Undergraduate Students

For your final project, you are going to build a movie recommendation system. The data should be downloaded from here: The top 5000 movie data from IMDB: https://www.kaggle.com/tmdb/tmdb-movie-metadata/data (Links to an external site.). This data contains information about 5000 movies from IMDB.

Instructions for project completion

All the coding will be done in a Java environment. MySQL will only be used to store the dataset and new tables and access them for information extraction. You will give a demo of the project in class/in instrctor's office (date and location will be finalized later). You will submit EVERYTHING that the instructor needs to execute your project.

When you create the database tables with the dataset, make sure that your tables are in 1NF (see the attached video). In layman's terms, each cell in your tables will contain at most 1 data entry.

1 Keywords

To develop the recommendation engine, you should make an extensive use of the keywords that describe the films. Indeed, a basic assumption is that films described by similar keywords should have similar contents. "Keywords" is an attribute in the tmdb_5000_movies.csv file.

Create and store a list of keywords which appear at least 5 times in the database. Do not store the keywords which appear less than 5 times in the given database.

2. Genres

The genres attribute will surely be important while building the recommendation engines since it describes the content of the film (i.e. Drama, Comedy, Action, ...). "Genres" is an attribute in the tmdb_5000_movies.csv file.

3. Recommendation Engine

3.1 Architecture of the Recommender System

In order to build the recommendation engine, you will proceed in two steps:

determine films with a content similar to the entry provided by the user
select the 5 most popular films among these films.

3.1.1 Similarity Metric (Links to an external site.)

When building the engine, the first step thus consists in defining a criterion that would tell us how close two films are. To do so, you have to build a matrix where each row corresponds to a film of the database and where the columns correspond to the previous quantities (director + actors + keywords) plus the k genres that were described in the previous paragraph.

movie title	director	actor 1	actor 2	actor 3	keyword 1	keyword 2	genre 1	genre 2	...	genre k
Film 1					...					
...					...					
Film i										aiq
...					...					
Film p	ap	ap			...					apq

In this matrix, the aij coefficients take either the value 0 or 1 depending on the correspondance between the significance of column j and the content of film i.

For exemple, if "keyword 1" is in film , we will have aij = 1 and 0 otherwise. Once this matrix has been defined, we determine the distance between two films, *m and n*, according to the following function:

$$d_{m,n} = \sqrt{\sum_{i=1}^{N} (a_{m,i} - a_{n,i})^2}$$

Lower distance will indicate that the films (*m and n*) are very similar and one can be recommended for the other. You will select *N* such most-similar films for the given movie *M*.