Open access Journal International journal of Emerging Trends in Science and Technology

# The Application of Data Mining In Online Bookstore

Authors

**Mr. Anil Vasoya*, Ankita Jain**, Sarika Patel***, Vrunda Desai******

*Department of IT Engineering Thakur College of Engineering and Technology
Email: anilk.*vasoya@thakureducation.org*
**Department of IT Engineering Thakur College of Engineering and Technology
Email: *ankita.1992@gmail.com*
***Department of IT Engineering Thakur College of Engineering and Technology
Email: *sarikapatel164@gmail.com*
****Department of IT Engineering, Thakur College of Engineering and Technology
Email: *vrunda.tcet@gmail.com*

*Abstract:*
*With the rapid development of Internet technology in recent years, Electronic Commerce has been an inevitable product of the economy, the science and the technology. This paper takes an online bookstore platform as a background, introduces the definition, functions, process and common analytical techniques of data mining. In the end, the experiment on association rules mining from the order data of online bookstore is completed by Improved Apriori algorithm.*
*Keywords:*
*Online bookstore; Data mining; Association rule; Improved Apriori Algorithm*

## INTRODUCTION

Along with the development of the computer and the database technology, all trades and professions start to adopt the computer and corresponding information technology to carry on the management and the operation. It has greatly improved the ability of producing, collecting, storing and processing data. Under such a background, people urgently need new computing technology and tools to excavate the knowledge contained in database, which guides technological decision and strengthens the competitiveness of enterprises.

Data mining has emerged as a means for identifying patterns and trends from a large amount of data [3]. The major reason that data mining has attracted a great deal of attention in the information industry in recent years is due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. Data mining can be viewed as a result of the natural evolution of information technology.

As computer's applications are wide spread, the accumulation of data is becoming larger and larger.

We can extract useful information by data mining techniques. The web site can process their business information deeply based on original information system so as to

strengthen their competitive advantage and increase their turnover.

The data mining of online shopping can help to predict sales, marketing, select the right time of participation in product sales promotion such an seckill, group purchase, which will useful for making profit for the shop.

**PROBLEM DEFINITION**

The basic apriori algorithm requires multiple passes over the database. For disk resident database, this requires reading the database completely for each pass resulting in a large number of disk I/OS. In these algorithms, the effort spent in performing just the I/O may be considerable for large databases. Apart from poor response times, this approach also places a huge burden on the I/O subsystem adversely affecting other users of the system. The problem can even be worse in a client-server environment.

To overcome the above problems the proposed system focuses on the method to improve the performance of finding the association rules in the transaction databases. The software gains an acceptable result when runs over a quite large databases.

The proposed system considers supermarket database and applies improved apriori algorithm.

The proposed system consists of the following steps:

1. To evaluate the importance of finding association rules and specifies the main cost of the process finding them.
2. To present, illustrate and analyze the strength and weakness of some algorithms using partitioning approach.
3. To build up a system to manage a small soft, find interesting rules related to customer routines. This system uses improved apriori algorithm that provides good efficiency.

**DATA MINING PROCESS**

Building a mining model is part of a larger process that includes everything from asking questions about the data and creating a model to answer those questions, to deploying the model into a working environment.

This process can be defined by using the following six basic steps.

- Defining the Problem
- Preparing Data
- Exploring Data
- Building Models
- Exploring and Validating Models
- Deploying and Updating Models

The following Figure 1 describes the relationships between each step in the process, and the technologies in Microsoft SQL Server R2 that can be used to complete each step.

Although the process illustrated in the diagram is circular, each step does not necessarily lead directly to the next step. Creating a data mining model is a dynamic and iterative process. After exploring the data, one may find that the data is insufficient to create the appropriate mining models, and that one therefore have to look for more data. Alternatively, you may build several models and then realize that the models do not adequately answer the problem you defined, and that you therefore must redefine the problem. You may have to update the models after they have been deployed because more data has become available. Each step in the process might need to be repeated many times in order to create a good model.

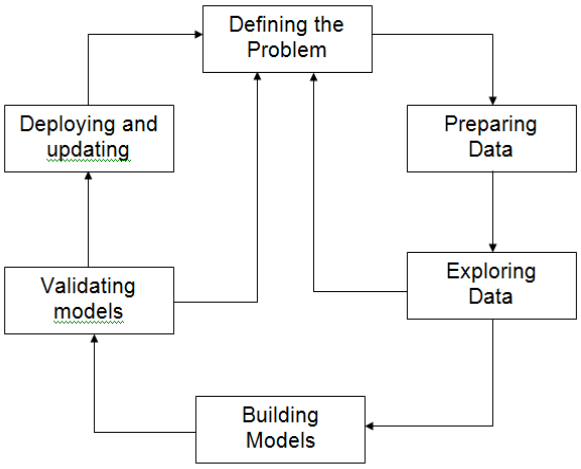


**Figure 1:** Relationships between six steps

**DESIGN AND IMPLEMENT OF ONLINE BOOKSTORE BASED ON ASP.NET**

An online bookstore is formed by the front and back business subsystem. The website is built by MS Visual Studio 2010 , ASP.net, Microsoft SQL Server R2, etc. Visitors can log in the online bookstore and choose books to buy. The whole shopping flowchart is as the following Figure 2.

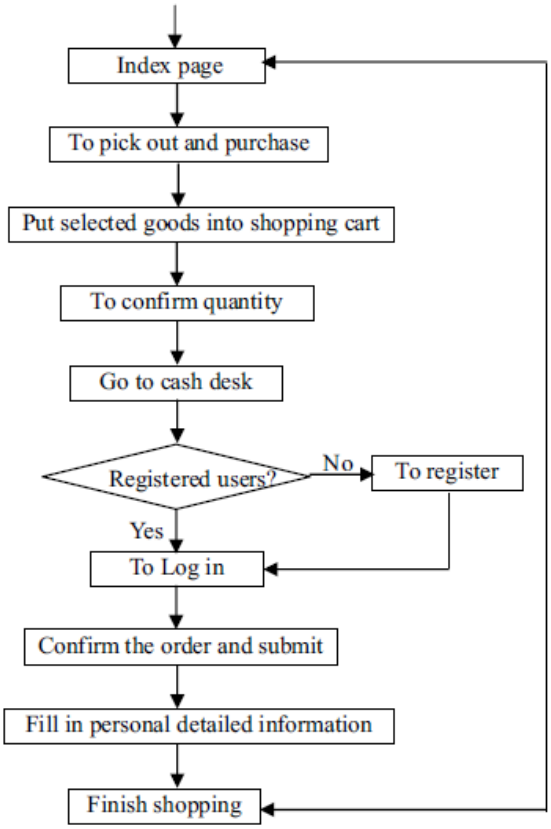


**Figure 2:** System overall design flowchart

## 4. The application of data mining in online bookstore

As the running of online bookstore, it will produce a large number of order lists which contain customer's shopping information on the net. It is very important to find valuable information or knowledge from all lists.

## APRIORI ALGORITHM

Apriori algorithm is easy to execute and very simple, is used to mine all frequent itemsets in database. The algorithm [2] makes many searches in database to find frequent itemsets where k-itemsets are used to generate k+1-itemsets. Each k-itemset must be greater than or equal to minimum support threshold to be frequency. Otherwise, it is called candidate itemsets. In the first step, the algorithm scan database to find frequency of 1-itemsets that contains only one item by counting each item in database. The frequency of 1-itemsets is used to find the itemsets in 2-itemsets which in turn is used to find 3-itemsets and so on until there are not any more k-itemsets. If an itemset is not frequent, any large subset from it is also non-frequent [1]; this condition prune from search space in database.

### 5.1 Limitations of Apriori Algorithm

Apriori algorithm suffers from some weakness in spite of being clear and simple. The main limitation is costly wasting of time to hold a vast number of candidate sets with much frequent itemsets, low minimum support or large itemsets. For example, if there are 104 from frequent 1-itemsets, it need to generate more than 107 candidates into 2-length which in turn they will be tested and accumulate [2]. Furthermore, to detect frequent pattern in size 100 (e.g.) v1, v2… v100, it have to generate 2100 candidate itemsets [1] that yield on costly and wasting of time of candidate generation.

So, it will check for many sets from candidate itemsets, also it will scan database many times repeatedly for finding candidate itemsets. Apriori will be very low and inefficiency when memory capacity is limited with large number of transactions. In this paper, we propose approach to reduce the time spent for searching in database transactions for frequent itemsets.

## THE IMPROVED ALGORITHM OF APRIORI

This section will address the improved Apriori ideas, the improved Apriori, an example of the improved Apriori, the analysis and evaluation of the improved Apriori and the experiments.

### 6.1 The improved Apriori ideas

In the process of Apriori, the following definitions are needed:

Definition 1: Suppose T={T1, T2, … , Tm},(m≥1) is a set of transactions, Ti= {I1, I2, … , In},(n≥1) is the set of items, and k-itemset = {i1, i2, … , ik},(k≥1) is also the set of k items, and k-itemset ⊆ I.

Definition 2: Suppose σ (itemset), is the support count of itemset or the frequency of occurrence of an itemset in transactions.

Definition 3: Suppose Ck is the candidate itemset of size k, and Lk is the frequent itemset of size k.

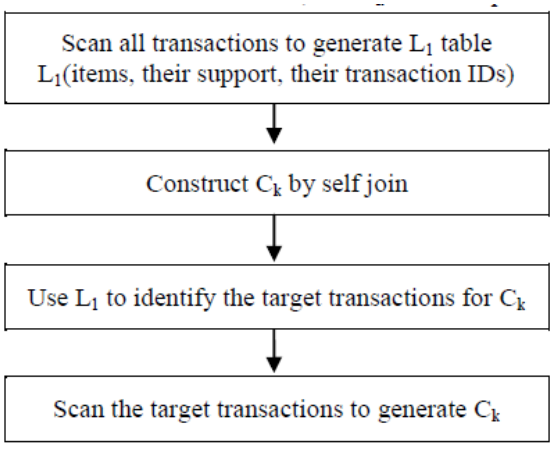


**Figure 3: Steps for Ck generation**

In our proposed approach, we enhance the Apriori algorithm to reduce the time consuming for candidates itemset generation. We firstly scan all transactions to generate L1 which contains the items, their support count and Transaction ID where the items are found. And then we use L1 later as a helper to generate L2, L3 ... Lk. When we want to generate C2, we make a self join L1 * L1 to construct 2-itemset C (x, y), where x and y are the items of C2. Before scanning all transaction records to count the support count of each candidate, use L1 to get the transaction IDs of the minimum support count between x and y, and thus scan for C2 only in these specific transactions. The same thing for C3, construct 3-itemset C (x, y, z), where x, y and z are the items of C3 and use L1 to get the transaction IDs of the minimum support count between x, y and z, then scan for C3 only in these specific transactions and repeat these steps until no new frequent itemsets are identified. The whole process is shown in the Figure 3.

### 6.2 The improved Apriori

The improvement of algorithm can be described as follows:

//Generate items, items support, their transaction ID

(1) L1 = find_frequent_1_itemsets (T);

(2) For (k = 2; Lk-1 ≠Φ; k++) {

//Generate the Ck from the LK-1

(3) Ck = candidates generated from Lk-1;

//get the item Iw with minimum support in Ck using L1, (1≤w≤k).

(4) x = Get _item_min_sup(Ck, L1);

// get the target transaction IDs that contain item x.

(5) Tgt = get_Transaction_ID(x);

(6) For each transaction t in Tgt Do

(7) Increment the count of all items in Ck that are found in Tgt;

(8) Lk= items in Ck ≥ min_support;

(9) End;

(10) }

### 6.3 An example of the improved Apriori

Suppose we have transaction set D has 9 transactions, and the minimum support = 3. The transaction set is shown in Table.1.

| T_ID | Items |
|---|---|
| $T_1$ | $I_1, I_2, I_5$ |
| $T_2$ | $I_2, I_4$ |
| $T_3$ | $I_2, I_4$ |
| $T_4$ | $I_1, I_2, I_4$ |
| $T_5$ | $I_1, I_3$ |
| $T_6$ | $I_2, I_3$ |
| $T_7$ | $I_1, I_3$ |
| $T_8$ | $I_1, I_2, I_3, I_5$ |
| $T_9$ | $I_1, I_2, I_3$ |

**TABLE 1**: THE TRANSACTIONS

| Items | support | |
|---|---|---|
| $I_1$ | 6 | |
| $I_2$ | 7 | |
| $I_3$ | 5 | |
| $I_4$ | 3 | |
| $I_5$ | 2 | deleted |

**TABLE 2**: THE CANDIDATE 1-ITEMSET

firstly, scan all transactions to get frequent 1-itemset l1 which contains the items and their support count and the transactions ids that contain these items, and then eliminate the candidates that are infrequent or their support are less than the min_sup. The frequent 1-itemset is shown in table 3.

| Items | support | T_IDs | |
|---|---|---|---|
| $I_1$ | 6 | $T_1, T_4, T_5, T_7, T_8, T_9$ | |
| $I_2$ | 7 | $T_1, T_2, T_3, T_4, T_6, T_8, T_9$ | |
| $I_3$ | 5 | $T_5, T_6, T_7, T_8, T_9$ | |
| $I_4$ | 3 | $T_2, T_3, T_4$ | |
| $I_5$ | 2 | $T_1, T_8$ | deleted |

**TABLE 3**: FREQUENT 1_ITEMSET

The next step is to generate candidate 2-itemset from L1. To get support count for every itemset, split each itemset in 2-itemset into two elements then use l1 table to determine the transactions where you can find the itemset in, rather than searching for them in all transactions. for example, let's take the first item in table.4 (I1, I2), in the original apriori we scan all 9 transactions to find the item (I1, I2); but in our proposed improved algorithm we will split the item (I1, I2) into I1 and I2 and get the minimum support between them using L1, here i1 has the smallest minimum support. after that we search for itemset (I1, I2) only in the transactions T1, T4, T5, T7, T8 and T9.

| Items | support | Min | Found in | |
|---|---|---|---|---|
| $I_1, I_2$ | 4 | $I_1$ | $T_1, T_4, T_5, T_7, T_8, T_9$ | |
| $I_1, I_3$ | 4 | $I_3$ | $T_5, T_6, T_7, T_8, T_9$ | |
| $I_1, I_4$ | 1 | $I_4$ | $T_2, T_3, T_4$ | deleted |
| $I_2, I_3$ | 3 | $I_3$ | $T_5, T_6, T_7, T_8, T_9$ | |
| $I_2, I_4$ | 3 | $I_4$ | $T_2, T_3, T_4$ | |
| $I_3, I_4$ | 0 | $I_4$ | $T_2, T_3, T_4$ | deleted |

**TABLE 4**: FREQUENT 2_ITEMSET

The same thing to generate 3-itemset depending on L1 table, as it is shown in table 5.

| Items | support | Min | Found in | |
|---|---|---|---|---|
| I1, I2 , I3 | 2 | $I_3$ | $T_5, T_6, T_7, T_8, T_9$ | deleted |
| I1, I2 , I4 | 1 | $I_4$ | $T_2, T_3, T_4$ | deleted |
| I1, I3 , I4 | 0 | $I_4$ | $T_2, T_3, T_4$ | deleted |
| I2, I3 , I4 | 0 | $I_4$ | $T_2, T_3, T_4$ | deleted |

**TABLE 5**: FREQUENT 3-ITEMSET

For a given frequent itemset LK, find all non-empty subsets that satisfy the minimum confidence, and then generate all candidate association rules.

in the previous example, if we count the number of scanned transactions to get (1, 2, 3)-itemset using the original apriori and our improved apriori, we will observe the obvious difference between number of

scanned transactions with our improved apriori and the original apriori. from the table 6, number of transactions in1-itemset is the same in both of sides, and whenever the k of k-itemset increase, the gap between our improved apriori and the original apriori increase from view of time consumed, and hence this will reduce the time consumed to generate candidate support count.

| | Original Apriori | Our improved Apriori |
|---|---|---|
| 1-itemset | 45 | 45 |
| 2-itemset | 54 | 25 |
| 3-itemset | 36 | 14 |
| sum | 135 | 84 |

**TABLE 6**: NUMBER OF TRANSACTIONS SCANNED EXPERIMENTS

We developed an implementation for original Apriori and our improved Apriori, and we collect 5 different groups of transactions as the follow:

- T1: 555 transactions.
- T2: 900 transactions.
- T3: 1230 transactions.
- T4: 2360 transactions.
- T5: 3000 transactions.

The first experiment compares the time consumed of original Apriori, and our improved algorithm by applying the five groups of transactions in the implementation. The result is shown in Figure 4.

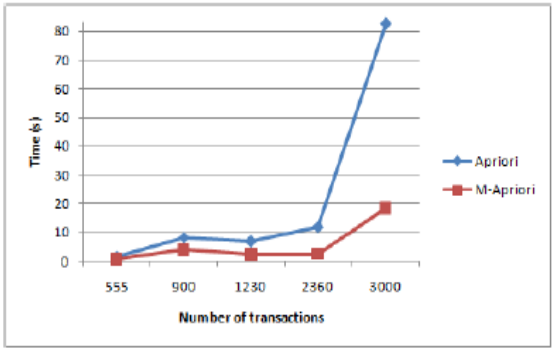


**Figure** 4: Time consuming comparison for different groups of transactions

The second experiment compares the time consumed of original Apriori, and our proposed algorithm by applying the one group of transactions through various values for minimum support in the implementation. The result is shown in Figure 5.

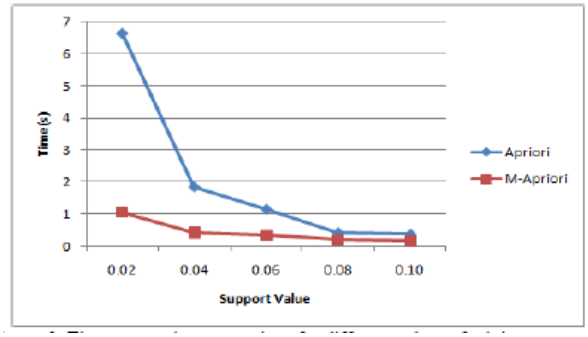


**Figure 5**: Time consuming comparison for different values of minimum support

## RESULT

Our system being time efficient by the use of the improved Apriori algorithm provides better services to the customer, thus making customer retention for longer period of time.

This system gives regular analyzes on the basis of the sales of the product in form of bar graphs and pie charts so that the admin can prepare quick and reliable strategies for better profits.

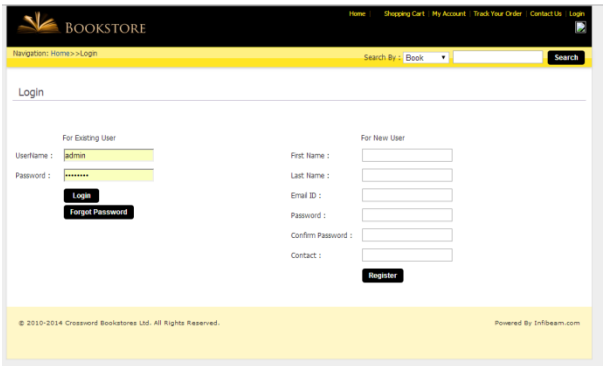


**Figure 6:** Login Page

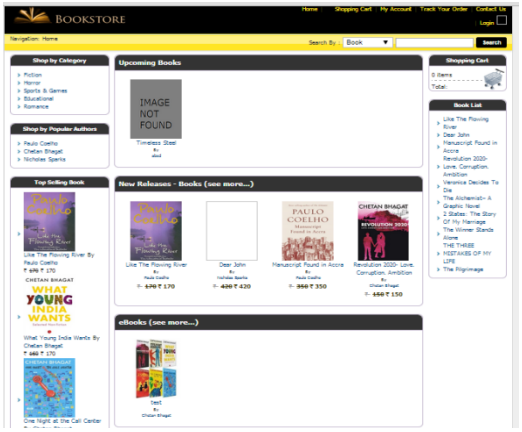


**Figure 7:** Home page

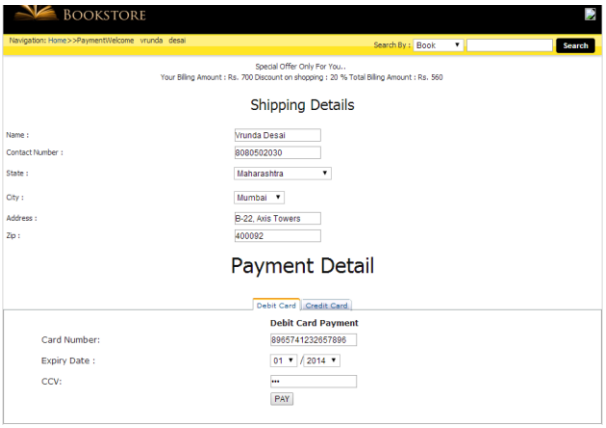


**Figure 10:** Payment Page

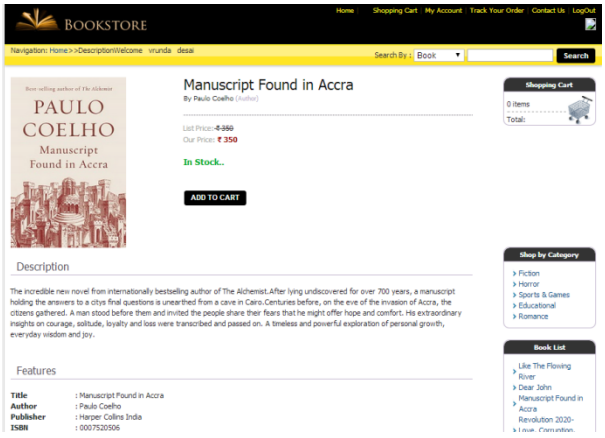


Figure 8. Book Description page

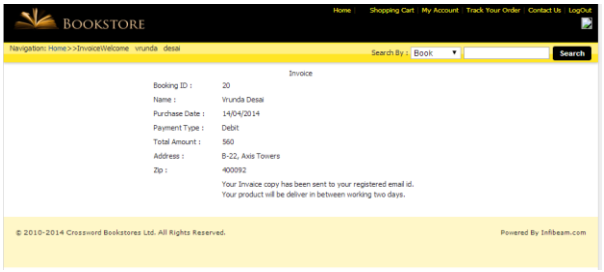


**Figure 11:** Invoice Generation Page

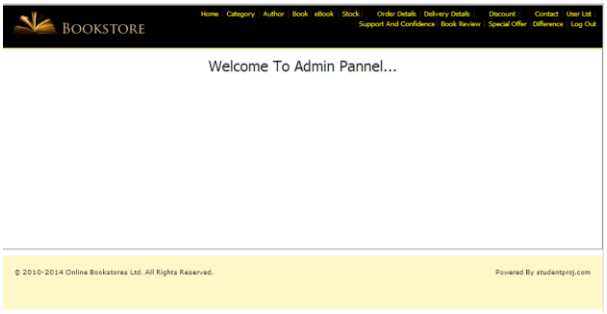


**Figure 12:** Admin Side Home Page

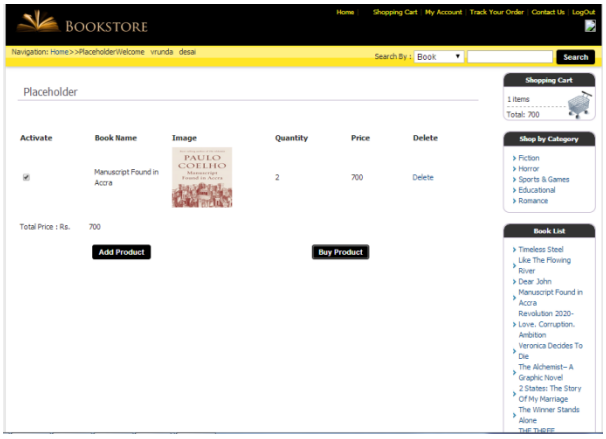


**Figure 9:** Shopping Cart

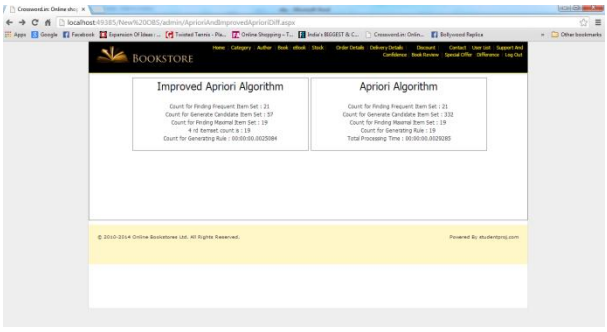


**Figure 13:** Difference in time used by apriori algorithm and improved algorithm(Admin side)

## CONCLUSIONS

In this paper, an online bookstore was created based on Asp.net. Then an experiment analyzes customer buying habits by finding associations between the order items in database. The discovery of such associations can help
retailers develop marketing strategies by gaining insight
into which items are frequently purchased together by increased sales by helping retailers do selective marketing and plan the bookshop space.

**Acknowledgments**

We would like to express my deep gratitude to DR. B.K Mishra, Principal, Thakur College of Engineering and Technology, Mumbai for extending the opportunity for major project and providing all the necessary resources for this purpose.

We express heartfelt thanks to Mr. Anil Vasoya for his wonderful support for preparing the project and for giving us an opportunity to do our project on "The Application Of Data Mining In Online Bookstore".

We are grateful to Mr.Vinayak Bharadi, Head of Department (Information Technology), Thakur College of Engineering and Technology, Mumbai and all the faculty members for conducting the project and their encouragement and co operation has been a source of great inspiration

Also we would like to thank Thakur College of Engineering and Technology for providing me all the facilities for timely completion of the project.

## REFERENCES

[1] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, Dec. 2007.
[2] S. Rao, R. Gupta, "Implementing Improved algorithm Over APRIORI Data Mining Association Rule Algorithm", *International Journal of Computer Science And Technology*, pp. 489-493, Mar. 2012
[3] J. Han, and M. Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann Publishers,
pp.1-35, 2001.
[4] Guo Hongli,Li Juntao, "The application of mining association rules in online shopping", 2011 Fourth International Symposium on Computational Intelligence and Design.

## Authors Profile

**Mr. Anil Vasoya** is presently working as Assistant Professor in InformationTechnology Department, Thakur College of Engineering & Technology, Mumbai. He has completed his B.E. in Information Technology from Atharva College of Engineering (Mumbai University) and M.E in Computer Engineering from Thakur College of Engineering & Technology.

**Ms. Ankita Jain** is currently pursuing B.E in Information Technology from Thakur College of Engineering & Technology, Mumbai.

**Ms. Sarika Patel** is currently pursuing B.E in Information Technology from Thakur College of Engineering & Technology, Mumbai.

**Ms. Vrunda Desai** is currently pursuing B.E in Information Technology from Thakur College of Engineering & Technology, Mumbai.