## A

**PROJECT REPORT ON**

**Food Item Recommender System**

**Submitted by,**

|  |  |
| --- | --- |
| **Niranjan Patil** | **T184104** |
| **Mehul Patil** | **T184099** |

**Guided by,**

**Mrs. Vaishali Wangikar**

**A Report submitted to MIT Academy of Engineering Alandi submitted in partial fulfilment of the requirement for Sixth Semester of BACHELOR OF TECHNOLOGY in School of Computer Science and Technology.**



**SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY**

## MIT Academy of Engineering Dehu Phata, Alandi (D)

**Pune - 412105, Maharashtra (India) 2018-19**

|  |  |
| --- | --- |
|  | School of Computer Science and Tech  **(Accredited by NBA, ISO 9001:2008 Certified)** |

**CERTIFICATE**

**Alandi (D), Pune – 412105**

This is certify that,

|  |  |
| --- | --- |
| **Niranjan Patil** | **T184104** |
| **Mehul Patil** | **T184099** |

of T. Y. B. Tech. have submitted a Report on,

## Food Item Recommender System

The said work is completed by putting the requirement of hours as per prescribed curriculum during the academic year 2018 – 19. The report is submitted in the partial fulfillment of the requirements for the course **Predictive Analytics** in the Sixth Semester of Degree of Engineering in MIT Academy of Engineering.

Date: 27/04/2019 Place: MIT AOE

Signature: Signature:

Name: Name:

**Internal Examiner External Examiner**

## ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our project guide Mrs. Vaishali Wangikar for providing her invaluable guidance, comments and suggestions throughout the course of the project, constantly motivating us to work harder. The constant motivation and encouragement in the due of completion of this project, ensured that we exceed our best to accomplish our desired results.

Also, we would like to thank Prof. Ranjana Bhadre; Dean, School of Computer Science and technology for providing us with the opportunity to craft our ideas and shape them into valuable projects. We would like to profoundly thank all those who have directly or indirectly guided and helped us in the project work.

## ABSTRACT

Exponential growth of the world-wide-web and emergence of e-commerce as a platform within reach of common customers had led to the development of numerous recommender systems that defines a personalized information retrieval technique used to identify set(s) of items that will be of interest to a certain user. Most of these researches related item recommender on the basis of user profile and item-oriented recommendation. But here we are exploring scope of frequent item set based recommendation by implementing Apriori algorithm. Apriori is mainly used to find frequently purchased items/products. The key idea behind this recommendation is that any item set that occurs frequently together must have each item (or any subset) occur at least as frequently.

## CONTENTS

|  |  |  |  |
| --- | --- | --- | --- |
| Title | | | Page Number |
| 1. | Introduction | | 6 |
|  | 1.1 | Problem Statement | 7 |
|  | 1.2 | Objectives | 7 |
| 2. | Dataset | | 8 |
|  | 2.1 | Data Collection | 8 |
|  | 2.2 | Dataset Description | 8 |
| 3. | Methodology | | 10 |
| 4. | Code | | 11 |
| 5. | Results and Explanation of all Parameters | | 14 |
| 6. | Conclusion | | 19 |
| 7. | Limitations and Further Enhancements | | 20 |
| 8. | References | | 21 |

1. **INTRODUCTION**

In recent years Internet is growing so rapidly which is leading in increase of activities. Whether be an old or young, rich or poor people are using Internet freely. One of the mostly used services is recommender system. Recommender Systems gives recommendation to the customers based on history, rating and the previous purchases of the customers with same taste. Recommender Systems helps user to take correct decisions, redefine browsing and enhance their experience. Recommender Systems are used in various domains like film, health, education, hospitality and many more. One of the domains which is growing its popularity is use of Recommender Systems for food recommendation. The craze of trying new dishes and cuisines is increasing among people, in such situations people who are unaware about most of the dishes and cuisines, so Recommender Systems can help them on the basis of their history and history of people who have similar taste and had bought that dish.

There are many reasons which can make food Recommender Systems challenging, not only the fact of encouraging people to try new dishes or dishes that go complimentary with the dishes they have purchased, but also it may be challenging in predicting what people will actually like to eat and analyzing their taste and openness to new cuisines.

## Problem Statement

Many a time customers are confused about the orders they need to place in Restaurants. They are unable to find the dishes which go complimentary with their taste or dish which they have purchased. With thousands of options available it becomes difficult for customers to explore each one of them and many a times they skip it because of lack of time. So all they need is system that will give them recommendations about dishes in very less amount of time.

## Objectives

* To design the system that will recommend user the dishes complimentary to dishes they have bought.
* To design the system that will recommend dish from different categories.

# DATASET

## Data Collection

The data was collected to the food items that are bought together. The all items were listed according to bill ids. These bills were collected from a local eatery in Alandi.

## Dataset Description

The dataset is shown in figure below. The dataset contains 3 columns and 60 rows. In total 10 order ids i.e. from 1 to 10, 6 categories i.e. Paratha, Raita, Beverages, Pakoda, Pickles and Oil, and labels total 18 labels 3 in each category.

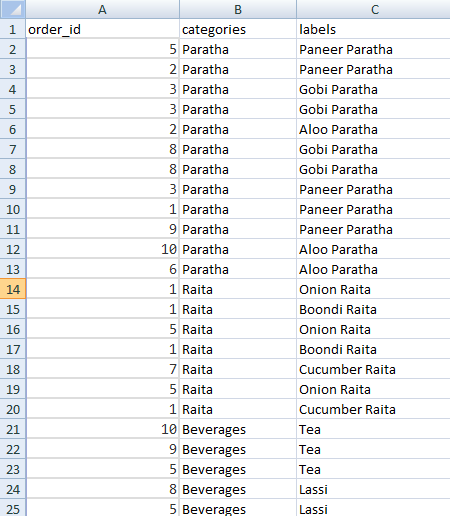


Fig. 2.1 Dataset

Initially all the food items having same order\_id was clubbed into one transaction and the column categories is dropped because here we are not using multilevel Apriori because we need not require recommendations from same categories as we are focusing on food items that will lead into whole meal platter

The example of transactions after clubbing is shown in figure below which shows the transactions and transaction ids.

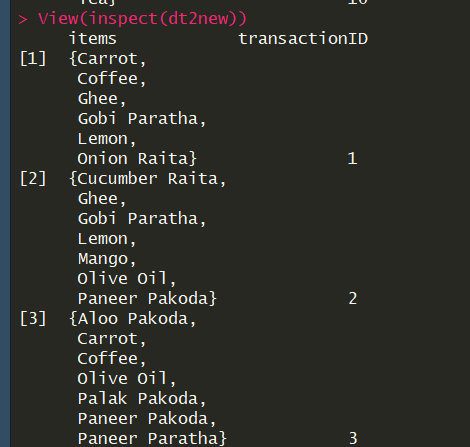
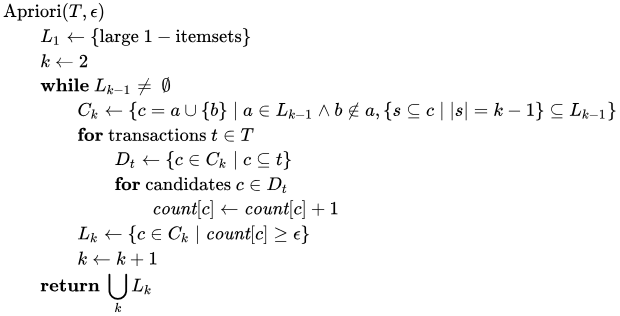


Fig. 2.2 Transactions

# METHODOLOGY

Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as *candidate generation*), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

Apriori uses breadth-first search and a Hash tree structure to count candidate itemsets efficiently. It generates candidate itemsets of length k from item sets of length k-1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The pseudo code for the algorithm is given below for a transaction database T, and a support threshold of epsilon . Usual set theoretic notation is employed; though note that T is a multiset. c(K) is the candidate set for level k. At each step, the algorithm is assumed to generate the candidate sets from the large itemsets of the preceding level, heeding the downward closure lemma. count[c] accesses a field of the data structure that represents candidate set c which is initially assumed to be zero. Many details are omitted below, usually the most important part of the implementation is the data structure used for storing the candidate sets, and counting their frequencies.

# CODE:

#installing required packages install.packages("readr") install.packages("arules") install.packages("arulesViz") install.packages("splitstackshape") install.packages("plyr") install.packages("dplyr") install.packages("tidyverse") install.packages("RColorBrewer") install.packages("ggplot2") install.packages("knitr") install.packages("grid") install.packages("rCBA")

#loading required packages require(readr) require(arules) require(splitstackshape) require(plyr)

require(dplyr) require(tidyverse) require(RColorBrewer) library(readxl) library(arulesViz)

#load excel file

snacks<-read\_excel("C:/Users/Arti/Desktop/Academics/PA/Apriori/snacksNew.xlsx") dtnew<- split(snacks$labels,snacks$order\_id)

# Converting data to a class of transactions dt2new = as(dtnew,"transactions")

#Plotting itemFrequencyPlot(dt2new,topN=20,type="absolute")

rulesNew = apriori(dt2new, parameter=list(support=0.1, confidence=0.8, minlen = 3)) plot(rulesNew,control=list(col=brewer.pal(11,"Spectral")),main="")

length(rulesNew) inspect(rulesNew[1:10])

# Remove Unnecessary Rules subset.rules =

which(colSums(is.subset(rulesNew, rulesNew)) > 1)

# get subset rules in vector

"which() returns the position of elements in the vector for which value is TRUE. colSums() forms a row and column sums for dataframes and numeric arrays. is.subset() Determines if elements of one vector contain all the elements of other" length(subset.rules)

# to delete the subset from superset as superset will have subset rules = rulesNew[-subset.rules]

length(rules)

# What are customers likely to buy before they purchase "Boondi Raita" rulesBoondiRaita<-apriori(dt2new, parameter=list(supp=0.1,conf = 0.8), appearance = list(default="lhs",rhs="Boondi Raita"))

rulesBoondiRaita<-sort(rulesBoondiRaita, decreasing=TRUE,by="confidence") inspect(rulesBoondiRaita[1:10])

toprules = rules[1:15] plot(rules, method = "graph")

plot(toprules, method = "graph", engine = "htmlwidget")

toprules\_lift = head(rules, n=20, by ="lift") plot(toprules\_lift, method="paracoord")

# RESULTS

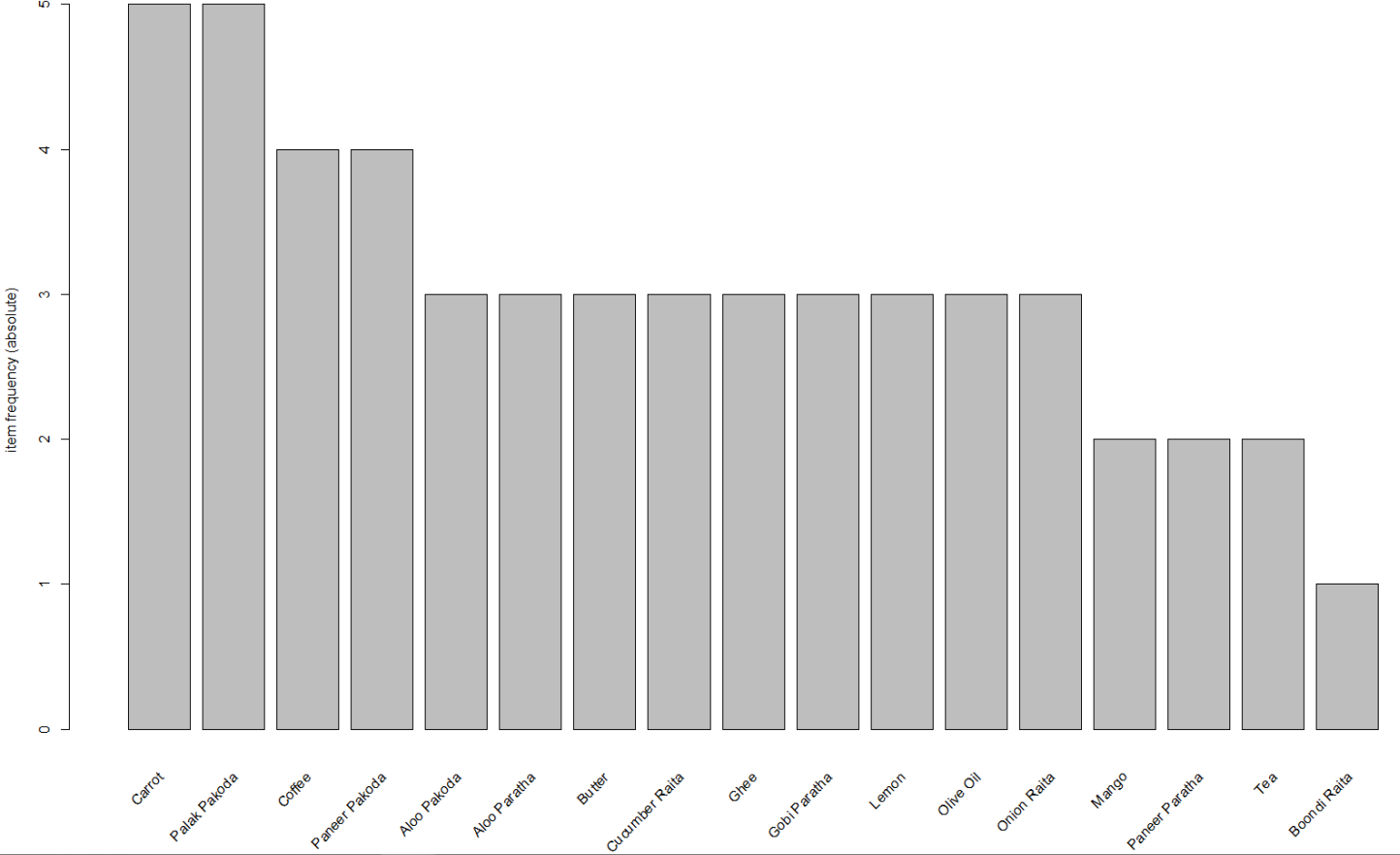


Fig 5.1 Frequent itemset



Fig 5.2 Graph of Association Rules

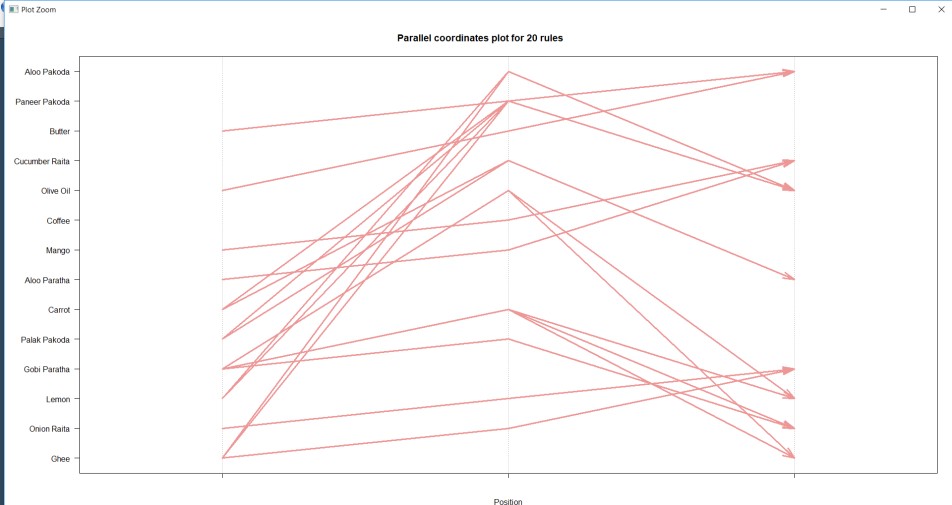


Fig 5.3 Top 20 rules

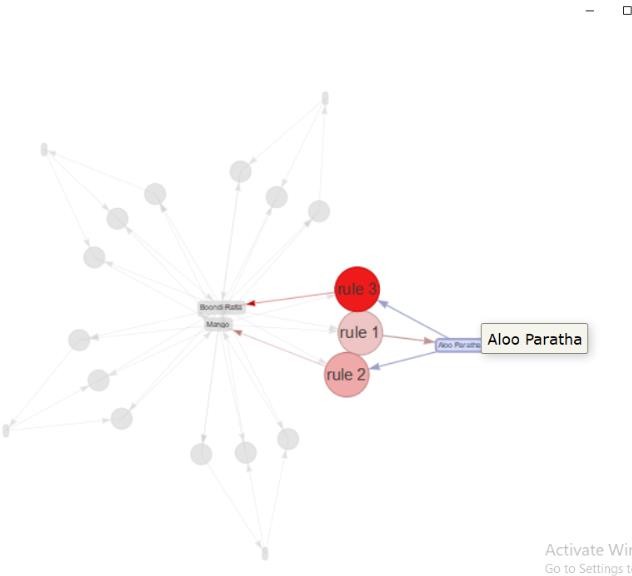


Fig 5.4 Top 3 rules

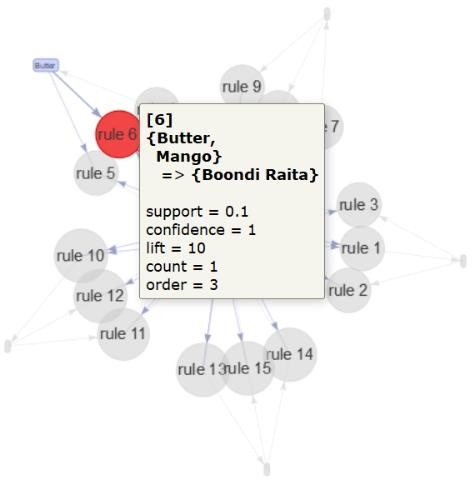


Fig 5.4 Rule 6

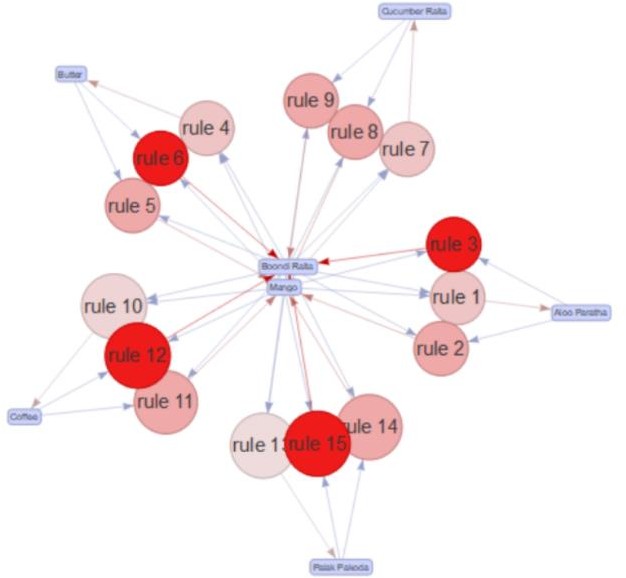


Fig 5.5 Graph of Top 15 Association Rules

The summary of the rules gives us some very interesting information:

* The distribution of rules by length: a length of 4 items has the most rules.
* The summary of quality measures: ranges of support, confidence, and lift.
* The graph gives the visualization of all rules

## Explanation of all the parameters of the results

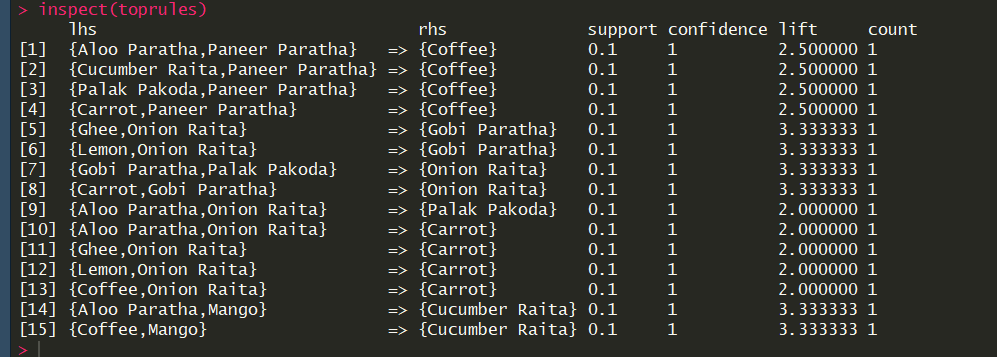


Fig 5.5 Results

Association rules are created by searching data for frequent if-then patterns and using the criteria

*support* and *confidence* to identify the most important relationships*.*

**Support** is an indication of how frequently the items appear in the data.

**Confidence** indicates the number of times the if-then statements are found true.

A third metric, called ***lift***, can be used to compare confidence with expected confidence.

Association rules are calculated from *itemsets*, which are made up of two or more items. If rules are built from analyzing all the possible itemsets, there could be so many rules that the rules hold little meaning. With that, association rules are typically created from rules well-represented in

18 | P a g e

data

## CONCLUSION

The Recommender System is built to give the recommendations of food items that ultimately leads to “Veg Thali” , which gives the customer a recommendation of that is required for whole meal platter. The recommendations also give the parameter which tells how likely and how frequently the food items are bought together. The results are achieved using apriori algorithm which deals the transactions of the food items which are build using the food items purchased in same bill.

## LIMITATIONS AND FURTHER ENHANCEMENTS

The results may vary with variation of support and confidence. The values of support and confidence must be accurately defined so as to get proper results. The system gives recommendation for the food items of Veg meal platter, the other food items are not taken into consideration in dataset. The algorithm is applied to only few transactions, large amount of dataset is not taken in consideration.

In future the system considering variety of food items can be made. The other algorithms like fp-growth, vertical frequent itemset mining can be used to go for comparative studies of these algorithms. The other factors like recommendations according to the intake of specific amount of calories, income of customers, geographical locations, season and many more.

## REFERENCES

[ 1 ]Christoph Trattner,David Elsweiler, “*Food Recommender System Important Contributions, Challenges and Future Research Directions*”,ResearchGate

[ 2 ] Machine Learning for Recommender systems — Part 1 (algorithms, evaluation and cold start), Recombee Blog

[ 3 ] Abhishek Saxena, Navneet K Gaur, “*Frequent Item Set Based Recommendation*

*using Apriori” ,* International Journal of Science, Engineering and Technology Research (IJSETR), Volume 4, Issue 5, May 2015