**A**

**PROJECT REPORT ON**

**Songs Recommendation System**

**Submitted by,**

|  |  |
| --- | --- |
| **Mehul Patil** | **T184099 (012016070)** |
| **Niranjan Patil** | **T184104 (0120160134)** |

**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,

|  |  |
| --- | --- |
| **Mehul Patil** | **T184099 (012016068)** |
| **Niranjan Patil** | **T184104 (0120160134)** |

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

## Songs Recommendation 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 fulfilment of the requirements for the course **Predictive Analytics** in the Sixth Semester of Degree of Engineering in MIT Academy of Engineering.

Date: 03/05/2019 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

Rapid development of mobile devices and internet has made possible for us to access different music resources freely. The number of songs available exceeds the listening capacity of single individual. People sometimes feel difficult to choose from millions of songs. Moreover, music service providers need an efficient way to manage songs and help their costumers to discover music by giving quality recommendation. Thus, there is a strong need of a good recommendation system.

But here we are exploring scope of frequent item set of songs 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 | Dataset Description | 8 |
| 3. | Methodology | | 10 |
| 4. | Code and Explanation of all Parameters | | 11 |
| 5. | Results | | 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.

With the explosion of network in the past decades, internet has become the major source of retrieving multimedia information such as video, books and music etc. Concerning a large amount of various data available on the Internet, there are existing websites which provides services for users to look for useful data. For text data in WebPages, the websites providing keyword-based searching or recommendation are developed, such as the search engine of Yahoo! [Yahoo] and the book recommendations of Amazon [Amazon]. For multimedia data, however, the websites providing such kinds of services are still limited. People have considered that music is an important aspect of their lives and they listen to music, an activity they engaged in frequently. Regarding the music recommendation, a preliminary Music, as a powerful communication and self-expression approach, therefore, has appealed a wealth of research.

## Problem Statement

Many a time peoples are interested about the song they need to listen in any place. They are unable to find the songs which go complimentary with their mood or age, etc criteria. Thousands of options available it becomes difficult for peoples to listen or searching songs each one of them and many a times they irritate it because of lack of time or not to search anything. So all they need is system that will give them recommendations about songs in very less amount of time and according to their priority.

## Objectives

* To design the system that will recommend user the songs according to their mood.
* To design the system that will recommend user the songs according to their artist.
* To design the system that will recommend user the songs according to their movie.

# DATASET

## Dataset Description

The dataset is shown in figure below. The dataset contains 6 columns and 1059 rows. According to 6 Columns are user id, Song title, Movie name, Artist name, mood of song, year of release. This attribute is select because basically peoples listen song on this attributes. This dataset is randomly generated by using excel built-in function like randbetween(), choose() and save in csv format.

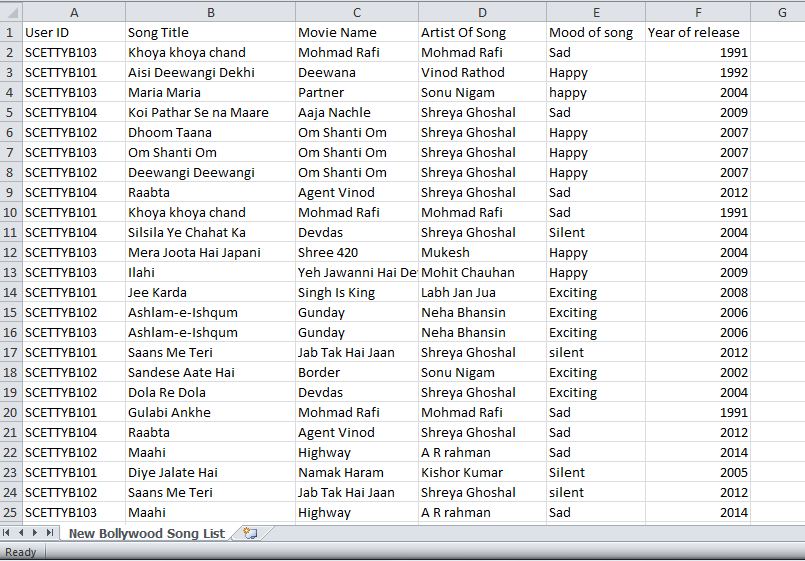
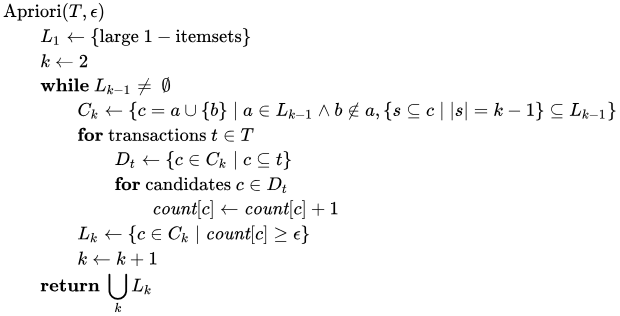


Fig. 2.1 Dataset

# 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:

# install.packages("dplyr")

# library(dplyr)

# install.packages("plyr", dependencies= TRUE)

# library(plyr);

# # arules’ available that provides functions to read the transactions and find association rules.

# install.packages("arules", dependencies=TRUE)

# library(arules)

# # Read the ‘New Bollywood song list ’ csv file.

# df\_SongsData <- read.csv("C:/Users/Admin/Desktop/Apriori/New Bollywood Song List.csv")

# glimpse(df\_SongsData)

# # we have to convert the dataframe into transactions format such that we have all the items

# # bought at the same time in one row. For this, we use a function called ddply, offered by

# # package plyr.

# # Mood wise assosiation rules

# # The next step is to actually convert the dataframe into basket format, based on the user id and mood of song

# # or artist or movies

# df\_SongList <- ddply(df\_SongsData,c("User.ID","Mood.of.song"),

# function(df1)paste(df1$Song.Title,

# collapse = ","))

# colnames(df\_SongList) <- c("","","song\_title")

# # Write the resulting table to a csv file. The reason we do this is, when we write a dataframe to

# # a .csv file, it attaches a row number by default. (unless, of course you were to explicitly tell

# # it not to, by using the argument “row.names=FALSE” in the write.csv function).

# write.csv(df\_SongList,"C:/Users/Admin/Desktop/Apriori/NewSongList.csv", row.names = TRUE)

# # Using the read.transactions() functions, we can read the file ItemList.csv and convert it to a transaction format

# txn = read.transactions(file="C:/Users/Admin/Desktop/Apriori/NewSongList.csv",

# rm.duplicates= TRUE, format="basket",sep=",",cols=1);

# # Parameters: Transaction file: ItemList.csv

# # rm.duplicates : to make sure that we have no duplicate transaction entried

# # format : basket (row 1: transaction ids, row 2: list of items)

# # sep: separator between items, in this case commas

# # cols : column number of transaction IDs

# txn@itemInfo$labels <- gsub("\"","",txn@itemInfo$labels)

# # Quotes are introduced in transactions, which are unnecessary and result in some incorrect results.

# # So, we must get rid of them:

# # Finally, run the apriori algorithm on the transactions by specifying minimum values for support and confidence.

# basket\_rules <- apriori(txn,parameter = list(sup = 0.03, conf = 0.05,target="rules"));

# df\_basket <- as(basket\_rules,"data.frame")

# View(df\_basket)

# summary(basket\_rules)

# # for finding correct rulles we speparte rules according to lift of 75% percentile according to sumary of rules

# basket\_rules = subset(basket\_rules, lift >= 29)

# df\_basket <- as(basket\_rules,"data.frame")

# View(df\_basket)

# # Plot a few graphs that can help you visualize the rules. Install and load the ‘arulesViz’

# # library for association rules specific visualizations:

# library(arulesViz)

# plot(basket\_rules)

# plot(basket\_rules, method = "grouped", control = list(k = 1))

# plot(basket\_rules, method="graph", control=list(type="items"))

# plot(basket\_rules, method="paracoord", control=list(alpha=.5, reorder=TRUE))

# plot(basket\_rules,measure=c("support","lift"),shading="confidence",interactive=T)

# # Movie wise Association Rules

# df\_SongList\_m <- ddply(df\_SongsData,c("User.ID","Movie.Name"),

# function(df1)paste(df1$Song.Title,

# collapse = ","))

# colnames(df\_SongList\_m) <- c("","","song\_title")

# write.csv(df\_SongList\_m,"C:/Users/Admin/Desktop/Apriori/M\_wise\_List.csv", row.names = TRUE)

# txn = read.transactions(file="C:/Users/Admin/Desktop/Apriori/M\_wise\_List.csv",

# rm.duplicates= TRUE, format="basket",sep=",",cols=1);

# txn@itemInfo$labels <- gsub("\"","",txn@itemInfo$labels)

# basket\_rules\_m <- apriori(txn,parameter = list(sup = 0.001, conf = 0.004,target="rules"));

# View(df\_basket\_m)

# df\_basket\_m <- as(basket\_rules\_m,"data.frame")

# summary(basket\_rules\_m)

# basket\_rules\_m = subset(basket\_rules\_m, lift >= 114)

# df\_basket\_m <- as(basket\_rules\_m,"data.frame")

# View(df\_basket\_m)

# library(arulesViz)

# plot(basket\_rules\_m)

# plot(basket\_rules\_m, method = "grouped", control = list(k = 1))

# plot(basket\_rules\_m, method="graph", control=list(type="items"))

# # Artist wise

# df\_SongList\_a <- ddply(df\_SongsData,c("User.ID","Artist.Of.Song"),

# function(df1)paste(df1$Song.Title,

# collapse = ","))

# colnames(df\_SongList\_a) <- c("","","song\_title")

# write.csv(df\_SongList\_a,"C:/Users/Admin/Desktop/Apriori/A\_SongList.csv", row.names = TRUE)

# txn = read.transactions(file="C:/Users/Admin/Desktop/Apriori/A\_SongList.csv",

# rm.duplicates= TRUE, format="basket",sep=",",cols=1);

# txn@itemInfo$labels <- gsub("\"","",txn@itemInfo$labels)

# basket\_rules\_a <- apriori(txn,parameter = list(sup = 0.04, conf = 0.5,target="rules"));

# df\_basket\_a <- as(basket\_rules\_a,"data.frame")

# View(df\_basket\_a)

# plot(basket\_rules\_a)

# plot(basket\_rules\_a, method = "grouped", control = list(k = 1))

# plot(basket\_rules\_a, method="graph", control=list(type="items"))

# 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

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 data.

18 | P a g e

## CONCLUSION

As I can see from the development of song recommenders over the past years, the given results tend to be more personalized and subjective. Only considering the song itself and human ratings are no longer sufficient. A great amount of work in recent years have been done in music perception, psychology, neuroscience and sport which study the relationship between music and the impact of human behavior. Undoubtedly, music always has been an important component of our life, and now I have greater access to it. All of these highlight that music recommender is not only a tool for relaxing, but also acts as an effective tool to meet our needs under different contexts. To our knowledge, there is few research based on these empirical results. Designing a personalized music recommender is complicated, and it is challenging to thoroughly understand the users’ needs and meet their requirements. As discussed above, the future research direction will be mainly focused on user centric music recommender systems. A survey among athletes showed practitioners in sport and exercise environments tend to select music in a rather arbitrary manner without full consideration of its motivational characteristics. Therefore, future music recommender should be able to lead the users reasonably choose music. To the end, I am hoping that through this study I can build the bridge among isolated research in all the other disciplines.

## 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 songs 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 songs 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 agewise, or age and mood simultaneously, language wise and many more.

## REFERENCES

1. <http://www.iosrjournals.org/iosr-jce/papers/Vol16-issue2/Version-11/R01621195100.pdf>
2. <https://github.com/nupur1492/RProjects/blob/master/MarketBasketAnalysis/Groceries_dataset.csv>
3. <https://datascienceplus.com/implementing-apriori-algorithm-in-r/>