5165 characters

Contents

[Introduction 1](#_Toc480064551)

[Dataset: Getting to know your data 1](#_Toc480064552)

[Research Questions 1](#_Toc480064553)

[Pre-Processing 2](#_Toc480064554)

[Algorithms 2](#_Toc480064555)

[Frequent Pattern Mining: Apriori 2](#_Toc480064556)

[K-Nearest-Neighbour 3](#_Toc480064557)

[K-Means 4](#_Toc480064558)

# Introduction

This report is documenting a micro-scale data mining project that was conducted within the course Data Mining. The foundation of this report is a dataset that was produced by the students of this course in the first lecture, where they were presented with a survey containing X questions. The resulting dataset was then handed over to the students to formulate research questions and conduct research using algorithms that were taught within the course. The work that was done during the lab-sessions will be used to carry out the remaining work. Similarities to the hand-ins of my group-members Ivan Mladenov, Vlad Limbean and Joachim Sogn are therefore inevitable.

# Dataset: Getting to know your data

The dataset contains all possible attribute types and has not been pre-processed before it was handed over. For this report it was not mandatory to process the entire dataset, but only a chosen set of attributes.

# Research Questions

1. For a given game X, how likely is it that another game Y is played by the same person?
2. Based on age, height and shoesize, how good are predictions towards gender?

Based on these research questions the following attributes were chosen:

Age – Gender -Shoesize – Height - Which of these games have you played?

*Consequently only this set of attributes were preprocessed and used within the algorithms.*

# Pre-Processing

This report also documents my learning-curve in java so that the things I implemented later on are more sophisticated than the things I did first, meaning that my pre-processing works but is not very elegant; I started doing most of it manually in the main method and only later on I created a pretty-maker class to handle some tasks in a more organized and efficient manner.

Data cleaning that has been conducted:

* Removing double quotes
* Replacing commas with dots
* Parsing and saving the data correctly
* Checking for missing values
  + Age:
* Gender: Takes the first letter of the answer, checks if it is “m” or “f” and if not it replaces with “m”.
* Shoesize:
* PlayedGames:
  + Splitting at every semicolon, using StringEnumerator to convert the nominal into a numeric attribute (Fifa 2017 = 0 etc.)
  + Saving numeric representation of a game in an array to allow processing in Apriori
  + Due to lack of time, I have not taken care of the option asking “I have not played any games”;

# Algorithms

## Frequent Pattern Mining: Apriori

To answer question one,

For a given game X, how likely is it that another game Y is played by the same person?

A frequent pattern mining technique called Apriori was implemented. The implementation of Apriori was quite time-consuming. The generation of candidate-1-itemset C1 and frequent-1-itemset L1 had to be done differently from the following reoccurring steps. Especially the joining was difficult because it required a lot of if-statements and nested for-loops.

The support-threshold was increased step-by-step, starting with 7 and increasing it up to 20 to actually cut down results to noteworthy and easily observable trends in the dataset.

The result shows a frequent itemset e.g. {3, 5} on the left-hand-side; each frequent-itemset is printed out twice, each referring to the likelihood of one game being played as a consequence of another.

For the first line, it reads: In the frequent-itemset {3, 5}, for game 3, the probability of students also playing game 5 is 44,68%.

For the second line, it reads: In the frequent-itemset {3, 5} for game 5, the probability of students also playing game 3 is 72,41%.



# K-Nearest-Neighbour

In order to provide an answer to question two,

Based on age, height and shoesize, how good are predictions towards gender?

I decided to implement k-nearest-neighbor, a lazy-learner classification model. For this purpose, the student-list was split into test- and trainingset. Based on age, height and shoesize, the Euclidean distance between the so called test-student and the student-for-comparison was calculated and saved. If more than half of the close-students are male, we assume the student is male. Then it is checked, if the student is actually male or not.

The results were as following:



Considering a success-rate of 70,8 % [TP + FP / total number], predictions towards gender based on age, height and shoesize are reasonable, especially considering the fact that there is not a high number of females in the dataset. When I switched to assuming females instead of males, the success rate was even higher with 83,34%. Obviously age is not the most useful attribute to use here in hindsight; upon removal the success rate was even better.

Compared to Apriori, k-nearest-neighbour is less time-consuming to implement because it requires basically only two methods for calculating the distance and making assumptions.

FP = False-Positive (Student was assumed female and is female)

FN = False-Negative (Student was assumed female but is male)

TP = True-Positive (Student was assumed male and is male)

TN = True-Negative (Student was assumed male but is female)

# K-Means