# COMP 4601 Assignment #2

# ANALYSIS

Prepared For: Anthony White

Prepared By: Sahaj Arora (100961220) and Jennifer Franklin (100315764)

Drafted: April 2, 2018

# Table of Contents

# Design Decisions

## Requirement 8 - /context

When loading /context the first-time classifications of the movie's as well as the user's community are performed. A flag is set by way of the creation of a collection called classified. It was decided that a flag would be used since the entire process takes 10+ minutes and might be interrupted. The best way to determine if the process has run to its completion is to check for a flag that is set at the end of the entire process. We assumed that a user's features would be the movies that he reviewed.

## Requirement 11

The communities found in analyzing the test data matched the categories of the movies that we classified. We classified the movies into 6 categories. Those categories being: Action, Adventure, Comedy, Drama, Horror, and Thriller. Communities were based on these movie categories, so we had a community of Action fans and a community of Adventure fans etc. In determining the user's community many design decisions were made. These decisions included:

- using weka to classify the movie pages,
- using a manual process to classify sentiment of the movie reviews,
- using collaborative filtering of each user and their friends to fill in as many blanks as possible in each user's set of reviews before classifying that specific user.

### Weka to classify the movie pages

Weka is a collection of machine learning algorithms for data mining tasks. Specifically, the Naive Bayes classifier Weka implementation was used. Five reviews for each movie category were selected for training. All of the distinct words from each of these reviews was amalgamated into one list used as the baseline for Weka. Each of the word counts of these distinct words for each review was then used as training data for Weka. Finally, Weka was then used to classify all of the other movie reviews in the database. A list of stop words, common words in the English language, were removed from all training and classifying lists used with Weka.

### Sentiment classification process

An attempt to use Weka for sentiment classification was made. It failed. In the case of sentiment, a "bag of words" approach to classification is insufficient to properly identify sentiment within each movie review's text. Too many stop words normally deemed unimportant are key in the identification of sentiment. I liked the movie. I did not like the movie. These are two very different phrases that need to be considered in full to grasp their meaning with respect to sentiment. Throwing away the common words "I", "did", "the" and "not" leave both phrases as positive in sentiment with one being liked movie and the other being like movie. In the end we used a manual process where specific phrases were identified for the purpose of classifying sentiment. Each phrase was classified as negative, neutral or

positive. The number of each phrase type was counted in each body of review text and the sentiment of the text was determined based on this count. There were approximately 11,000 reviews of 74,000 that did not match any of the phrases identified for sentiment analysis. Given more time either more phrases or a better approach should be determined. Whatever the approach taken it must be one that considers words in context, i.e., full phrases in order for the sentiment analysis to be effective to any degree at all.

### Collaborative filtering of each user and their friends

Before identifying each user's community their movie review data was filled out as much as possible using collaborative filtering with their friends' movie review data. Every movie reviewed by each user and all of their friends was read into a Hash Map. One distinct entry per unique movie title. For every movie that their friends had reviewed but that they had not reviewed, an attempt using collaborative filtering was made to predict the user's rating for that movie. The user's actual ratings as well as predicted ratings were then used to identify which community, aka, movie category they belonged to. Only movies that they rated positively were considered for this categorization. In the case where the user did not have any positive movie reviews, the user was randomly assigned to a community. This could have been done better. If they did not have any positive reviews, then a check on the number of neutral reviews might have been a better way of determining their community. Another approach that might have been better would have been to consider only those communities in which they did not have any reviews, in other words, if they watched action and did not like any of the action movies then they definitely did not belong in the action community. None of these superior approaches were implemented due to lack of time.

## SUGGEST Algorithm

1. Design and document an algorithm called ***SUGGEST*** (but do not implement it) that, given a new user, *u*, and a social graph *G=<V, E>*, where *u* is represented as a vertex along with the existing users, will suggest a set of pages that the new user might want to access. Also, indicate how the advertising system would work in this revised system. For your interest, a social graph that connects the users provided in the users directory may be found in the graph directory. This directory contains one web page per user. Each web page stores the links to other users representing a social relationship.

In the case of the SUGGEST algorithm the new user has presumably not yet reviewed any movies, BUT they have a social graph with a list of friends who also presumably have reviewed movies.

The algorithm SUGGEST could assume that they are similar to their friends. It could determine which movies were liked by the user's friends and suggest these reviews as a set of pages that the new user might want to access. This could be done using Sentiment analysis to come up

with a list of positively reviewed movies. The current application's sentiment analysis could be used although suggested improvements could be made. The user could be assigned a community based on their friends' communities. So, categorize their friends as we did using the existing process implemented in our application and if, for example, most of their friends belonged to the "Action" community, then place the new user in the "Action" community. Once the user has a list of pages and a community then the advertising system could work exactly the same as it does now. One ad specific to the user's community ("Action" in this example) and one ad specific to the page that they are viewing ("Horror" for example).