CSE DEPARTMENT

Big Data Project 2020

Course code: CSE412

Project name: Apriori algorithm

Team members:

Andro Morcos Naguib (BN:1500334)

Amir Ramy Zareef (BN:1500313)

Andrew Atef Fathy (BN:1500333)

Mohammed Ashraf Youssef (BN:1501158)

Table of contents

Abstract

Apriori Algorithm:
Prerequisite – Frequent Item set in Data set (Association Rule Mining)
Apriori algorithm is given by R. Agrawal and R. Srikant in 1994 for finding frequent item sets in a dataset for boolean association rule. Name of the algorithm is Apriori because it uses prior knowledge of frequent itemset properties. We apply an iterative approach or level-wise search where k-frequent item sets are used to find k+1 item sets.

Introduction

Apriori is an algorithm for frequent item set mining and association rule learning over relational databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis.

Purpose(project description)

The purpose of this project is to apply Apriori algorithm to find all the association rules in the given dataset. The dataset itself represents the customer data for an insurance company; it has 86 attributes with 5822 records, and also to find all the possible association rules for user-defined values of support and confidence. Additionally, computing the lift and leverage for each rule so that the the rules are prioritized. The dataset and the description of its attributes are available at

http://www.liacs.nl/~putten/library/cc2000/ .

List of definitions

- Unique values: frequency of single items.
- List of support: list containing all combinations of frequent items to make the process easier
- All variables and functions used are declared in their full name to ease the understanding of the code
- Apriori Property: Any subset of frequent itemset must be frequent.
- Frequent Itemset: An itemset whose support is greater than or equal to a minimum support threshold

Overview

This document will give a breif information about apriori and needed defintions to understand it and it will explain who may use and benefit from it and its main objectives and the details of the project as used environment and results from the tool written.

Beneficiaries

The main benefits of this project will indireclty effect the people , but it will directly effect huge companies in insurance industry with relations and techologies that will inspire new ways of effectiveness in production as always the human is the base but introducing ai and computers will always give a different prespective that humans may miss by giving alternative solutions and studied analysis to increase the potentails of the companies.

Project Aims and Objectives

Project Aims and Objectives are all about making better decisions, faster and with greater accuracy. Applied in the right way, letting you to see the connections between vast amounts of often disparate data and information. This is critically important in the area of fraud detection, but certainly applies to many other aspects of the insurance industry as well, from claims automation to underwriting to new product and service offerings.

Detailed Project Description

The project has a certain sequence that is :

- extracting data from the database 12 coloumns which are: MBERMIDD Middle management , MBERARBG Skilled labourers , MBERARBO Unskilled labourers , MSKA Social class A , MSKA Social class B1 , MSKA Social class B2 , MSKA Social class C , MSKA Social class D , MHHUUR Rented house , MHKOOP Home owners with 5822 different values all range from 0-9
- generating list of supports which is the frequency of each combination of attributes with multivalues which starts by making a list of unique values excluding all values that didn't pass the minimum support, then adding it to the final list and then from the list of uniques generating combination of pairs and calculating their support and excluding that didn't pass and then adding it to the final list and so on . The main idea here is generating the next combinations from the previous ones that passed the minimum support making the solution optimum.
- other functions are easier by completing the list of supports as lift and leverage are just equations calculated in simple functions, confidence is calculated by gathering the list of supports starting from the highest combinations and searching in the list of supports to calculate the support of all combinations inside the biggest ones it is made easier by the list of support and searching for values needed to calculate it

Project Phases

Phase I : creating teams,understanding project,disturbuting tasks.

Phase II : file reading , creating tables for chosen data,creating uniques list .

Phase III : completing list of supports , and other wanted functions as : confidence , lift and leverage.

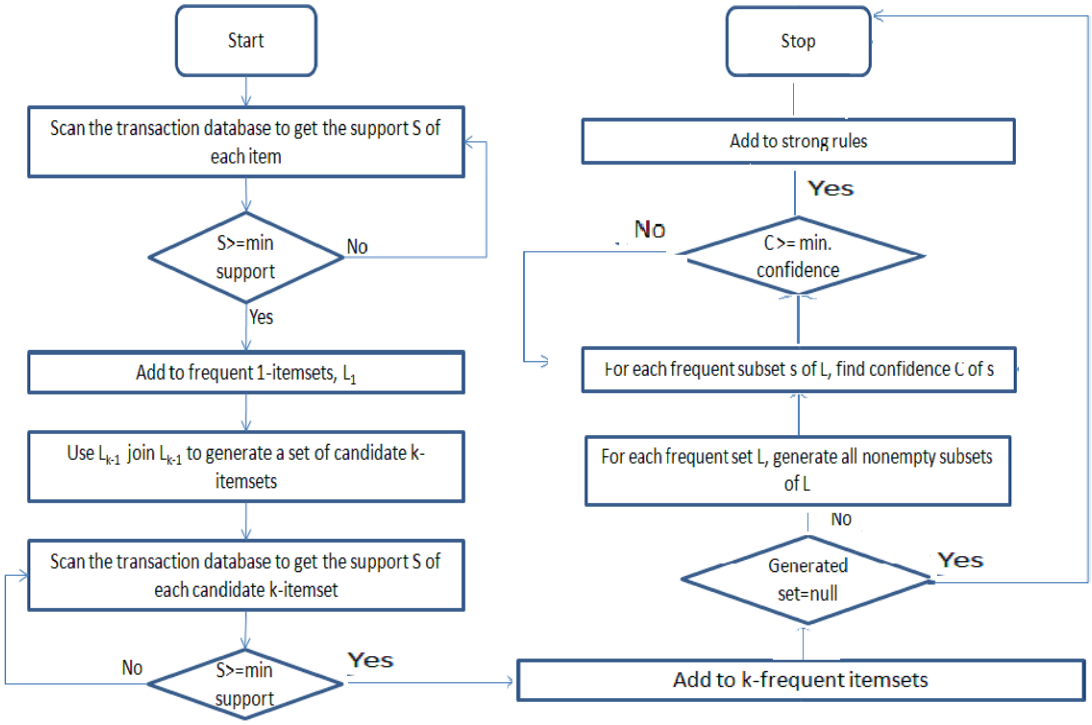
Phase IV : integrating all parts of the project together and unification of the whole file .

Phase V : testing and error handling and last editions of the project , in addition to reporting the project.

progress in project

Name	Planned time	Actual time
Phase I	2 days	1 day (2h)
Phase II	1 day	1 day (3h)
Phase III	3days	5 days (3-5h)
Phase IV	1 day	2 days (3h)
Phase V	1 day	1 day (4h)

System Architecture



Development Environment :

Language : Python (version 3.7.2)

IDE: *Jupyter Notebook* (open-source web application)

Libraries used : Pandas and itertools.

Testing Cases and Results :

Support : 20% , Confidence : 30%

No Rules

Support : 10% , Confidence : 80%

```
['MSKA Social class D_0', 'MHKOOP Home owners_9']   -->   MHHUUR Rented house_0
Confidence =  1.0
Lift =  6.134878819810326
Leverage =  0.10480440518621782
['MSKA Social class D_0', 'MHHUUR Rented house_0']   -->   MHKOOP Home owners_9
Confidence =  1.0
Lift =  6.134878819810326
Leverage =  0.10480440518621782
```

Support : 10% , Confidence :50%

```
MSKA Social class D_0   -->   ['MHHUUR Rented house_0', 'MHKOOP Home owners_9']
Confidence =  0.27963176064441886
Lift =  1.715506965723716
Leverage =  0.05222479062526073
['MHHUUR Rented house_0', 'MHKOOP Home owners_9']   -->   MSKA Social class D_0
Confidence =  0.768177028451001
Lift =  1.715506965723716
Leverage =  0.05222479062526073
MHHUUR Rented house_0   -->   ['MSKA Social class D_0', 'MHKOOP Home owners_9']
Confidence =  0.768177028451001
Lift =  6.134878819810326
Leverage =  0.10480440518621782
['MSKA Social class D_0', 'MHKOOP Home owners_9']   -->   MHHUUR Rented house_0
Confidence =  1.0
Lift =  6.134878819810326
Leverage =  0.10480440518621782
MHKOOP Home owners_9   -->   ['MSKA Social class D_0', 'MHHUUR Rented house_0']
Confidence =  0.768177028451001
Lift =  6.134878819810326
Leverage =  0.10480440518621782
['MSKA Social class D_0', 'MHHUUR Rented house_0']   -->   MHKOOP Home owners_9
Confidence =  1.0
Lift =  6.134878819810326
Leverage =  0.10480440518621782
```

Support: 5%, Confidence :60%

```
['MBERARBO Unskilled labourers_0', 'MHHUUR Rented house_0', 'MHKOOP Home owners_9']   -->   MSKA Social class D_0
Confidence =  0.964824120603015
Lift =  2.1546628424053527
Leverage =  0.03534556198954415
['MBERARBO Unskilled labourers_0', 'MSKA Social class D_0', 'MHKOOP Home owners_9']   -->   MHHUUR Rented house_0
Confidence =  1.0
Lift =  6.134878819810326
Leverage =  0.055205612608378106
['MBERARBO Unskilled labourers_0', 'MSKA Social class D_0', 'MHHUUR Rented house_0']   -->   MHKOOP Home owners_9
Confidence =  1.0
Lift =  6.134878819810326
Leverage =  0.055205612608378106
['MBERARBO Unskilled labourers_0', 'MHHUUR Rented house_0']   -->   ['MSKA Social class D_0', 'MHKOOP Home owners_9']
Confidence =  0.964824120603015
Lift =  7.705358066050417
Leverage =  0.057396865040398655
['MBERARBO Unskilled labourers_0', 'MHKOOP Home owners_9']   -->   ['MSKA Social class D_0', 'MHHUUR Rented house_0']
Confidence =  0.964824120603015
Lift =  7.705358066050417
Leverage =  0.057396865040398655
['MBERARBG Skilled labourers_0', 'MSKA Social class D_0', 'MHKOOP Home owners_9']   -->   MHHUUR Rented house_0
Confidence =  1.0
Lift =  6.134878819810326
Leverage =  0.052474084901192736
['MBERARBG Skilled labourers_0', 'MSKA Social class D_0', 'MHHUUR Rented house_0']   -->   MHKOOP Home owners_9
Confidence =  1.0
Lift =  6.134878819810326
Leverage =  0.052474084901192736
```

Earned values

- Learning how to work in teams.
- Learning new libraries in python(pandas)
- Learning how to document and write a full report.