

DreamSwipe Tinder für Filme

Leon Gieringer, Robin Meckler, Vincent Schreck

Studienarbeit

10. Februar 2021

Inhaltsverzeichnis

1	Einleitung	1
2	Motivation	1
3	Theoretische Grundlagen	1
3.1	Framework	1
4	Einleitung	1
4.1	Language	1
4.2	IDE	1
4.3	Database	1
4.4	Firebase	1
4.5	Recommendationssystem	1
4.5.1	Collaborative Filtering	2
4.5.2	Content-based filtering	2
4.5.3	Cold Start Problem	2
5	Konzept?	3
6	Funktionen/Komponenten	3
6.1	Swipe/Aussuchen/Voting	3
6.2	Matches/Chat	3
6.3	Film-/Serienvorschläge	3
6.4	Gruppenorgien	3
6.5	Gespeicherte Filme/Filmliste	3
6.6	Zugänglichkeit/Behindertenfreundlichkeit	3
7	Benutzeroberflächen	3
7.1	Home-Screen	3
7.2	Gruppen	3
7.3	Chat	3
7.4	Filmliste	3
8	CodeBeispiele	3
9	Probleme	3
10	Fazit	3

1 Einleitung

2 Motivation

3 Theoretische Grundlagen

3.1 Framework

4 Einleitung

4.1 Language

Hier steht mein Language Text.

4.2 IDE

Hier steht mein IDE Text.

4.3 Database

Hier steht mein Database Text.

4.4 Firebase

Hier steht mein Firebase Text.

4.5 Recommendationssystem

Auf der Webseite Youtube allein werden minütlich mehr als 500 Stunden Videomaterial hochgeladen. (<https://blog.youtube/press/>, 10.02.2021) Um bei einer solch unvorstellbaren Menge an Daten (allein auf einer Webseite) den Überblick als Endnutzer behalten zu können, ist ein personalisierten Filtersystems unausweichlich.

Solche Filtersysteme, auch Recommendation System genannt, nutzt bisher gesammelte Daten um Nutzern potentiell interessante Objekte jeweils individuell vorzuschlagen. Ein sogenannter *Candidate Generator* ist hierbei ein Recommendation System, welches die Menge M als Eingabe erhält und für jeden Nutzer eine Menge N ausgibt. Hierbei umfasst M alle Objekte und gleichzeitig gilt $N \subset M$.

Die Bestimmung einer solchen Menge N beruht grundlegend auf zwei Informationsarten. Erstens die sogenannten Nutzer-Objekt Interaktionen, also beispielsweise Bewertungen oder auch Verhaltensmuster; Und zweitens die Attributwerte von jeweils Nutzer oder Item, also beispielsweise Vorlieben von Nutzern oder Eigenschaften von Items. (Charu C. Aggarwal - Recommender Systems) Systeme, welche zum Bewerten ersteres benutzen, werden *collaborative filtering* Modelle genannt. Andere, welche zweiteres verwenden, werden *content-based filtering* Modelle genannt. Wichtig hierbei ist jedoch, dass *content-based filtering* Modelle ebenfalls Nutzer-Objekt Interaktionen (v.a. Bewertungen) verwenden können, jedoch bezieht sich dieses Modell nur auf einzelne Nutzer - *collaborative filtering* basiert auf Verhaltensmustern von allen Nutzern bzw. allen Objekten.

Ein solches Recommendation System kann im einfachsten Fall wie in 4.5 als Matrix dargestellt werden.

		Items					
		1	2	...	i	...	m
Users	1	2		1			3
	2	4			5		
	...			1			4
	u		4		5		1
		2				3	
	n		4		3		

Tabelle 1: Nutzer-Item Matrix mit Bewertungen. Jede Zelle $r_{u,i}$ steht hierbei für die Bewertung des Nutzers u an der Stelle i

4.5.1 Collaborative Filtering

Unter *collaborative filtering* versteht man das Betrachten von Ähnlichkeiten im Verhalten von Nutzern anhand von Bewertungen und Präferenzen, bzw. anhand der Ähnlichkeiten von Objekten.

Diese Art leidet sehr unter dem *sparsity* Problem, also dass die Nutzer zu wenige Bewertungen von Objekten ausüben. Daher sind Vorhersagen über Ähnlichkeit von Nutzern aufgrund unzureichender Datensätze nicht sinnvoll möglich (siehe 4.5.3).

Generell unterscheidet man in zwei Typen:

1. *Memory-based Methoden*: Es wird, wie oben beschrieben, aus gesammelten Daten Ähnlichkeit herausgearbeitet und Nutzer-Objekt Kombinationen durch eben diese vorhergesagt. Daher wird dieser Typ auch *neighborhood-based collaborative filtering* genannt. Man unterscheidet weiter in:
 - (a) *User-based*: Ausgehend von einem Nutzer A werden Nutzer mit ähnlichen Nutzer-Objekt Kombinationen gesucht, um Vorhersagen für Bewertungen von A zu treffen. Ähnlichkeitsbeziehungen werden also über die Reihen der Bewertungsmatrix berechnet.
 - (b) *Item-based*: Hierbei werden ähnliche Objekte gesucht und diese genutzt um die Bewertung eines Nutzers für ein Objekt vorherzusagen. Es werden somit Spalten für die Berechnung der Ähnlichkeitsbeziehungen verwendet.
2. *Model-based Methoden*: Machine Learning und Data Mining Methoden werden verwendet um Vorhersagen über Nutzer-Objekt Kombinationen zu treffen. Hierbei sind auch gute Vorhersagen bei niedriger Bewertungsdichte in der Matrix möglich.

(Charu C. Aggarwal - Recommender Systems)

4.5.2 Content-based filtering

4.5.3 Cold Start Problem

<https://dl.acm.org/doi/pdf/10.1145/3383313.3412488> <http://www.microlinkcolleges.net/elib/files/undergraduate/Photography/504703.pdf>

kann man das als Quelle angeben??

5 Konzept?

6 Funktionen/Komponenten

6.1 Swipe/Aussuchen/Voting

6.2 Matches/Chat

6.3 Film-/Serienvorschläge

6.4 Gruppenorgien

6.5 Gespeicherte Filme/Filmliste

6.6 Zugänglichkeit/Behindertenfreundlichkeit

7 Benutzeroberflächen

7.1 Home-Screen

7.2 Gruppen

7.3 Chat

7.4 Filmliste

8 CodeBeispiele

9 Probleme

10 Fazit