National College of Ireland

Anticipating Customer Behavior With Association Rule Techniques Using Apriori Algorithm

Analytical CRM
Msc. Data Analytics

Rohan Dongare
Student ID: X18120199

School of Computing
National College of Ireland

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Rohan Dongare

Student ID: X18120199

Programme: Msc. Data Analytics **Year:** 2018-2019

Module: Analytical CRM

Supervisor: Prof. Vikas Sahni

Submission Due Date: 03/04/2019

Project Title: Anticipating Customer Behaviour using Association Rule mining techniques with Apriori Algorithm

Word Count: 3786 **Page Count** 09

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: 

Date: 03/04/2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Anticipating Customer Behavior With Association Rule Techniques Using Apriori Algorithm

Rohan Vijay Dongare
Analytical CRM
Msc. Data Analytics
X18120199

Abstract— **As storing data has gotten a lot cheaper and easier over the years due to the advances in IT technologies, industries of all sizes are now able to effectively obtain and store huge amount of data. Using the right data mining techniques, the information from these databases can be used in forming sales, operation, marketing and service strategies for any organization. A big obstacle in most organizations is finding profitable hidden patterns in customer purchases which would enable them to achieve a competitive advantage. One such algorithm which is widely used for this task of finding hidden customer purchase patterns from frequent item sets is the Apriori algorithm. This paper will illustrate the use of association rule mining to discover customer purchase patterns by extracting association rules from a customer transactions dataset using the apriori algorithm. In addition, various correlations in the purchases of the customers with respect to other features of the dataset such as the gender, age group, marital status, location of the customers will be analyzed using charts.**

Keywords—Association rule mining, Apriori algorithm, hidden purchase patterns, Correlations.

I. INTRODUCTION

Most organizations are investing heavily in gathering consumer data but once this stage is completed, they struggle with effectively utilizing this information to gain useful insight from it. Various statistical and machine learning methods are now being utilized to achieve these insights. Association rule mining (popularly known as market basket analysis) is one of these techniques which is used for analyzing purchasing patterns of customers by extracting rules representing co-occurrences of products from a transactional database. These rules describe what customers are more likely to buy once they have already purchased a product. This information can then be leveraged in various ways such as designing layouts, bundling products together, formulating new marketing strategies etc.

Using this technique on the existing customer transactions, cross selling of products can be carried out by recommending existing customers to new products which they are statistically more likely to buy. In a retail setting, products that are often purchased together can also be closely placed or sold in pairs. Sometimes if certain products are most likely to be bought together, they can be placed at distance from one another so that the customer is also introduced to other products on his way to get the more likely product.

As these association rules easy to understand this technique has many business applications and can be used across various sectors such as Retail, Marketing, Finance, Telecommunication, Insurance etc.

In this paper association rule mining using apriori algorithm will be carried out using R programming language. A visual analysis of correlations between customer purchases and other features such as age, gender, marital status and location will also be carried out.

The organization of the paper is as follows: The background of the dataset is described in section II while section III states the guiding hypothesis for the project. Section IV states the business value of the project. Section V contains the literature review, whereas Section VI discusses about the methodology used. Section has the implementation and interpretation while Section VIII and IX have conclusion and references respectively.

II. BACKGROUND OF THE DATSET

The data for this project was acquired from an ongoing competition on Analytics Vidya[1]. While the goal of the competition is to predict the purchase value, this project is focused on performing market basket analysis of the products sold. The dataset provided is of an anonymous company with 12 variables and 550,000 records. The dataset consists of user id which is a uniquely assigned value to every user, product id which is uniquely assigned to every product, customer's gender which is a binary variable with M indicating male and F indicating female. The age of the customer is given in 7 different age brackets (0-17, 18-25, 26-35, 36-45, 46-50, 51-55, 55+) with one age bracket assigned to every customer. A variable stay

[1] https://datahack.analyticsvidhya.com/contest/black-friday/#data_dictionary

indicates the number of years the person has been residing in the city which can take either one of the 5 values depending on the number of years the customer has been residing in the city (0, 1, 2, 3, 4+). A variable marital status indicates if the customer is married (0 representing no and 1 yes). There are 3 product category variables (category 1,2 and 3) which indicate which category the product user bought belonged to. The customer occupation variable has 20 unique values (ranging from 0-20) with one of the values assigned to every customer.

III. HYPOTHESIS

The project is guided by the hypothesis that a specific set of products purchased by the customers have interdependencies with other set of products. The hypothesis basically suggests that there is a logical association between product purchases, so if a customer buys a certain set of items, he is also likely to also buy a different set of items which have certain level of association with the products he purchased.

IV. BUSINESS VALUE

- Using association rules cross-selling and up-selling of products can be carried out.
- Products can be bundled and sold together as a part of promotional campaigns.
- Optimal store layouts can be designed according to the analyzed association rules such that it would increase sales.
- Apriori algorithm can also help in analyzing consumer purchase trends.

V. LITERATURE REVIEW

Various techniques have been used for association rule mining in the past. In [1] the authors compare several association rule techniques like the AIS algorithm, Apriori algorithm, Eclat and FP-Growth algorithm. The AIS algorithm uses candidate generation for mining frequent patterns. In the candidate generation technique only one item at a time is considered while generating association rules. So, a single item is compared with all the other items in the transaction in the first step, then two items with a single item and this goes until all the items in the transactions are covered. The algorithm is hence computationally inefficient and expensive. The SETM algorithm uses relational operations for mining frequent patterns and was created for SQL. The SETM algorithm works in a similar fashion as the AIS algorithm except it uses SQL join operations to generate new candidates. A major drawback of SETM in addition to being computationally expensive, is that it requires data to be stored in a SQL database. The Eclat algorithm uses bit matrices to find the support associated with every object in the transaction. In the bit matrix every column corresponds to a transaction and every row to an item. A true bit is associated to every item present in the transaction while a false bit to the ones that are absent. The algorithm in a depth first manner traverses the prefix tree of the bit matrix. The nodes of the tree that intersect with other nodes (i.e. transactions with similar itemsets) are retained while others are discarded. A drawback of this algorithm is that it performs poorly when the size of the transaction is increased. The FP-Growth algorithm uses a frequent pattern tree for association rule mining. The FP-tree divides the data into partitions based on their frequency. And then generates a prefix tree with every node in the tree representing an item and a set of transactions is represented by the path. It compresses a large database into a tree structure. While it is very effective for large databases the algorithm performs poorly for small sets of data. Another drawback of the fp-growth algorithm is that understanding the tree structure becomes very difficult with complex associations. The apriori algorithm employs a breadth first search for finding association rules. The association rules are primarily based on two parameters, support and confidence. The performance of the algorithm is not dependent on the size of the data. It is computationally more efficient compared to AIS and SETM. Although FP-growth is more efficient computationally, apriori is easier to interpret as it clearly lays down association rules.

Association rule mining using apriori algorithm has been carried out several times in the past and continues to remain a widely used technique. Apriori algorithm has been implemented in a variety of business scenarios from identifying changes in market data trends to resource allocation. In [2] the authors use apriori algorithm to mine association rules for allocating resources by using data from an event log. They try to find ordered co-relationships among items in an event log dataset. Resource allocation is a big issue in business process management, and they tackle it using two different association rule techniques, apriori and resource allocation rule miner. Their study finds out that apriori algorithm generates more decision rules than the resource allocation miner but is computationally more expensive and hence also has more execution time. A drawback of association rule mining techniques is that it cannot be used when data is dynamic i.e. if it changes with time. Once the values of support, lift and confidence are calculated by the algorithm they are not automatically recalculated if there is a change in data. The authors in [3] tweak the apriori algorithm for it to recalculate these values when presented with dynamic data. They use a bakery dataset which has 26 items and find association rules

among these items but have 4 windows of data which are updated with time. In [4] the authors explain the use of association rules in a multistore environment such as in the case of Walmart or Tesco, they argue that the existing methods fail to find associations in these scenarios due to the underlying assumption that all the products are present on shelves in each store at all times. They propose an algorithm which is an extended version of Apriori wherein they also calculate store chain association rules along with product association rules and test it on a synthetically generated dataset. In [5] the authors use association rule techniques for giving personalized recommendations to users. They acquire their dataset from a website log, they modify the apriori algorithm a little to suit their data and find association rules between user visited websites and recommend websites to user based on these rules.
In [6] the authors select 22 commercial business problems related to finance, insurance, retail, manufacturing and telecommunication and analyse different data mining techniques that can be applied to every scenario. They give a thorough analysis of how different set of algorithms which include clustering, association rules, classification can be used in different business cases. In their research they determine that for finding relationships among customer characteristics for the business application of cross-selling Apriori algorithm is the most suitable data mining technique.

VI. METHODOLOGY

Association Rule mining is a most widely used method for finding hidden associations between objects and is carried out by finding frequent patterns among objects in a transactional database.
Apriori algorithm was first introduced by authors in [7] as a means to find association rules between products. Support and confidence are two measures based on which association rules are determined. If there are two objects, the association rule is generally given by:

Object1→Object2 [Support, Confidence]

The support of an association rule is the number of transactions in which Object1 and Object2 are bought together and can be represented as:

$$\text{Support (Object1}\rightarrow\text{Object2)} = \frac{frequency(Object1\ and\ Object2\ bought\ together)}{Total\ number\ of\ transactions}$$

Whereas the confidence of an association rule shows the frequency with which object 2 is bought with object 1 and can be represented as:

$$\text{Confidence (Object1}\rightarrow\text{Object2)} = \frac{frequency(Object1\ and\ Object2\ being\ bought\ together)}{Frequency(Object1)}$$

The apriori algorithm can hence be decomposed into these logical steps:

1. As the algorithm follows a bottom-up approach, all the frequent item-sets are first calculated along with their support and confidence.
2. Then the item-sets are compared to a threshold support and frequency which is set by the user, all the item-sets below the set threshold are filtered out.
3. Items that are below the set threshold are divided into smaller subsets of items and then the support and confidence values are recalculated for these subsets and compared with our set threshold. This procedure is repeated until convergence.
4. Finally, a set of association rules are generated with support and confidence values above the threshold.

A major drawback with just using the support and confidence in mining association rules is that the individual frequency of the object on the right-hand side of the association rule (object 2 in our case) is not taken into consideration. So, another measure called lift will be taken into consideration while applying the apriori algorithm, it considers the individual frequencies of both the entities on the right as well as the left side of the association rules. Lift of an association rule can be expressed as:

$$\text{Lift (Object1}\rightarrow\text{Object2)} = \frac{Support(Object1 \rightarrow Object2)}{Support(Object1) * Support(Object2)}$$

VII. IMPLEMENTATION AND INTERPRETATION OF THE APRIORI ALGORITHM.

Some co-relations between customer purchases and other features in our dataset are visually represented before the implementation of apriori algorithm.

1.1 Dealing with missing values

The dataset came with a few missing values in some product categories. These were replaced with zero as if the product belonged to category1 it basically meant,

the product did not belong to category 2 or category 3, and hence those values were left unfilled in the dataset and replacing them with any other value would not be appropriate.

1.2 Visual Analysis of Co-relationships between Customer Purchases and their Demographic Information.

1.2.1 Which Gender Had More Turnout to the shop and Who Spent More?

It is vital to know which gender had the most turnout for shopping, to know who the key consumers are and can then be targeted for specific marketing campaigns.

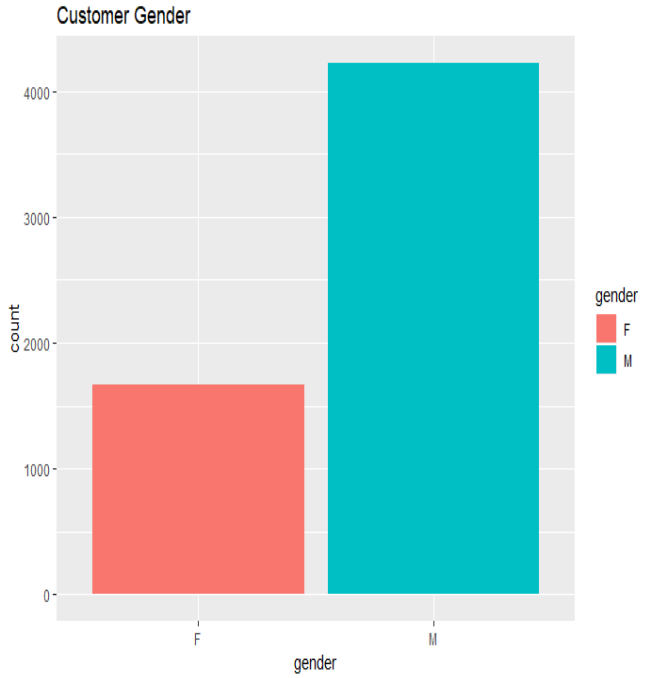

Fig.1

```
gender
F:1666
M:4225
```

```
cust_gender = customer_data %>% select(gender,u_id) %>%
group_by(u_id) %>% distinct()
summary(user_gender)
gender<-cust_gender$gender
gender_plot  = ggplot(data = cust_gender) + g
eom_bar(mapping = aes(x = gender, y = ..count.., fill = gender))
+labs(title = 'Customer Gender')
plot(gender_plot)
```

The count of male customers was way higher than the female customers, which would suggest focusing more aggressively on the male customers. The count of males was nearly 2.5 times greater than females, but in case of average spending by the two genders we can see this gap narrowing down.

Fig.2

```
  gender     C  purchases     Avg
  <fct>  <int>      <dbl>   <dbl>
1 F       1666 1186232642 712024.
2 M       4225 3909580100 925344.
avg_spending = purchase_and_gender %>% group_by(gender) %>%
summarize(C = n(),
purchases = sum(Purchase_value_total), Avg = purchases/C)
print(avg_spending)
avg_spend  = ggplot(data = avg_spending) +
geom_bar(mapping = aes(x = gender, y =Avg, fill = gender),
stat = 'identity')
+labs(title = 'Average Spending By Gender')
plot(avg_spend)
```

Women on an average spend very close to what men do in this retail scenario, men despite being 2.5 times greater in count only spend 1.2 times higher than women.

1.2.2 Which Age Bracket Spends the Most?

If the retail decides on having an email marketing campaign and if users were to be selected at random, we could in fact target our consumers based on the age bracket to get a higher response. It is vital to understand which age bracket the retail store appeals to the most. The next visualization explains the gender distribution of the customers with respect the purchases made by them.

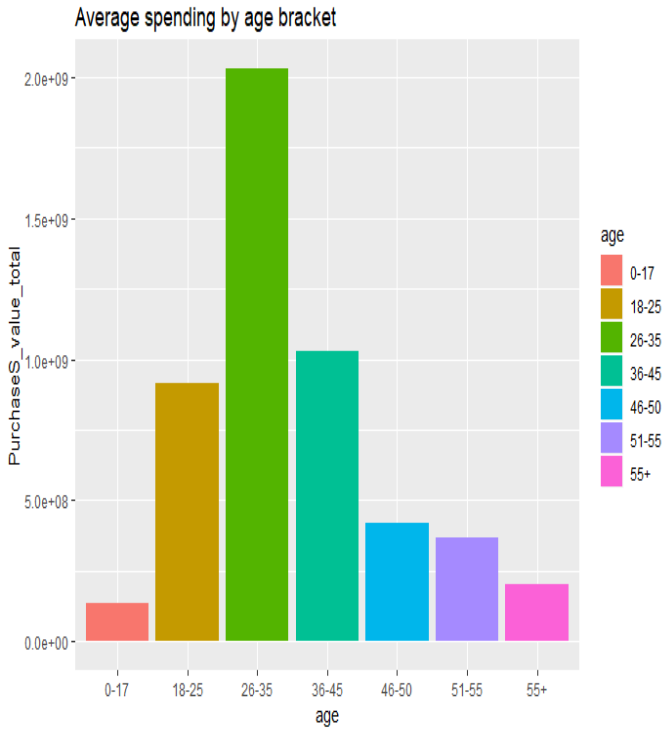

Fig.3

	age	PurchaseS_value_total
	<fct>	<int>
1	0-17	134913183
2	18-25	913848675
3	26-35	2031770578
4	36-45	1026569884
5	46-50	420843403
6	51-55	367099644

```
cust_age = customer_data %>% select(age,u_id,purchase) %>%
distinct() %>% count(age)
age_purchases = customer_data %>%select(purchase,u_id,age) %>%
group_by(age) %>% arrange(age) %>%
summarise(PurchaseS_value_total = sum(purchase))
head(age_purchases)
age_purchase<-  ggplot(data = age_purchases) +
geom_bar(mapping =aes(x = age, y =PurchaseS_value_total, fill = age),
stat = 'identity')
+labs(title = 'Average spending by age bracket')
plot(age_purchase)
```

We can observe that our most profitable customers lie in the 26-35 age bracket, 39% of the total purchases were made by this age bracket.
60% of the total purchases came from the combined age bracket of 26-45. This age bracket would be a very good target for marketing campaigns.

1.2.4 How Are Purchases affected By the Marital Status of the Customer?

The market can be segmented based on the marital status of the customers. It is crucial to find out which segment the shop appeals to the most, whether it is married or unmarried individuals.

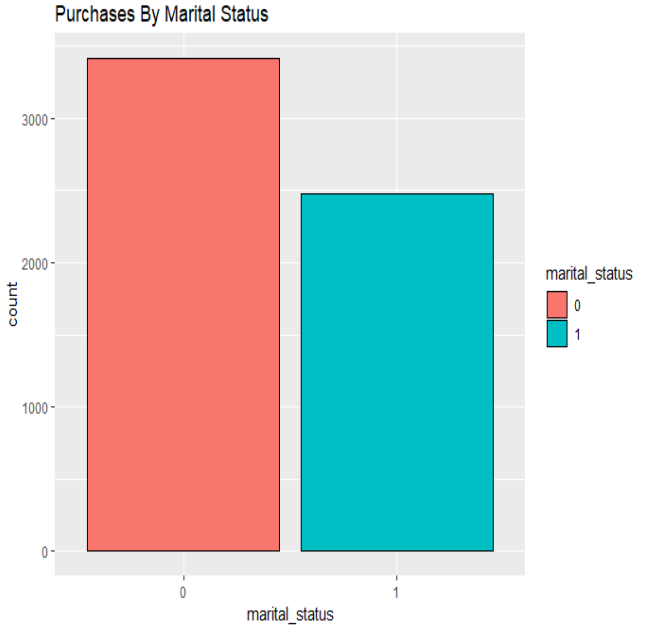

Fig.4

```
m_status = customer_data%>%select(u_id, marital_status) %>%
group_by(u_id) %>%
distinct()
ms_plot<- ggplot(data = m_status)+geom_bar(color="black",
mapping=aes(x=marital_status,y=..count..,fill=marital_status))
+labs(title = "Purchases By Marital Status")
plot(ms_plot)
```

Purchases made by unmarried shoppers are greater than those made by married ones. We can use this information to selectively market products to these customers. This information can be used to clone customers.

1.2.3 Does the Location of the Consumer Affect the Purchase Patterns?

Since the business is located across 3 cities it is important to understand which city attracts more customers. Also, it equally important to find out if more customers in a city equates to higher sales.
This information is useful if the business decides in the near future to shut down of one its less profitable locations and rather focus on the more profitable locations by expanding them.

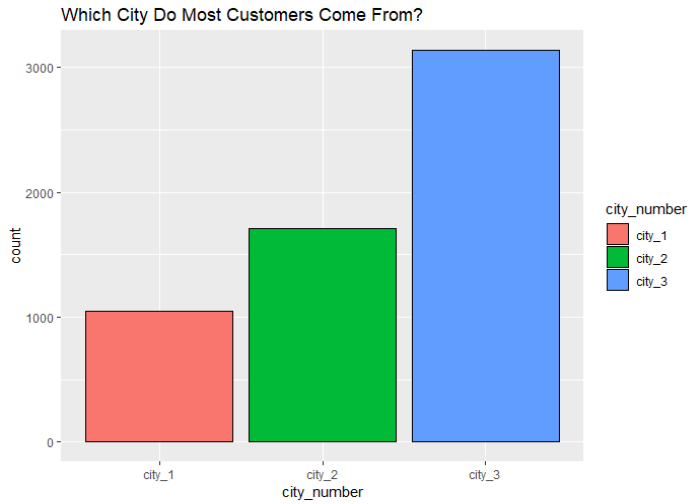

Fig.5

```
cust_loc =  customer_data %>% select(u_id, city_number,purchase)
%>% distinct()
head(cust_loc)
loc_plot = ggplot(data = cust_loc) + geom_bar(color = "black",
mapping = aes(x = city_number, y = ..count.., fill = city_number)) +
labs(title = 'Which City Do Most Customers Come From?')
plot(loc_plot)
summary(cust_loc)
```

It can be observed that most consumers of the store hail from city 3 nearly half from city 2 and one fourth from city 1.

Fig.6

```
city_1<-filter(customer_data,city_number=="A")
city1_sum<-sum(city_1$purchase)
city_2<-filter(customer_data,city_number=="B")
city2_sum<-sum(city_2$purchase)
city_3<-filter(customer_data,city_number=="C")
city3_sum<-sum(city_3$purchase)
#creating a new dataframe
City_Purchase<-c(city1_sum,city2_sum,city3_sum)
City_Number<-c("City_1","City_2","City_3")
city_purchase_final<-data.frame(City_Number,City_Purchase)
city_purchase_final
city_purchase_plot = ggplot(data = city_purchase_final) +
geom_bar(color = "black", mapping = aes(x = City_Number,
y = City_Purchase, fill = City_Number), stat = 'identity') +
labs(title = 'Which City Spends The Most?')
plot(city_purchase_plot)
```

Whereas from the above visualization we can observe that although most customers hail from city 3, customers spent more in city 2. This probably suggests that although the customers in city 2 were less they were buying more expensive products. Based on this information segmentation of the market can be carried out and inventory of these stores can also be managed. City 3 which spends less on products can have marketing campaigns promoting products that are value for money whereas in City 2, more expensive range of products can be marketed.

1.3 Implementing the Apriori Algorithm.

1.3.1 Data Pre-processing:

Implementation of the apriori algorithm requires data to be presented in a sparse matrix format and does not accept any other values except for binary values (1/0). So, the data was first transformed and spread in a way such that every user ID was allocated a column representing every product ID, and if the user purchased the product the column was assigned the value 1 else 0. The sparse matrix created then needs to be stored in a transaction class which is carried out by the read.transactions function in R.

```
product_sparse<-customer_data %>%
select(u_id,p_id)%>% group_by(u_id) %>% arrange(u_id)
sparse_matrix_products = product_sparse %>%
dplyr::mutate(id = row_number()) %>% spread(u_id, p_id)
head(sparse_matrix_products)
write.csv(customers_products, file = 'C:/Users/rohan/OneDrive
/Desktop/1.ACRM-Proposal/sparse_matrix_products.csv')
customer_product_data = read.transactions('C:/Users/rohan
/OneDrive/Desktop/1.ACRM-Proposal
/customers_products.csv',
sep = ',', rm.duplicates = TRUE)
```

```
> summary(customer_product_data)
transactions as itemMatrix in sparse format with
 5892 rows (elements/itemsets/transactions) and
 10548 columns (items) and a density of 0.008962118

most frequent items:
P00265242 P00025442 P00110742 P00112142 P00057642   (Other)
     1880      1615      1612      1562      1470    548846

 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 7.00   27.00   55.00   94.53  118.00 1027.00
```

The summary indicates that there are 5,892 unique transactions made over 10,548 items. The density value indicates the of non-empty cells in our matrix, its basically the total number of items that were purchased by the consumer divided by all the possible items present in our dataset. The summary also includes the mean, median and quartile information of the products sold. It shows that on an average a customer bought 94 products but as we can observe from the max(1027) and third quartile value(118) that some customers bought way more than 94 products in the range of 1000's which would be deemed outliers in this scenario, so using median would make more sense to analyze how many products an average customer bought which is 55.

1.3.2 Item Frequency Plot:

A plot of the most sold products can be represented using the arules library. The visualization below represents 15 most frequently bought products.

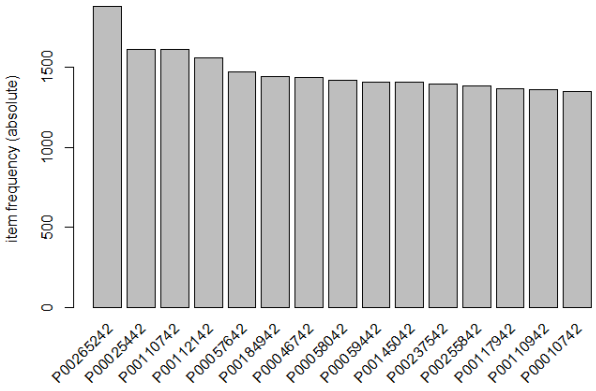

Fig.7

```
itemFrequencyPlot(customer_product_data,topN=15, type="absolute")
```

It can be observed that Item 'P00265242', 'P0025442' were the most frequently bought items. The less frequently bought items probably are not noticed as much, so to increase their sales they could be placed with these more frequently bought items.

1.3.3 Finding Association Rules and Interpretation of the Results.

Before implementing the Apriori algorithm we must set up the support and confidence values. Support is related to the minimum frequency of an item in a repository. So, we must determine how frequently an item must appear in the dataset before it can be a part of the association rule.

To have strong association rules, the confidence of the rules i.e. probability that a customer will purchase product B given that he already has purchased product A, should also be as high as possible. While this is true the support, which is the frequency of the products being purchased, should also be high.

In this setup, the minimum number of transactions will be set up to 40, which means a product has to appear at least 40 times in our purchases before an association rule can be formed related to the product. Since our total number of transactions are 5,892 our support value will be equal to (40/5892) which is equal to 0.00678. A confidence above 80% would suggest strong association rules which can be relied upon. The confidence value is hence set to 88% to achieve strong association rules. As discussed in the methodology the confidence of an association rule will only take into account the individual frequency of products on the left-hand side of the association rule to overcome this drawback, we also take into account a measure called lift. A positive value of lift indicates the products on the left-hand side of the association rule are dependent on the right-hand side of the rule. Higher value of lift indicates a stronger association rule between products.

```
rules_customer_product = apriori(data = customer_product_data, parameter =
list(support = 0.00678, confidence = 0.88, maxtime = 0))
inspect(sort(rules_customer_product, by = 'lift'))
```

```
    lhs                                    rhs         support     confidence lift     count
[1] {P00006942,P00127842,P00277642}     => {P00145042} 0.006788866 0.8888889  3.724988 40
[2] {P00025442,P00105142,P00127342,P00173842} => {P00057642} 0.006958588 0.8913043  3.572493 41
[3] {P00057942,P00102642,P00111142,P00183342} => {P00110742} 0.006788866 0.8888889  3.248966 40
```

With confidence greater than 88% we derive 3 association rules. The rules can be read as, for instance the first rule: Whenever a customer buys products 'P0006942', 'P00127842' and 'P00277642' there is an 88.8% chance that the customer will also buy product 'P0014502'. The support value in our output will always be above our set threshold which we have determined before. A count of 40 indicates, that 40 times in our dataset these four products were bought together.

We can observe that the product on the right-hand side of the first association rule('P0014504') is not present in top 15 most frequently bought products, whereas products on the left-hand side of the association rule ('P0006942', 'P00127842',' P00277642') are present. So, these three products can be bundled and sold together which can drastically increase the sales of our less frequently bought product.

The association rules between these products can be visually represented using a plot from the arulesViz library in R.

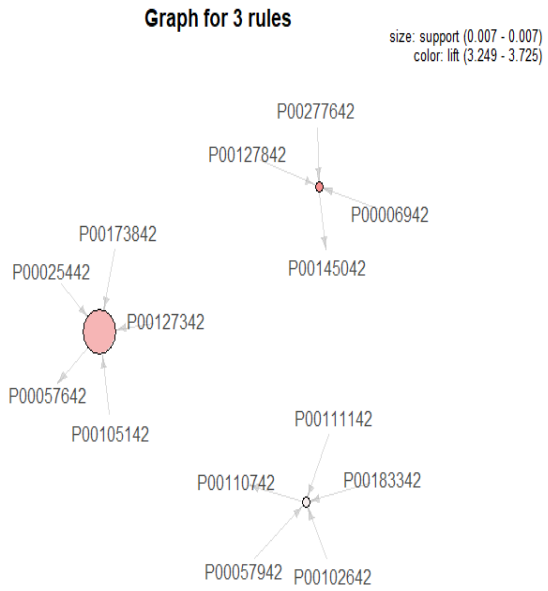

Fig.8

The incoming arrows represent the left-hand side of the association rule while the outgoing arrows represent the right-hand side.
Higher support values are represented by large size of the bubbles and darker shades of bubbles represent higher lift values.

The association rules can be increased from three rules to even more but that comes at the cost of lower confidence.

	lhs		rhs	support	confidence
[1]	{P00113042,P00221242}	=>	{P00112542}	0.006788866	0.8510638
[2]	{P00110742,P00155442,P00209742}	=>	{P00112542}	0.006788866	0.8510638
[3]	{P00057642,P00111142,P00154042,P00183342}	=>	{P00270942}	0.006958588	0.8541667
[4]	{P00006942,P00127842,P00277642}	=>	{P00145042}	0.006788866	0.8888889
[5]	{P00014842,P00113042,P00117442}	=>	{P00110942}	0.006788866	0.8510638
[6]	{P00032042,P00110942,P00222942}	=>	{P00145042}	0.006958588	0.8723404
[7]	{P00128942,P00157542,P00277442}	=>	{P00145042}	0.007637475	0.8653846
[8]	{P00051442,P00110742,P00117442,P00125942}	=>	{P00145042}	0.006958588	0.8541667
[9]	{P00112542,P00117442,P00255842,P00271142}	=>	{P00058042}	0.007467753	0.8627451
[10]	{P00025442,P00105142,P00127342,P00173842}	=>	{P00057642}	0.006958588	0.8913043
[11]	{P00140742,P00157542,P00184442}	=>	{P00059442}	0.006788866	0.8510638
[12]	{P00057942,P00102642,P00111142,P00183342}	=>	{P00110742}	0.006788866	0.8888889
[13]	{P00046742,P00057942,P00058042,P00173842}	=>	{P00110742}	0.007128310	0.8750000
[14]	{P00105142,P00193542,P00299142}	=>	{P00110742}	0.006788866	0.8510638
[15]	{P00003442,P00112142,P00217442}	=>	{P00025442}	0.006788866	0.8510638

Fig.9

```
rules_customer_product_2 = apriori(data = customer_product_data,
parameter = list(support = 0.00678, confidence = 0.85, maxtime = 0))
inspect(sort(rules_customer_product_2, by = 'lift'))
```

When the confidence is set to 85%, we can generate 15 association rules and after decreasing the confidence to 80% 303 association rules between products can be generated.
A lower confidence in association rules would translate to a higher risk in implementing the association rule in real life. Because if say we use an association rule with a lower confidence value and based on it if we bundle certain products together and introduce them in the market, it could be possible that customers may not like the combination of the products. This promotional mix may backfire, and customers might turn to other stores where they would rather buy the individual product.
Finding the right confidence value would therefore depend upon if taking the additional risk of a lower confidence value would translate into higher profits.

VIII. CONCLUSION

In this project Apriori algorithm was successfully implemented and association rules between products with confidence greater than 88% were determined. The guiding hypothesis for the project were also supported by the results as there exists a logical association between product purchases. This information can be leveraged to design store layouts based on the consumer purchase trends to maximize sales and hence profits. A thorough analysis of customer behavior patterns can also be carried out. Introducing consumers to new products based on their purchase patterns and hence cross-selling products is also made easy and straightforward by this technique. And it is not just restricted to it has various other

benefits such as up-selling products in the form of add-ons and analyzing seasonal trends.

IX. REFERENCES

[1] M. Girotra, K. Nagpal, S. Minocha and N. Sharma, "Comparative Survey on Association Rule Mining Algorithms", *International Journal of Computer Applications*, vol. 84, no. 10, pp. 18-22, 2013. Available: 10.5120/14612-2862

[2] Z. Huang, X. Lu and H. Duan, "Mining association rules to support resource allocation in business process management", *Expert Systems with Applications*, vol. 38, no. 8, pp. 9483-9490, 2011. Available: 10.1016/j.eswa.2011.01.146

[3] M. Kaur and S. Kang, "Market Basket Analysis: Identify the Changing Trends of Market Data Using Association Rule Mining", *Procedia Computer Science*, vol. 85, pp. 78-85, 2016. Available: 10.1016/j.procs.2016.05.180

[4] Y. Chen, K. Tang, R. Shen and Y. Hu, "Market basket analysis in a multiple store environment", *Decision Support Systems*, vol. 40, no. 2, pp. 339-354, 2005. Available: 10.1016/j.dss.2004.04.009

[5] E. Lazcorreta, F. Botella and A. Fernández-Caballero, "Towards personalized recommendation by two-step modified Apriori data mining algorithm", *Expert Systems with Applications*, vol. 35, no. 3, pp. 1422-1429, 2008. Available: 10.1016/j.eswa.2007.08.048

[6] J. Seng and T. Chen, "An analytic approach to select data mining for business decision", *Expert Systems with Applications*, vol. 37, no. 12, pp. 8042-8057, 2010. Available: 10.1016/j.eswa.2010.05.083

[7] R. Agrawal, T. Imieliński and A. Swami, "Mining association rules between sets of items in large databases", *Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93*, 1993. Available: 10.1145/170035.170072

APPENDIX

ACTIVITY LOG:

Week	Actions Carried Out
Week 1	Exploring topics on which the project can be implemented.
Week 2	Studying various methodologies that can be implemented and finalizing the dataset which will be used for the project.
Week 3	Setting up the hypothesis guiding the project and finalizing the methodology to be used.
Week 4	Understanding the data by exploratory analysis.
Week 5	Data wrangling and Implementation of the chosen algorithm.
Week 6	Interpretation and analysis of the Results.
Week 7	Writing the first draft of the report.
Week 8	Making changes in the report where ever they are necessary and preparing the final report.