Data Analytics Project

Jeroen Bob Clement Vermazeren, UniKey: jver9334, SID: 470513619 &
Xavier Weber, UniKey: xweb7645, SID: 470514373

October 25, 2017

Abstract

This project analyses the possibility of a renewed license purchase strategy for movies for streaming service Netflix[1]

1 Project Stage 1: Obtain data, clean it and load it

1.1 Business Understanding

1.1.1 Background

The available resources for the project is the MovieLens datasets which can be downloaded from the website of GroupLens[2]. This data-set was already used by Harper and Konstan and described in a paper. (Harper and Konstan, 2015). Further resources are the time the researcher has to investigate this case. Currently Netflix has extensive negotiations with the owners of specific content and with the information which this project could provide this negotiations could be made more easy since the owners of Netflix have a better view about the quality of the content for them.

1.1.2 Business Objectives

The goal of this investigation is improving the quality of licenses for streaming movies by Netflix based on the rating given by a collection of users. This way Netflix could be more efficient in buying licenses so people watch more of their content instead of buying licenses for movies users will not appreciate and/or watch. Business question regarding this problem are:

- Is it possible to identify groups within the complete set of all users?
- Will increasing the quality of the content according to the current users lead to an increase of the total amount of users?

1.1.3 Business Success Criteria

Objective Success Criteria

- increase of general rating of the content of Netflix by 5%
- increase of total users of 2%

Subjective Success Criteria

- discover subsets of users in the total set of users of Netflix

These objective success criteria can be checked for fulfilment by looking at the data Netflix has of their customers. The subjective success criterion has to be checked by multiple experts in the field of movies and data analytics.

1.2 Assess Situation

Inventory of Resources

Hardware

- standard computer

[1] www.netflix.com

[2] www.grouplens.org/datasets/movielens/

Data Sources

The resources used for this study are the MovieLens datasets which are downloaded from the GroupLens website, `www.grouplens.org/datasets/movielens/`. The exact name of the data-set is 'ml-latest.zip' which consists of six files written as comma-separated values[3].

- *'genome-scores.csv'*
- *'genome-tags.csv'*
- *'links.csv'*
- *'movies.csv'*
- *'ratings.csv'*
- *'tags.csv'*

Instances that contain commas (',') are escaped using double-quotes('"'). The Movielens users were selected randomly their ids are consistent between 'ratings.csv' and 'tags.csv'. Furthermore only movies with at least one rating or tag are included in the dataset. The movie ids are consistent over the MovieLens website and the files.

1.2.1 Requirements, Assumptions and Constraints

Requirements The project is divided in 4 stages:

- Project Stage 1: Obtain data, clean it and load it.
- Project Stage 2: Summarize and analyse the data.
- Project Stage 3: Develop and test a predictive model.
- Project Stage 4: presentation of result

Stage 1 has to be submitted the August 24. Stage 2 has to be submitted September 21. Stage 3 has to be submitted October 19. Stage 4 has to be submitted November 2.

Assumptions Assumed is that the data is reliable since it has been used by several researchers and origins from the year 1995 and has been updated since then.

Constraints There are no financial constraints since this is a student project although there are time constraints. The time constraint is the time the student has a week. There is only one person working on this project.

1.2.2 Risks and Contingencies

Risks There is only one risk in this case:

- the inability of the student to finish the project due to illness and other external factors.

Contingency plan Record any external factor and discuss with the appropriate persons from the university what to with possible events. This could mean extending the deadlines or other appropriate solutions.

1.2.3 Terminology

UserID

UserID refers to the ID of a user. According to MovieLens the users have been selected at random and the IDs have been anonymized. The ID are consistent over all data files.

MovieID

MovieID refers to the ID of a movie. Only movies with at least one rating are included in the data files. The MovieID's are consistent between all data files.

1.2.4 Costs and Benefits

Costs There are no costs because it is a student project.

Benefits Improving the quality of the Netflix service in general. Furthermore more users so more profit for the company.

[3] `http://en.wikipedia.org/wiki/Comma-separated_values`

1.3 Determine Data Mining Goals

1.3.1 Data Mining Goals

Cluster Goal Determine which customers are similar to each other using all the data which is related to rating and genres. Meaning that they like the same genres or specific set of movies. The clusters do not have to be the same size. The end goal is making sure that newly licensed movies are liked by the biggest amount of users of the general service Netflix.

Classification Goal Classify what movies will be liked by a cluster such that the prediction of which licenses of movies should be purchased is accurate using the data from rating and genres..

1.3.2 Data Mining Success Criteria

- Use 80% of the date to classify 20% of the data and check by the actual values if it is correct.

- Compare different intensities of clustering with each other and see which ones makes the best classification

1.4 Produce Project Plan

1.4.1 Project Plan

In the table 1 the project plan can be seen. In the table the stages are briefly described as well as the time, deadlines and risks.

1.4.2 Initial Assessment of Tools and Technique

Excel, weka[4] and possibly other tools such as Python or R.

2 Data Understanding

2.1 Collect Initial Data

2.1.1 Initial Data Collection Report

Initially the data of MovieLens seems very reliable. The data is compact so every column is interesting but the rating and genres seem to be the most interesting columns for the goals mentioned above. There is a lot of data which is very promising but because of the time and personnel constraints of this project it might be necessary to be more selective and take a smaller set of the total data.

2.2 Describe Data

2.2.1 Data Description Report

In table 2 the summary of the data can be seen.

Table 2: Data summary

Name	Size (MB)	Number of columns	Number of instances
genome-scores.csv	344.9	3	2200997
genome-tags.csv	0.018	3	1129
links.csv	0.989	3	45844
movies.csv	2.300	3	45844
ratings.csv	709.6	4	4543170
tags.csv	27.10	4	753171

Data Structure *’genome-tags.csv’* and *’genome-scores.csv’*
This is a copy of the current Tag Genome data (Vig et al., 2012). The file *’genome-scores.csv’* contains the movie-tag relevance data in following format: *movieId(numeric), tagId(numeric), relevance(numeric)*. The file *’genome-tags.csv’*, gives the tag description for the tag IDs in the following format: *tagId, tag*. Relevance is a score which indicates the relevance of the tag next to the movie. (Vig et al., 2012)

Data Structure of *’links.csv’*
The file *’links.csv’* is used as identifier to link to other sources of movie data. Each line represents one movie and has the format: *movieId(numeric), imdbId(numeric), tmdbId(numeric)*. *movieId* is an identifier for movies used by MovieLens. *imdbId* is an

[4] `www.cs.waikato.ac.nz/ml/weka/`

Table 1: Project plan

Stage	Description	Time	Deadline	Risks
1	Obtain data, clean it and load it	1 week	24 August	student illness
2	Summarize and analyse the data	4 weeks	11 September	student illness, data incomplete
3	Develop and test predictive model	4 weeks	19 October	student illness, insufficient skill
4	Presentation of results	4 weeks	2 November	student illness

identifer for movies used by `www.imdb.com`. tmdbId is an identifier for movies used by `www.themoviedb.org`.

Data Structure of *'movies.csv'*
Each line of this file represents one movie with format: *movieId(numeric), title(string), genres (categorical).* The list of genres:

- action
- adventure
- animation
- children's
- comedy
- crime
- documentary
- drama
- fantasy
- film-noir
- horror
- musical
- mystery
- romance
- sci-fi
- thriller
- war
- western
- (no genres listed)

Data Structure of *'ratings.csv'*
The ratings are contained in the file *'ratings.csv'*. Each line represents one rating of one movie by one user with the following format: *userId(numeric), movieId(numeric), rating(numeric), timestamp(seconds).* Rating is a score between 0.5 and 5 stars with half star increments. Timestamp is the number of seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970.

Data Structure of *'tags.csv'*
All tags are contained in the file *'tags.csv'*. Each line represents one tag applied to one movie by one user, and has format: *userId(numeric), movieId(numeric), tag(string), timestamp(seconds).* Tags are user-generated metadata about movies. Each tag is typically a single word of short phrase. The meaning, value, and purpose of a particular tag is determined by each user. Timestamp is the number of seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970.

2.3 Explore Data

2.3.1 Data Exploration Report

Hypothesis about the Data Quality The data has been used in a previous research therefore the data is expected to be of good quality. (Vig et al., 2012) Furthermore it is a big data set which can be seen as positive since it could make the result more complete.

Promising Attributes

'genome-scores.csv' The file has three attributes namely, 'movieId','tagId' and 'relevance'. 'movieId','tagId' are not interesting variables since these are just references to both the movies and the tags. The only promising attribute in this file is 'relevance'. This attribute could help clustering movies together and clustering users together.

'genome-tags.csv' The file has two attributes, 'tagId' and 'tag'. This file only makes a link between the actual string value 'tag' and its index 'tagId'. This means that this file does not have any promising attributes.

'links.csv' The file has three attributes, 'movieId','ImdbId' and 'tmdbId'. The file only makes a link between the 'movieId' attribute and the other attribute 'ImdbId' and 'tmdbId'. These attributes are relevant for interpreting the final results but might be unnecessary during the research.

'movies.csv' The file has three attributes, 'movieId', 'title' and 'genres'. The attribute 'genres' could be valuable and promising for the both the data mining goals. This attribute is promising for the cluster goal because an user could prefer certain genres and so it would be able to cluster users based on their preferences for specific genres.

'ratings.csv' This file is probably the most valuable for this research since it contains all the info about the users preferences. 'ratings.csv' has 4 attributes, 'userId', 'movieId', 'rating' and 'timestamp'. The attribute 'timestamp' does not seem promising or valuable since it is only a indication of time and therefore removing this attribute is considered. The attributes 'userId' and 'movieId' connect the movies and users to each other, therefore the attributes are relevant for interpreting the final results. The attribute 'rating' is very promising for this research as it represents how much a user likes a movie and the attribute will be relevant for both the cluster goal and the classification goal. 'rating' is valuable for the cluster and classification goal because it gives the possibility to make a set of movies a user likes.

'tags.csv' This file has four attributes, 'userId', 'movieId','tag' and 'timestamp'. The attributes 'userId' and 'movieId' are relevant for interpreting the final results but do not contain any other information. The attribute 'tag' contains the string value for what tag a user gave to a certain movie. This data might be irrelevant because it might be irrelevant who gave which tag to which movie however this should be done with a lot of care since there might be some information about the user in this file.

Explored Characteristics of the Data

The exploration of the data until now gave the following insights:

Low number of tags assigned to some movies Some instances in the file 'genome-scores.csv' might be irrelevant because they have to few tags assigned to them so this might have to be cleaned.

Similarity of tags After inspecting the file 'tags.csv' there can be noticed that there are a of of similar tags so cleaning this by compressing some similar tags together would be a good idea since interpreting the data will be easier this way and some tags might provide more information.

Low number of ratings of a movie The file 'ratings.csv' has some movies that have not been rated by a lot of people therefore it might be useful to remove these instances.

Valuable users The file 'tags.csv' contains information about how many times a user gave a tag to a movie. This amount of tags that a user gave to distinct movies might give information about the user and we might consider these users valuable since they have seen a lot of movies and therefore provide a wide spectrum of data.

Revising Data Mining Goals

The data mining goals are still valid.

2.4 Verify Data Quality

2.4.1 Data Quality Report

Missing Data
All files have been checked in RStudio with the function 'na.omit(filename'), where filename is the name of the file. If there were missing values then the instance have been deleted

- *'genome-scores.csv'* No missing values therefore no instances have to be deleted
- *'genome-tags.csv'* No missing values therefore no instances have to be deleted
- *'links.csv'* No missing values therefore no instances have to be deleted
- *'movies.csv'* No missing values therefore no instances have to be deleted
- *'ratings.csv'* No missing values therefore no instances have to deleted
- *'tags.csv'* 16 instances can be deleted because they did not have a value in the 'tag' attribute and so they are not relevant.

Data Errors There are some tags which are wrong or useless inputs of the users. In 'tags.csv' there are values of attribute 'tags' which are equal to ':)' these can be deleted since they have no semantic meaning.

Low Occurrence of Movies, Tags or Users
Some tags, movies and users do not occur often in the data sets and therefore these instances might not provide any information to this research so these instances might have to be deleted to increase the quality of the research.

3 Data Preparation

3.1 Select Data & Cleaning Data

3.1.1 Data Cleaning Report

Missing Data 16 instances have been deleted in the data set 'tags.csv' because the file had NA/null values.

Data Errors Irrelevant tags like ':)' have been deleted in data 'tags.csv'.

Low Occurrence of Movies, Tags or Users
Movies and tags have been deleted to decrease the sizes of the data set when it did not occur enough. Furthermore some users also have been deleted for the same reasons.

3.1.2 Rationale for Inclusion or Exclusion

For every file is explained why some instances or attributes have been included or excluded.

'genome-scores.csv' In this research an user which only tagged less then 50 movies is seen as an invaluable user and therefore has to be deleted. This measure is taken to reduce the size of the data set and only take into account users that tagged more movies which could be seen as more valuable. Furthermore some movies can be deleted for the similar reasons. A movie that which has less than 100 tags is seen as invaluable and therefore has to be deleted. Because of these measures the number of instances in *'genome-scores.csv'* has been decreased by 47.5%. These bars, 50 for users and 100 for movies are set by estimation so the might have to be changed if necessary.

RStudio code used for exclusion of users:

```
  genome-scores<-genome-scores[df$userId
%in% names(table(genome-scores$userId))
[table(genome-scores$userId) >= 50],]
```

RStudio code used for exclusion of movies:

```
  genome-scores<-genome-scores
[genome-scores$movieId %in%
names(table(genome-scores$movieId))
[table(genome-scores$movieId) >= 100],]
```

'genome-tags.csv' All instances of this file can be included since file is compact and does not have duplicates.

'links.csv' This file will probably not be used for the research since its just a meta data file. Therefore the file can be left untouched and excluded in the research but it might be included in the interpretation of the results.

'movies.csv' This is a meta data set so nothing has to be deleted.

'ratings.csv' For this file, 'ratings.csv', a similar measure has to be taken as for the 'genome-scores.csv'. Decided is that all users that rated less than 50 movies are deleted and all movies than have been rated less then 10000 times are deleted as well. The data set will be stronger for the data mining goals because of this measure. A user that rated less than 50 movies seems to be invaluable and a movie that is rated less than 10000 times seems to be invaluable too. Although later findings could adjust these bars. Furthermore the attribute 'timestamp' is not relevant for this research so this can be deleted. These measured resulted in a decrease of instances of 61.3%.

RStudio code used for exclusion of users:

```
  ratings<-ratings[ratings$userId
%in% names(table(ratings$userId))
[table(ratings$userId) >= 50],]
```

RStudio code used for exclusion of movies:

```
  ratings<-ratings[ratings$movieId
%in% names(table(ratings$userId))
[table(ratings$movieId) >= 10000],]
```

'tags.csv' This file is a meta data file containing all the tags given by all the users. This can be left intact because its needed for reference although the attribute 'timestamp' can be deleted since it is not relevant for this research.

3.2 Construct Data

3.2.1 Derived Attributes

'movies.csv' This file has to be extended since all genres are stored in one cell. Every genre has to be stored as one instance to be useful in the future models. Splitting the attribute genres increased the number of instances by 87.6%.

RStudio code used for creating new attributes:

```
  s <- strsplit(newmovies$genres, split =
"\\|")
newmovies <- data.frame(movieId=
rep(newmovies$movieId, sapply(s,
length)),title = rep(newmovies$title,
sapply(s, length)), genre = unlist(s))
```

3.2.2 Generated Record

3.3 Integrate Data

Merging and appending are not necessary for this data set because it is already a compact data set. Merging or appending would make the data set ambiguous.

3.4 Format Data

In order to use the K-means algorithm, the data has to be formatted. First every movieId instance needs to have the genres as attributes with binary values. Value 1 if that movie has that certain genre, and 0 if it does not. Now K-means can be applied, to cluster the movies. In this case, the optimal number of movie clusters is three, as computed by the elbow method.

In order to cluster the users, we need to format the data again. For each user, their mean rating per movie cluster was calculated. All users with mean ratings lower than 4 were omitted, as only high mean ratings are interesting since the goal is to find clusters with users who have interests (so high ratings) in similar movies. So the final dataset for clustering users is a data frame with 3 columns: UserId, ClusterId, MeanRating.

Table 3: Example KMeans Dataset

movieId	adventure	animation	childrent	comedy
1	1	1	1	1
2	1	0	1	0
3	0	0	0	1
4	0	0	0	1

4 Modelling

4.1 Select Modelling Techniques

Modelling Technique The modelling techniques that are used:

- KMeans
- Elbow Method to determine clusters
- Principal Component Analysis

Modelling Assumptions

KMeans Models To make the dataset suitable for a K mean clustering model performed is the PCA analysis. The original dataset is transformed into a dataset where all genres have been transformed to columns with binary values. In table 3 can be seen how this dataset looks like.

Elbow Method to Determine Clusters K is 2 until 15 are considered as potential K's. For the elbow method no other assumptions are made.

Principal Component Analysis No other assumptions are made for the Principal Component Analysis. The first 2 components are used to visualize the data.

4.2 Generate Test Design

Test Design

Partitioning of the Dataset Since the dataset is huge it has to be made even smaller to get any results. In future cases more powerful computers and more time can be used to do it on a larger dataset. The partitioning that has been done is instead of using all users only the first 10000 users are used in the 'ratings.csv' dataset.

Test and Train Data To train 66% of the data is used for the KMeans model and 33% is left for future use of validating the model and explaining the model.

Measurement of Success of Models In this case it is fairly difficult to test the success of a model since the model has first to be applied to a real world scenario to define the results. Although the preferable results are that there are distinct clusters which have different amount of users so it is easy to see if a movie will be liked by a lot of users or a small amount of users.

Rerunning the Models For this research it is possible to rerun the model with different sizes of datasets and different parameters. It will be necessary to decide over time which model has the best outcome and should be applied. It might be interesting to decide to discard the elbow method to make sure there are more clusters. More clusters could mean that users are better divided and make better partitions for the goals of this research.

4.3 Build Model

Parameter Settings Models For the PCA, the "princomp" function in R is used. The inputted parameter is a 66%-33% partitioned dataset and is run with correlation as true.

To find out the optimal number of clusters for the K-means algorithm, the elbow method was applied. For every k the sum of squares is calculated and this is plotted afterwards in a graph. In the graph can be seen where the 'elbow' is and so the k can be determined.

In this case, the optimal number turned out to be three, so 3 was given as parameter in the K-means algorithm.

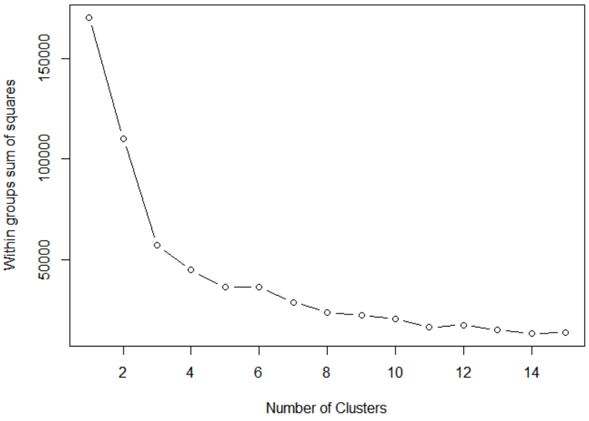

Figure 1: Elbow Method, k=3

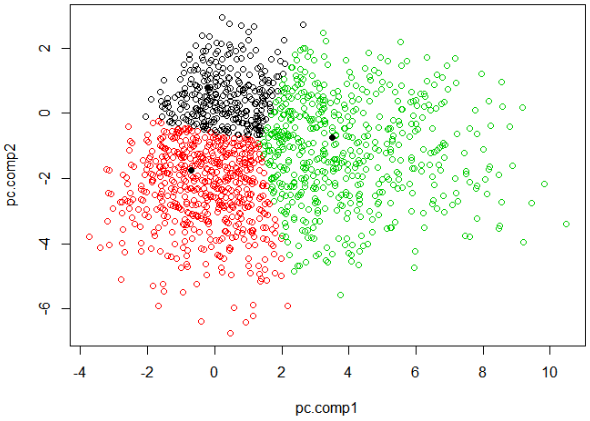

Figure 2: KMeans clustering

Models Produced To determine the number 'k' of clusters the 'elbow method' has been used. In graph 1 the sum of squares is compared with the number of clusters. In graph 1 can be seen that the desired number of clusters according to 'elbow method' is 3.

The KMeans model that is produced has 3 clusters. Cluster 1 has 3254 movies in its cluster. Cluster 2 has 2635 movies in its cluster. Cluster 3 has 2906 movies in its cluster. The clusters can be plotted in respect to the first two components of the Principal Component Analysis. This plot can be seen in graph 2

Model Description

Meaningful conclusions The clusters will have different sizes. The cluster with the most amount of users in it will be the most popular cluster. If only a small subset of instances are used in the model, the difference in size of the three cluster is minimal. No difference in size means that every cluster is as popular as the other and so will provide no meaningful results. This problem goes away once the number of instances used are increased. The result is that the difference in size of the clusters increases and so a very popular cluster emerges. Netflix can then put their potential movies in the corresponding clusters. If one of these movies falls in the popular cluster, Netflix has found their next movie which will be liked by the largest number of users. According to the model that movie will be the most popular and so will make the most profit. If Netflix its case only satisfying 1% more could mean an increase in monthly subscription and a decrease in unsubscription which could potentially lead to large profit increase.

Execution Problems In the current algorithm it is very computationally expensive to compare the clusters with the 'ratings.csv' dataset since the algorithm has to loop through the table to find the similar id's. There are more than 100.000 instances in 'ratings.csv' so due to time and computational power constraints it is impossible to handle all ratings. Therefore only a select amount 'ratings.csv' are evaluated, namely 10.000.

4.4 Model Assessment

Review of result based on business problem Since there are 3 distinct clusters with different sizes, Netflix is able to put their potential movies in these clusters. Netflix can then view which movie(s) will be the most popular and then buy the license of that(/those) movie(s). This will increase the popularity of their site, which will attract more users and increase the rating of the content of Netflix from current users. So in the case of Netflix having 1 million users, if the number of users is increased by 1%, 10,000 people will have joined their site. Meaning $ 100,000 more revenue per month if the price for a subscription would be 10$ per month.

Deployability The model is easy to deploy since it can be run in the background. Netflix could reconsider the initial thoughts about a potential new movie they want to buy by comparing the initial thoughts with the model. If the model seems to be off this can be send as feedback to the data analyst and he can easily change the parameters of the model. Over time Netflix can decide if the model is accurate.

Impact of Results on Success Criteria It is not known what impact the results have on the success criteria. This can be determined after the model is deployed.

Revised Parameters In future research it might be valuable to have a look at different k's for the KMeans algorithm which are not determined by the 'elbow method'.

5 Evaluation

5.1 Evaluate Results

Clear visualization of Result In the visualization of our result in 4 easily can be seen that choosing a movie which is in cluster 2 satisfies 19% less users than choosing a movie which is in cluster 1 which could potentially make a huge difference in terms of profit. The same holds for choosing a movie in cluster 3 which has satisfy 12% less users than cluster 1.

Assessment of Data Mining Results w.r.t. Business Success Criteria Clusters are found, and new movies can be put in these clusters. The model then shows which movie will be the most popular. However, this is only a prediction and might not result in the desired outcome. That is, to increase the number of users of Netflix and increase the general rating of the site.

Approved Models

- K-Means clustering for Movies based on genres
- Clustering based ratings

5.2 Review of Process

Phase 1 Determining realistic goals and criteria can be difficult because the goals and criteria set in this research cannot be satisfied without running the models first. In a future project it might be wise to also decide on criteria and goals which can be satisfied without running the model on a real world scenario.

Phase 2 Excluding data is essential for clarity. In this case too many data was considered so that it was hard to keep overview of the process. Although for this project the dataset was already made smaller, this in the end did not seem sufficient and advised is decreasing the dataset even more.

Phase 3 Formatting the data is paramount for successfully implementing the algorithms to produce models. There is a learning curve to R which will make this phase in future projects more streamlined. Other models, especially Collaborative Filtering, were planned to be implemented but due to time constraints and the learning curve of the data mining tools these were not completed and a different approach was chosen.

Different Strategies A future project should again research Collaborative Filtering and discover if this is a better model than the current model. Collaborative Filtering seems more suitable for the user clustering problem.

Table 4: Result of Clustering Users

Cluster	**Number of Users**	**Difference with Best Cluster(# of users)**	**Difference with Best Cluster(%)**
1	3254	0	0%
2	2635	619	-19%
3	2906	348	-12%

5.3 Determine Next Steps

The model was successful and addresses the data-mining and business goals. It is not sure whether these results are optimal, so the model could be re-implemented with different parameters to potentially find better results. One parameter that could be tweaked is the number of clusters in the K-means algorithm.

List of Possible Actions

- Research Collaborative Filtering
- Researching different K's for the K-Means algorithm
- Using more computational power and time to evaluate the whole dataset.

A Example of Data Files

A.1 *'genome-scores'*

Table 5: *Example of 'genome-scores'*

movieId	tagId	relevance
1	1	0.02475
1	2	0.02475
1	3	0.049
1	4	0.0775
1	5	0.1245
1	6	0.23875
1	7	0.06575
1	8	0.28575
1	9	0.254
1	10	0.02625

A.2 *'genome-tags'*

Table 6: Example of *'genome-tags'*

tagId	tag
1	7
2	007 (series)
3	18th century
4	1920s
5	1930s
6	1950s
7	1960s
8	1970s
9	1980s
10	19th century

A.3 *’links’*

Table 7: Example of *’links’*

movieId	imdbId	tmdbId
1	114709	862
2	113497	8844
3	113228	15602
4	114885	31357
5	113041	11862
6	113277	949
7	114319	11860
8	112302	45325
9	114576	9091
10	113189	710

A.4 *’movies’*

Table 8: Example of *’movies’*

movieId	title	genres
1	Toy Story (1995)	Adventure
2	Jumanji (1995)	Adventure—Children
3	Grumpier Old Men (1995)	Comedy—Romance
4	Waiting to Exhale (1995)	Comedy—Drama—Romance
5	Father of the Bride Part II (1995)	Comedy
6	Heat (1995)	Action—Crime—Thriller
7	Sabrina (1995)	Comedy—Romance
8	Tom and Huck (1995)	Adventure—Children
9	Sudden Death (1995)	Action
10	GoldenEye (1995)	Action—Adventure—Thriller

A.5 *’ratings’*

Table 9: Example of *’ratings’*

userId	movieId	rating	timestamp
1	110	1	1425941529
1	147	4.5	1425942435
1	858	5	1425941523
1	1221	5	1425941546
1	1246	5	1425941556
1	1968	4	1425942148
1	2762	4.5	1425941300
1	2918	5	1425941593
1	2959	4	1425941601
1	4226	4	1425942228

A.6 *’tags’*

Table 10: Example of *’tags’*

userId	movieId	tag	timestamp
1	318	narrated	1425942391
20	4306	Dreamworks	1459855607
20	89302	England	1400778834
20	89302	espionage	1400778836
20	89302	jazz	1400778841
20	89302	politics	1400778841
20	96079	nostalgic	1407930249
20	113315	coming of age	1407837917
20	113315	dark comedy	1407838006
20	113315	kafkaesque	1407837913

B Link to Dataset

- `http://grouplens.org/datasets/movielens/latest/`

References

Harper, F. M. and Konstan, J. A. (2015). The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19.

Vig, J., Sen, S., and Riedl, J. (2012). The tag genome: Encoding community knowledge to support novel interaction. *ACM Trans. Interact. Intell. Syst.*, 2(3):13:1–13:44.