

Machine Learning

*Master of Arts in Banking and Finance*

*Spring term 2020*

**Development of a movie recommendation engine using demographic, content-based and collaborative filtering … methods**

Term paper

Nils Kaufmann | nils.kaufmann@student.unisg.ch | xy

Yifan Zhu | yifan.zhu@student.unisg.ch | xy

Jan Scheidegger | jan.scheidegger@student.unisg.ch | xy

Daniel Rüger | daniel.rueger@student.unisg.ch | 09-593-377

submitted on 14 May 2020

Prof. Dr. Anna-Lena Horlemann

**Abstract**

Today platforms such as Netflix or Amazon offer tens of thousands of films which can be streamed online. In view of this huge amount, the user faces an information overload problem which makes choice of a movie suited to his interests and needs time consuming and complicated. In order to increase convenience and quality of movie selection we present three approaches of moving recommendation system using machine learning algorithms. [*To be improved:* The implemented demographic filtering technique allows to suggest movies for unknown users, while the two content-based techniques we implement make use of the past consumer behavior. With a collaborative-filtering method, we exploit comparisons to other consumers and finally,..]

**Tbd by Daniel:**

* References, quotations
* List of references

**List of figures**

**Table of contents**

[1 Introduction 3](#_Toc42359297)

[2 Data 5](#_Toc42359298)

[3 Application 5](#_Toc42359299)

[4 Conclusion 5](#_Toc42359300)

[5 List of references 5](#_Toc42359301)

[6 Appendix 5](#_Toc42359302)

# Introduction

The data available in the internet is constantly increasing at fast pace. Consumers informing themselves in the internet, are confronted with a large amount of information which can easily become overstraining (016555). Information overload can be the consequence (Design, p. 1) Therefore, consumers need to invest much time and resources in order to analyze the information and choosing a product can become complicated (as citet in 016555). A field which is affected by this evolution is movie selection (Design, p. 1). Platforms such as Netflix or Amazon offer a huge number of movies which customers can stream online. (Improved Collaborative, p. 1). Therefore, for consumers, it is difficult to identify the product best suited to their needs.

Information overload can be reduced by applying recommendation systems, which screen the available information. These algorithms allow to individualize recommendations considering consumer characteristics such as personal taste and needs. Regarding a large number of subjects, recommendation systems have been successfully implemented in order to improve recommendation results, e.g. movies, music, news, tourism (as cited in, 07816, p. 1). Netflix for example considers the films a client has consumed and suggests further movies which are comparable to them (content-basedMovieRecommendation).

Various methods can be used to implement a recommendation system. Three important categories are content-based, collaborative and hybrid algorithms. Content-based systems analyze the user’s past consumption behavior and suggest movies based thereon. Collaborative filtering examines the past ratings and experiences of consumers and compares it with further clients. Based on the choices of the users being the most comparable suggestions are generated. In order to avoid the individual drawbacks of both methods, hybrid models have been proposed. (Conent-baesdMovieRecommendation)

The aim of this paper is to develop a movie recommendation system using machine learning algorithms. Our product shall help consumers save time and resources invested in choosing a movie, improve convenience of the process and allow them to find a film better suited to their tastes and needs.

We propose three different approaches to overcome the disadvantages of the individual methods. At first, we implement a demographic filtering technique, then content-based and finally collaborative filtering algorithms. The demographic filtering method allows to suggest films for persons with unknown characteristics. As content-based methods we apply a k-NN algorithm as well as a plot-based recommender. Both exploit the consumer history of the user. In order to make possible comparison with experiences of other consumers we implement the collaborative filtering method. Finally, a decision tree algorithm is used because… and why the last algorithm.,,

Different data sets regarding movies exist and could be used to develop a movie recommendation system. We have chosen the TMDB (The Movie Database) 5000 movie dataset of kaggle because it contains many variables such as revenue and ratings. This allows a large variety of machine learning algorithms to be implemented. TMDB 5000 consists of XY individual movies. An alternative would have been the IMDB Movie dataset. However, as it has over 40 characteristics for every individual film it would have become too complicated to clean and prepare the data.

The further parts of this paper are structured as follows. At first, we describe the dataset in more detail. In the application section that follows, we describe the techniques used and their actual implementation. Finally, we end with the results and conclusion.

Due to the limited scope of this article, we will not test all the predictions made by our recommendation (Question Daniel: Is that correct?). For the same reason we neither merge the three approaches in order to receive a single recommendation. Instead, the user should assess the three individual propositions the algorithms deliver him on his own.

Many papers which present movie recommendation systems using machine learning methods have been published in recent years. A k-NN collaborative filtering algorithm was for instance used by Cui Bei-Bei (Deisgn\_and). S. Reddy et al. applied a content-based method based on genre correlations (Content-basedmovie). Wang, Sang, Zeng and Hirokawa have implemented a support vector machine method and improved particle swarm optimization (E100.D).

# Data

# Application

# Conclusion

* Test the predictions and compare directly (in part we have done this as far as I have seen)
* In future works, the three methods we proposed could be combined and their outcome weighted (give points) in order to receive one recommendation based on all of the three algorithms.
* Further methods as support vector machine could be implemented, also more elaborated/complex methods?

# List of references

# Appendix