Measures	A-Priori	Utility Based
Precision	54%	**61%***
Recall	78%	**94%***
F-Score	0.63	**0.74***

Table 3. Average Performance Measures.

It was found that for the algorithm proposed by us, the measure of precision was **7 percentage points** more than the traditional algorithm. Similarly, it was found that the utility based algorithm exceeded the A-priori algorithm by the measure of **16 percentage points** and the F-score for traditional algorithm is less than the Utility based algorithm by **11 percentage points**.

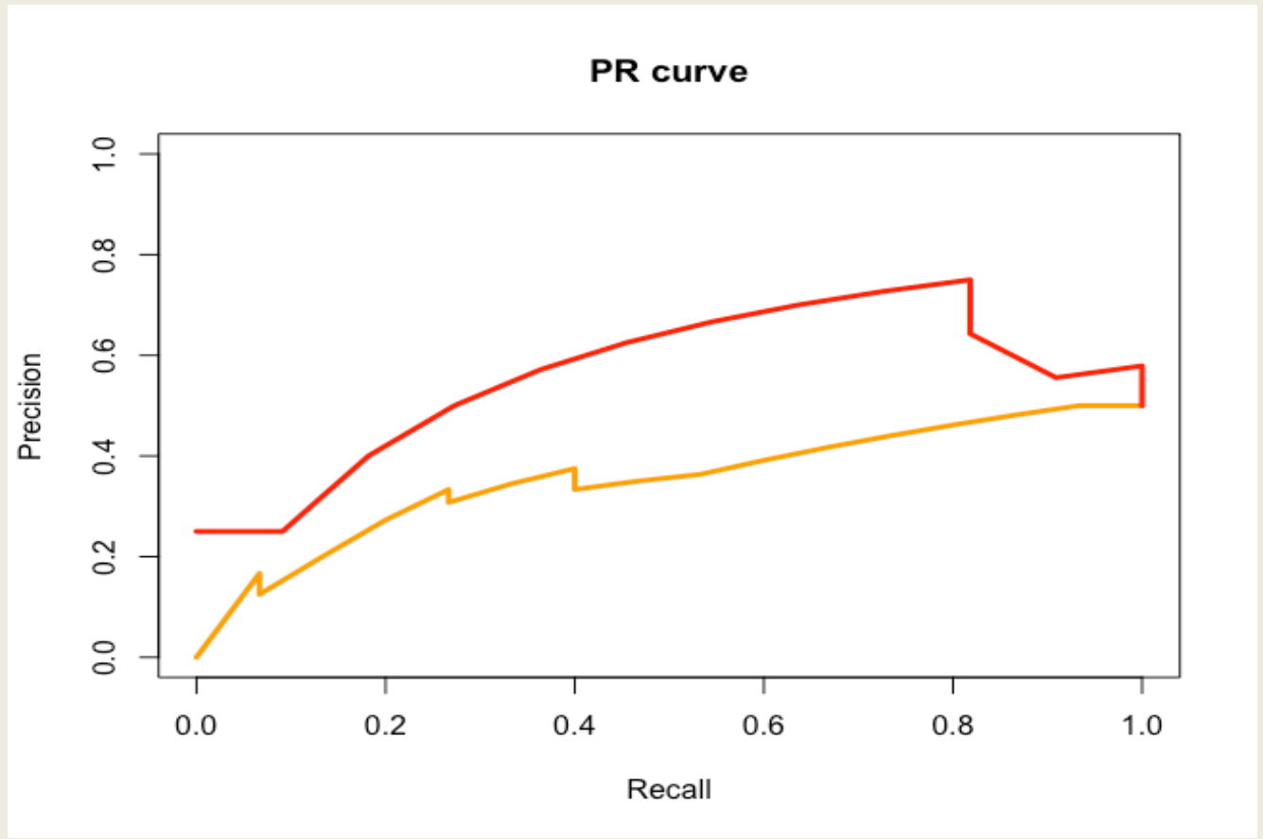


Figure 6. Overlapping PR Curve.
*Orange - A-priori, Red - Utility Based

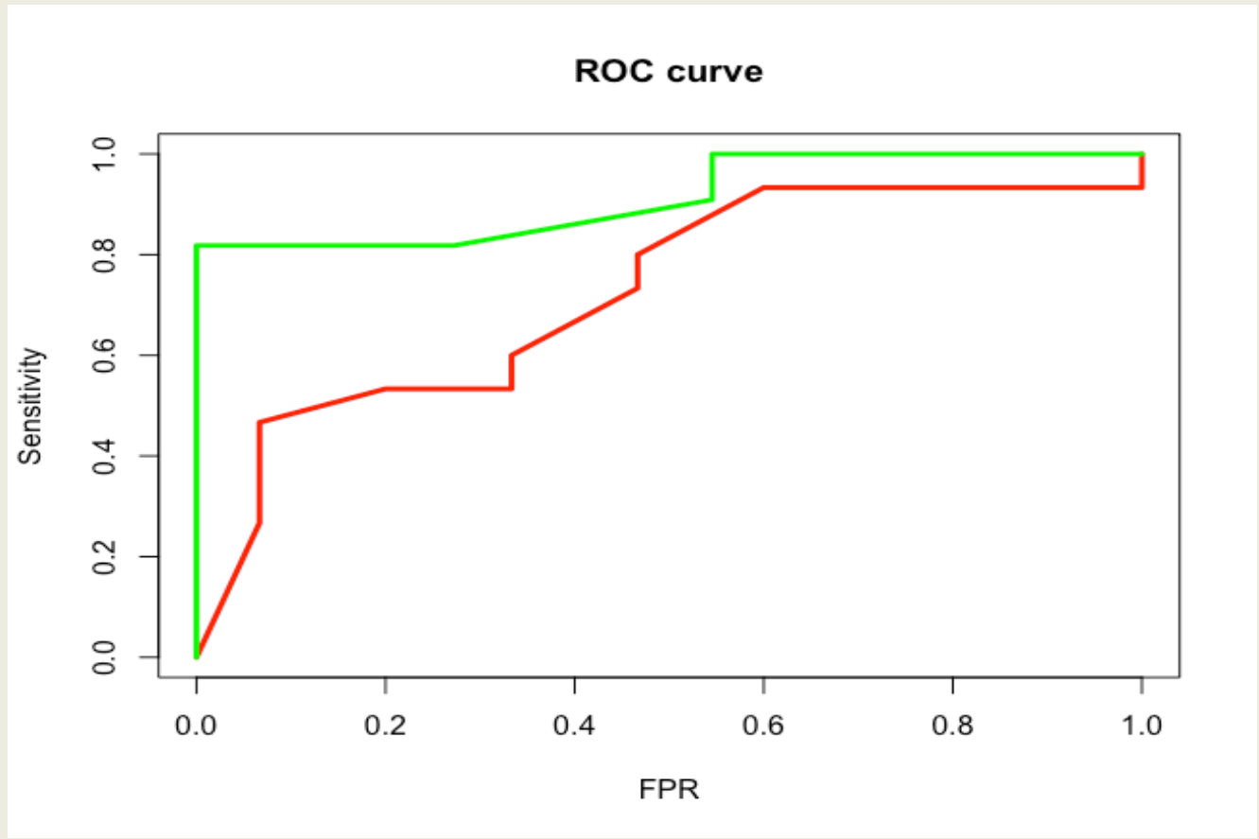


Figure 7. Overlapping ROC Curve.
*Red - A-priori, Green - Utility Based

The higher AUC for PR curve and ROC curve for utility based rules signifies a greater accuracy of the recommender system as opposed to association rule based system.

CONCLUSIONS

The proposed Bi-Phase algorithm that recognizes high utility item sets more effectively and the recommender system modeled on the rules generated by them outperforms the classical association rule based recommender system on all performance measures.
However, a very high threshold value in filtering could let some valuable results get lost and give versa. Also high complexity of algorithms with huge data sets to be processed caused large running time.

REFERENCES

1. Agrawal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB* (Vol. 1215, pp. 487-499).
2. Cakir, O., & Aras, M. E. (2012). A recommendation engine by using association rules. Procedia-Social and Behavioral Sciences, 62, 452-456.