



Универзитет Св. Кирил и Методиј - Скопје  
Факултет за информатички науки и  
компјутерско инженерство



Проектна задача по предметот

## **Мултимедиски Мрежи**

# Системи за препораки *(прилагодени за видео содржини)*

---

Ментор

Доц. д-р Сашо Граматиков

Изработил

Ана Мангаровска 171175

Скопје, Август 2023

**Содржина :**

<b>1. Вовед.....</b>	<b>3</b>
<b>2. Што се системи за препорака ?.....</b>	<b>3</b>
<b>3. Типови системи за препораки.....</b>	<b>3</b>
3.1. Персонализирани системи.....	4
3.1.1. Колаборативно филтрирање ( Collaborative Filtering ).....	4
1.1.1.1. Модел-базирани ( Model based ).....	4
1.1.1.2. Мемориско-базирани ( Memory based ).....	5
1.1.1.1.1. Базирани на корисник ( User based ).....	5
1.1.1.1.2. Базирани на ставка ( Item based ).....	5
3.1.2. Содржинско-базирани ( Content-based Filtering ).....	6
3.1.3. Хибридните системи за препораки.....	6
3.2. Не персонализирани.....	6
3.2.1. Базирани на популарност.....	6
3.3. Дополнителни категории.....	6
3.3.1. Демографско-базирани.....	6
3.3.2. Utility-базирани.....	7
3.3.3. Системи базирани на знаење.....	7
3.3.4. Филтрирање врз база на контекст.....	7
<b>4. Пример на познат систем за препораки.....</b>	<b>7</b>
4.1. Netflix.....	7
<b>5. Практичен пример ( систем за препорака на филмови ).....</b>	<b>10</b>
<b>6. Дополнителна активност.....</b>	<b>15</b>
<b>7. Заклучок.....</b>	<b>17</b>
<b>8. Користена литература.....</b>	<b>18</b>

## 1. Вовед

Во денешната дигитална доба експоненцијалниот раст на видео содржини на онлајн платформите (YouTube, Netflix, Amazon...) доведува до значителен предизвик во асистенција на корисниците да ги откријат релевантните содржини од огромниот достапен избор. Системите за препораки се појавуваат како клучно решение за справување со овој предизвик, нудејќи персонализирано корисничко искуство кое ја зголемува интеракцијата и задоволството на корисниците, заштедувајќи им многу време и енергија.

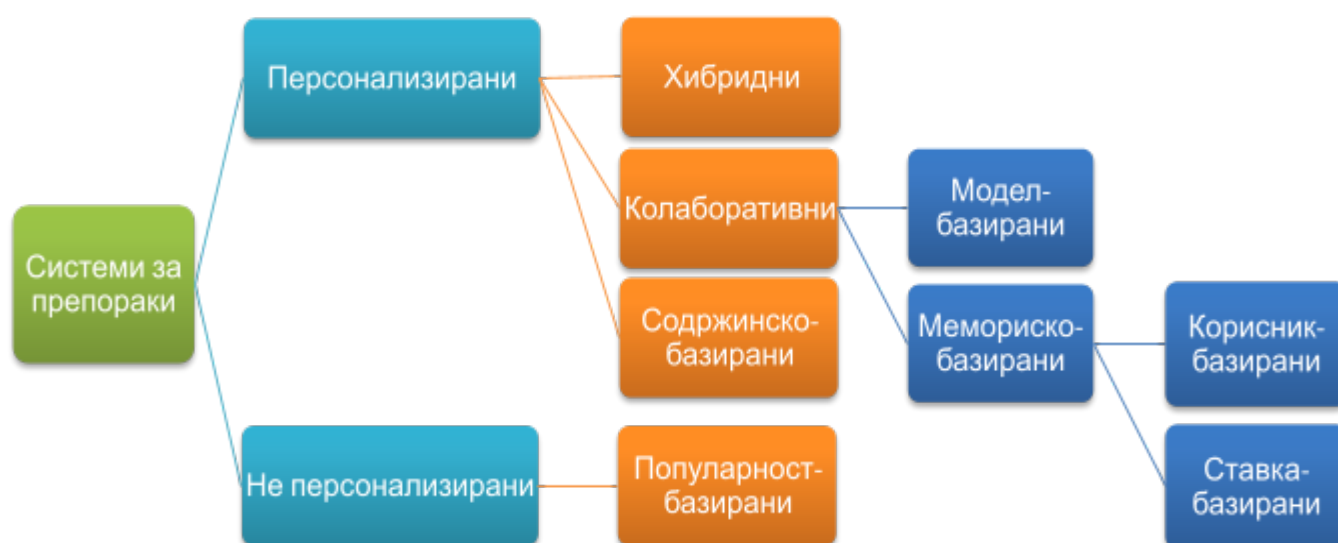
Оваа семинарска дава преглед на достигнувањата во системите за препораки пред се на оние прилагодени за видео содржини.

## 2. Што се системи за препорака ?

Системите за препораки работат на тој начин што вршат филтрирање на информациите со кои потоа на корисниците им обезбедуваат персонализирана содржина или пак препораки за производи. Овие системи ги анализираат шемите во податоците за секој корисник, како што се минатите одлуки, преференции, интеракции како и нивната демографија, за да дадат предлози кои корисникот би ги нашол корисни. Системите за препораки се многу често користени во различни апликации за да се подобрат корисничките искуства и да се поттикне пофреквентна интеракција.

## 3. Типови системи за препораки

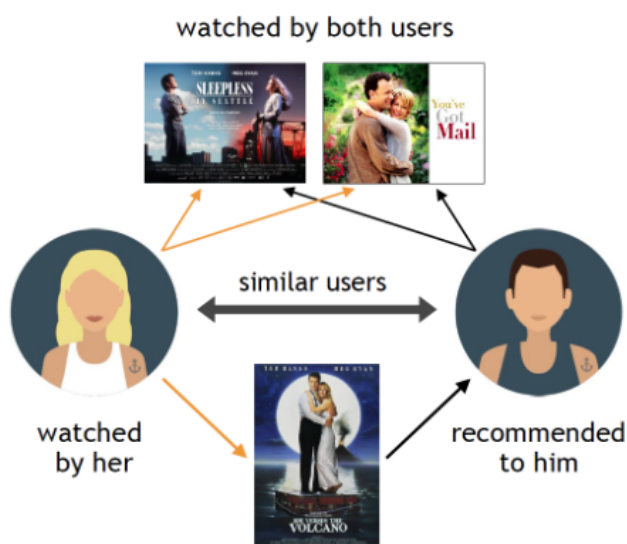
Има повеќе типови системи за препораки кои според начинот на донесување на одлуките спаѓаат во една од следните категории:



### 3.1. Персонализирани системи

Овие системи ги анализираат корисничките податоци подетално, нивните купувања, оценки и поврзаности со другите корисници. На тој начин секој корисник добива прилагодени за него препораки.

#### 3.1.1. Колаборативно филтрирање ( Collaborative Filtering )



Едноставен концепт кој работи врз основа на информациите за преференции на многу корисници и сличностите во нивните однесувања и одлуки. Со нивна помош учи да ги предвидува идните интеракции на секој корисник и врз основа на тоа да им препорача соодветни содржини.

#### 1.1.1.1. Модел-базирани ( Model based )

Овие модели се развиени со користење на алгоритми за машинско учење. Се креира модел и врз основа на него, а не сите податоци, се даваат препораки, со што се забрзува работата на системот. Овој пристап постигнува подобра скалабилност. Најпознатиот тип на овој пристап е факторизација на матрица (Слика 2).

	sci-fi	romance
Book 1	3	1
Book 2	1	2
Book 3	1	4
Book 4	3	1
Book 5	1	3

	sci-fi	romance
	✓	✗
	✗	✓
	✓	✗
	✓	✓

	Book 1	Book 2	Book 3	Book 4	Book 5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

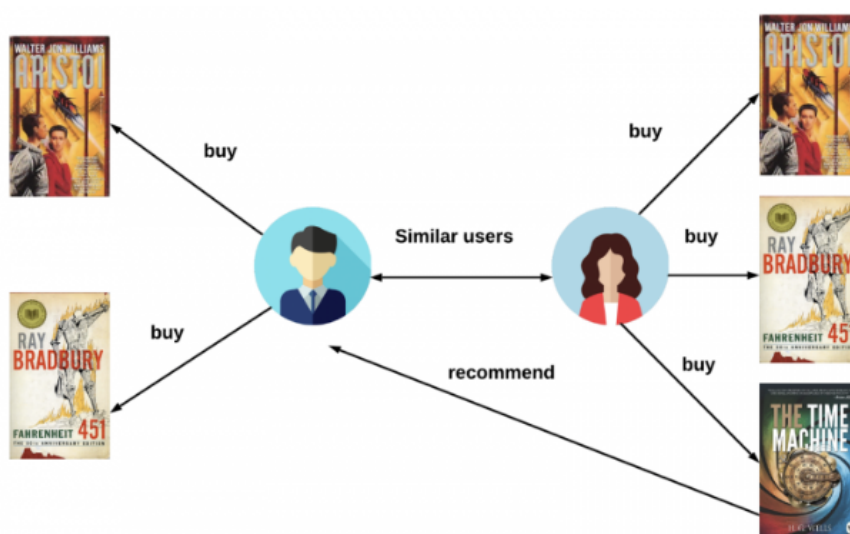
Слика 2 – Факторизација на матрица

### 1.1.1.2. Мемориско-базирани ( Memory based )

Техники базирани на меморија се применуваат врз необработени податоци. Тие се лесни за имплементација, но секој пат е потребно да се прават предвидувања за сите податоци што го успорува системот.

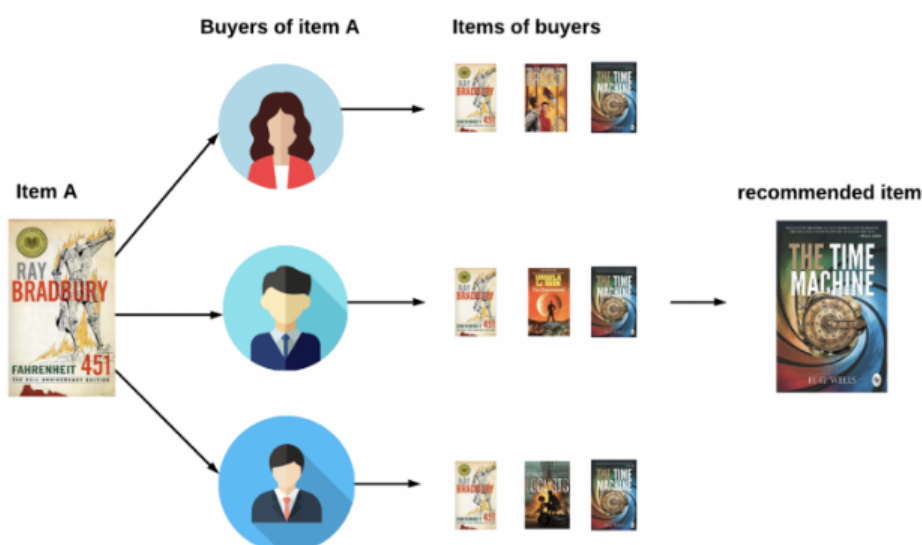
#### 1.1.1.1.1. Базирани на корисник ( User based )

Препораките на корисникот се вршат врз основа на фактот дека она што е препорачано е купено/му се допаднало на корисник со слични интереси како него.



#### 1.1.1.1.2. Базирани на ставка ( Item based )

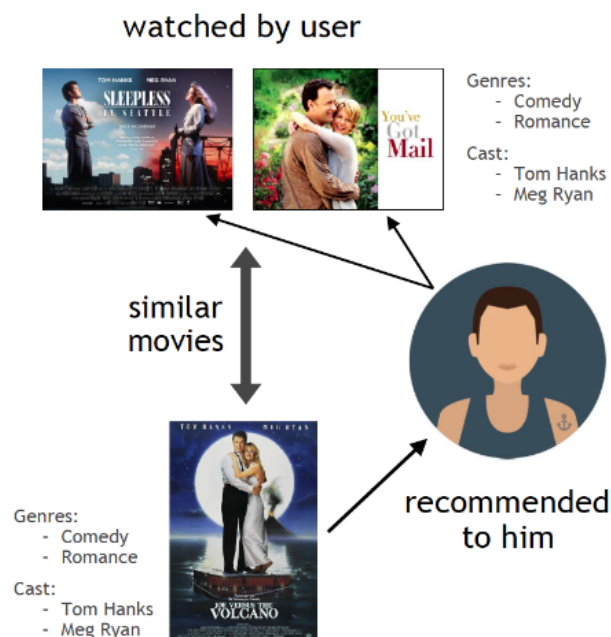
Пресметката на сличноста се врши врз основа на интеракциите со ставките од страна на корисниците кои биле групирани заедно со помош на колаборативниот филтер ( слични корисници ).



### 3.1.2. Содржинско-базирани ( Content-based Filtering )

Овој содржинско базиран систем ги користи атрибутите или одредени карактеристики на дадените содржини за да препорача други слични содржини врз основа на преференците на корисникот - информациите за корисникот, односно знаење акумулирано за корисникот и содржините со кои најчесто имал интеракција.

Оваа техника користи сличности во карактеристиките за донесување одлуки и често се користи во системи за препорака, кои се алгоритми дизајнирани да ги рекламираат или препорачуваат нештата на корисниците.



### 3.1.3. Хибридните системи за препораки

Ги комбинираат предностите на горенаведените типови за да создадат посеопфатен систем за препораки.

## 3.2. Не персонализирани

### 3.2.1. Базирани на популарност

Овде, производите се препорачуваат врз основа на тоа колку се популарни. Ставките со кои многу корисници комуницираат се сметаат за „во тренд“ и се препорачуваат на други корисници.

## 3.3. Дополнителни категории

### 3.3.1. Демографско-базирани

Овие системи препорачуваат содржини врз основа на демографскиот профил на корисникот. Основната претпоставка на овој аспект е дека треба да се генерираат различни препораки за различни демографски профили. На алгоритмите прво им треба соодветно истражување на пазарот во наведениот регион придружено со кратка анкета за да се соберат податоци за категоризација на корисниците.

### 3.3.2. Utility-базирани

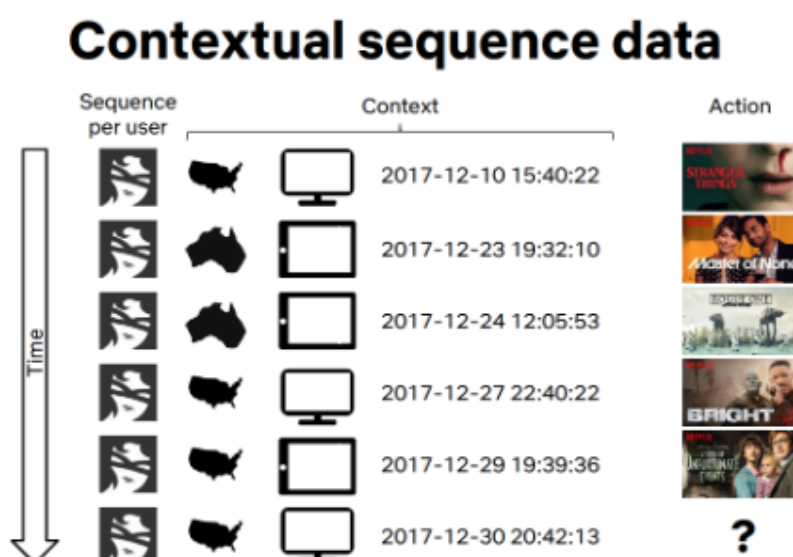
Овој систем дава предлози врз основа на пресметување на корисноста на секоја ставка за корисникот.

### 3.3.3. Системи базирани на знаење

Овој тип на систем за препораки се обидува да даде предлози врз основа на заклучоци за потребите и преференциите на корисникот.

### 3.3.4. Филтрирање врз база на контекст

Овој пристап користи низа од контекстуални кориснички активности, плус тековниот контекст, за да ја предвиди веројатноста за следната негова активност. На пример Netflix, со дадена една секвенца за секој корисник - земјата, уредот, датумот и времето кога гледал филм - тие тренирале модел да предвиди што би сакал да гледа следно.



## 4. Пример на познат систем за препораки

### 4.1. Netflix

Со преку 220 милиони претплатници, системот за препораки на Netflix е еден од најпознатите примери на персонализиран систем за препораки. Тоа е една од најпопуларните стриминг платформи со стотици филмови, анимации и телевизиски емисии. Над 80% од содржината на Netflix која корисниците ја гледаат, е преку системот за препораки на Netflix.

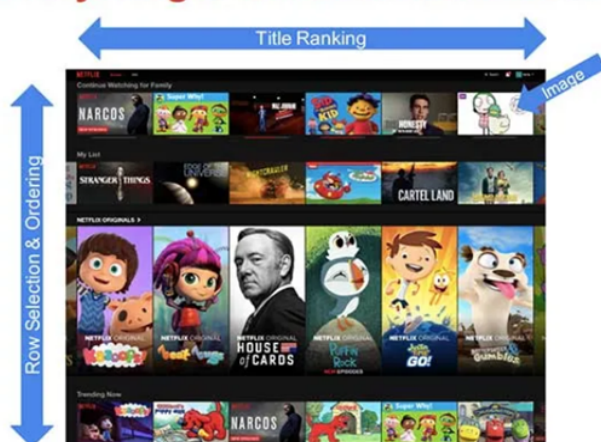




## Краток преглед како функционира системот за препораки на Netflix:

Содржината се организира во редови категоризирани обично по жанровски тип или по специфичен термин или фраза која ги поврзува сите наслови по ред. Овие категории се персонализирани за да се совпаѓаат со интересите на корисниците.

### Everything is a Recommendation



За да биде ова уште поефикасно, Netflix алгоритмот секогаш се обидува да се подобри. Секој пат кога корисникот гледа наслов, Netflix собира повеќе кориснички податоци и овие податоци се внесуваат во Netflix алгоритмот. Ова му дава на Netflix повратни информации за тоа кои техники на категоризација биле ефикасни и кои не биле.

Бидејќи ова е итеративен процес, колку повеќе време трошиме гледајќи содржини на Netflix, толку повеќе податоци Netflix собира и внесува во алгоритмот, што потоа му овозможува на системот за препораки да даде подобри и поперсонализирани препораки.

### ✓ Профилирање на корисници:

Веројатноста дека корисникот сака да гледа одреден наслов се проценува врз основа на голем број фактори вклучувајќи:

- интеракциите на корисникот со сервисот (како што е историја на гледање и оценка на другите наслови)
- други корисници со слични вкусови и преференци
- информации за насловите, како што се нивниот жанр категории, актери, година на издавање, итн.
- време во денот кога корисникот гледа
- уредите на кои гледа Netflix
- колку долго гледа.

Системот за препораки не вклучува демографски информации (како што се возраста или полот) како дел од процесот на донесување на одлука.

При пребарување на наслови, првите резултати се базирани на други корисници кои имале исти или слични пребарувања (базирани на популарност).



✓ **Каталог на содржини:**

Netflix одржува обемен каталог на филмови и ТВ емисии, од кои сите се карактеризираат со различни атрибути, како што се жанр, актери, режисери, година на издавање и многу повеќе. Овие податоци го формираат делот од системот за препораки базиран на содржина.

✓ **Колаборативно филтрирање:**

Netflix исто така користи техники за заедничко филтрирање. Ова вклучува анализа на однесувањето на корисниците и пронаоѓање модели на сличност меѓу корисниците.

✓ **Факторизација на матрица:**

Системот за препораки на Netflix користи напредни техники за факторизирање на матрици за да ја разложи матрицата за интеракција помеѓу корисникот и ставката. Техники како распаѓање на еднина вредност (SVD) или факторизација на матрица со спуштање на градиент се применуваат за да се откријат латентни фактори кои влијаат на преференциите на корисникот. Ова му овозможува на системот да дава препораки дури и кога корисниците експлицитно не оцениле или гледале одредени наслови.

✓ **Персонализација во реално време:**

Netflix постојано ги ажурира своите препораки во реално време додека корисниците комуницираат со платформата. Ако почнете да гледате нова серија или оцените филм, системот соодветно ќе ги прилагоди своите препораки.

✓ **Хибриден пристап:**

Системот за препораки на Netflix комбинира различни методи, вклучувајќи филтрирање за соработка, филтрирање базирано на содржина и модели за длабоко учење. Овој хибриден пристап помага да се подобри точноста и разновидноста на препораките.

✓ **Тестирање A/B:**

Виталниот дел од систем на препораки на Netflix се потпира на A/B тестирањето. Тие постојано тестираат различни опции во врска со сугестија за филмот, начинот на кој се организираат насловите за да се утврди што предизвикува најголем интерес.

**Главна цел на системот за препораки на Netflix:**

Главната цел на Netflix е да ги држи корисниците што е можно повеќе ангажирани и инволвирани.

Користејќи го системот на редови и на интерактивното собирање и анализа на податоци, Netflix е во состојба да генерира брзи препораки кои стануваат подобри како што корисниците стануваат поинволвирани. Исто така, со користење на уникатни слики за секој корисник, Netflix го зголемува интересот на корисниците за насловите кои можеби ги прескокнале со влијание на нивната психологија.

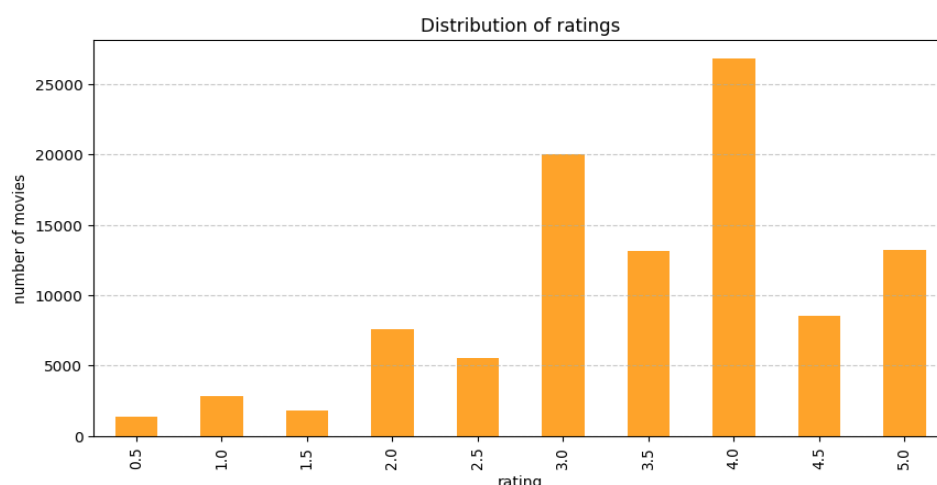
## 5. Практичен пример ( систем за препорака на филмови )

За овој пракричен пример е користено [MovieLense](#) податочното множество кое содржи околу 100 000 оцени од корисници, 9 000 филмови и 600 корисници.

```
Total data :
-----
Total number of Ratings      : 100836
Total number of unique Users : 610
Total number of unique Movies : 9724
```

[Линк](#) од Google Colab средината каде е изработен проектот.

- Како прв чекор извршено е **истражување на податочното множество** од каде можеме да ја претставиме дистрибуцијата на оцените за филмовите :

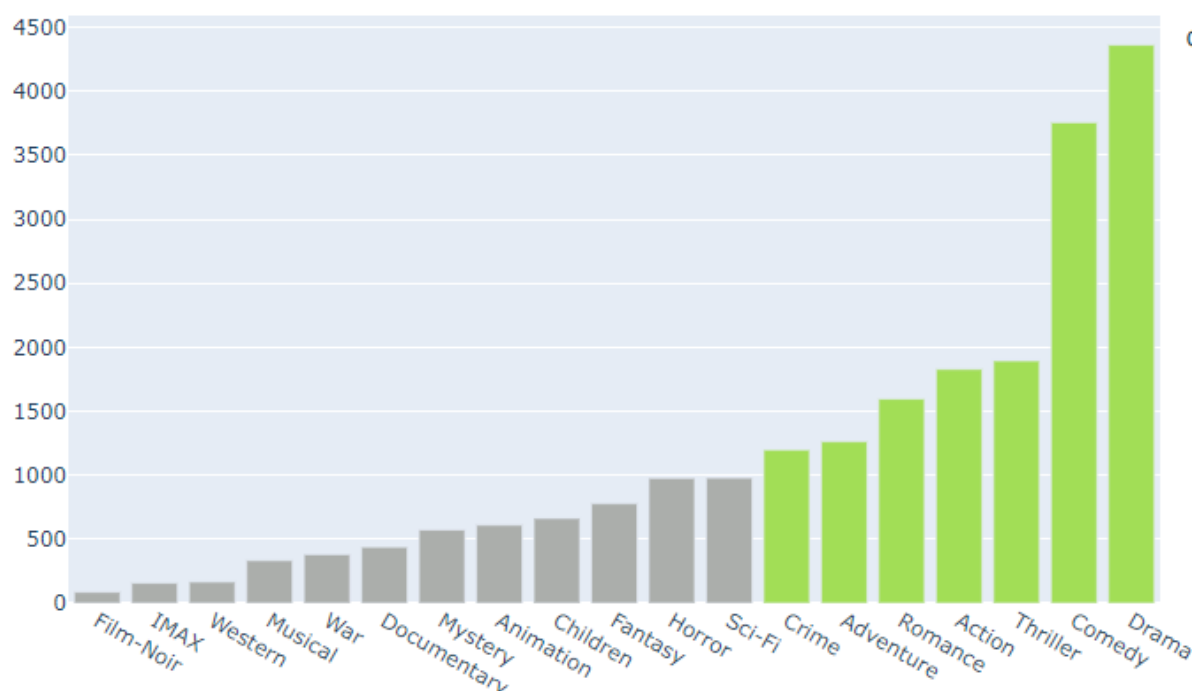


Можеме да ги откриеме 10-те најактивни корисници и бројот на дадени оцени на секој од нив, како и 10-те најпопулатни филмови за кои има најголем број оцени и кој е тој број.

Top 10 most rated movies along with the number of ratings :

title	number_of_ratings
Forrest Gump (1994)	329
Shawshank Redemption, The (1994)	317
Pulp Fiction (1994)	307
Silence of the Lambs, The (1991)	279
Matrix, The (1999)	278
Star Wars: Episode IV - A New Hope (1977)	251
Jurassic Park (1993)	238
Braveheart (1995)	237
Terminator 2: Judgment Day (1991)	224
Schindler's List (1993)	220

Дознаваме дека има 20 уникатни жанрови и кои од нив се најпопуларни.



→ **Колаборативно филтрирање (базирано на корисник)** :

Во овој дел првично се креира табела во која редовите се корисниците, а колоните се филмовите. Нивниот пресек во табелата ја претставува оцената која корисник ја има дадено за одреден филм доколку го има гледано, во спротивно вредноста во ќелијата е 0.0 :

userId	1	2	3	4	5	6	7	8	9	10	...	601	602	603	604	605	606	607	608	609	610
movieId																					
1	4.0	0.0	0.0	0.0	4.0	0.0	4.5	0.0	0.0	0.0	...	4.0	0.0	4.0	3.0	4.0	2.5	4.0	2.5	3.0	5.0
2	0.0	0.0	0.0	0.0	0.0	4.0	0.0	4.0	0.0	0.0	...	0.0	4.0	0.0	5.0	3.5	0.0	0.0	2.0	0.0	0.0
3	4.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0

Понатаму, го користиме K-Nearest Neighbour (KNN) алгоритмот и косинусна сличност. Нивната цел е да ги ставките слични со оние со кои корисникот имал позитивна интеракција. Поконкретно KNN ги наоѓа филмовите слични на даден филм базирано на интеракциите од корисниците, додека косинусната сличност ни помага при пресметка на сличностите сопредметка на аголот помеѓу два вектора за оцена. Поголема вредност за косинусна сличност значи поголемо совпаѓање.

Во класата **Recomender\_collab** има два методи:

- **recommend\_on\_movie()** кој враќа препораки за дадениот наслов чие ID се додава во листата hist која ги чува филмовите кои ги барал корисникот. Препоракат се врши со горенаведениот алгоритам со тоа што најголема сличност бараниот филм ќе има сам со себе (1) поради што го испуштаме првиот филм од оние кои се враќаат.
- **recommend\_on\_history()** кој враќа препораки заисно од содржината во листата hist (историјата филмови кои ги барал корисникот). Повторно се користи истиот алгоритам, но овој пат се испуштаат сите филмови од листата hist.

```
# no history found yet
recommender.recommend_on_history()
```

```
No history found
```

```
# recommendation based on this movie
recommender.recommend_on_movie('GoldenEye (1995)')
```

```
['Die Hard: With a Vengeance (1995)',
 'True Lies (1994)',
 'Clear and Present Danger (1994)',
 'Speed (1994)',
 'Batman (1989)']
```

```
# recommendation based on past watched movies, this time 1 movie in history
recommender.recommend_on_history()
```

```
['Die Hard: With a Vengeance (1995)',
 'True Lies (1994)',
 'Clear and Present Danger (1994)',
 'Speed (1994)',
 'Batman (1989)']
```

```
recommender.recommend_on_movie('Pocahontas (1995)')
```

```
['Beauty and the Beast (1991)',
 'Casper (1995)',
 'Lion King, The (1994)',
 'Aladdin (1992)',
 'Snow White and the Seven Dwarfs (1937)']
```

```
recommender.recommend_on_history()
```

```
['Aladdin (1992)',
 'Speed (1994)',
 'Die Hard: With a Vengeance (1995)',
 'Lion King, The (1994)',
 'Jurassic Park (1993)']
```

→ **Содржински-базирано филтрирање :**

Ова филтрирање ќе биде базирано на жанровите во кои припаѓа секој филм. Затоа првиот чекор ќе биде векторизација на содржината (жанровите) на податочното множество за филмови со помош на **CountVectorizer** кој ги трансформира жанровите во нумеричка репрезентација - бинарна матрица во која редовите се филмови а колоните уникатен жанр. Доколку одреден филм припаѓа на даден жанр вредноста на ќелијата каде што се сечат е 1, во спротивно 0.

	action	adventure	animation	children	comedy	crime	documentary	drama	fantasy	fi	...	horror	imax	musical	mystery	noir	romance	sci	thriller	war	western
0	0	1	1	1	1	0	0	0	1	0	...	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0	1	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	1	0	0	0	0
3	0	0	0	0	1	0	0	1	0	0	...	0	0	0	0	0	1	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	1	0	0
6	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	1	0	0	0	0
7	0	1	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
9	1	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	0

Понатаму со помош на KNN алгоритмот и косинусна сличност кои го наоѓаат најсличниот филм на дадениот базирано на жанрови овој пат.

Во класата **Recomender\_content** има два методи:

- **recommend\_on\_movie()** кој ги зема наслов од филм и враќа препораки базирани на сличностите пресметани погоре не вклучувајќи го филмот за кој се бараат препоракиистовремено ставајќи го во hist листата.
- **recommend\_on\_history()** кој враќа препораки заисно од содржината во листата hist (историјата филмови кои ги барал корисникот). Повторно се користи истиот алгоритам, но овој пат се испуштаат сите филмови од листата hist.

```
# no history found yet
recommender2.recommend_on_history()
```

No history found

```
recommender2.recommend_on_movie('Toy Story (1995)')
```

```
['Tale of Despereaux, The (2008)',
 'Shrek the Third (2007)',
 "Emperor's New Groove, The (2000)",
 'Wild, The (2006)',
 'Moana (2016)']
```

```
# recommendation based on past watched movies, and this time 1 movie is in history.
recommender2.recommend_on_history()
```

```
['Tale of Despereaux, The (2008)',
 'Shrek the Third (2007)',
 "Emperor's New Groove, The (2000)",
 'Wild, The (2006)',
 'Moana (2016)']
```

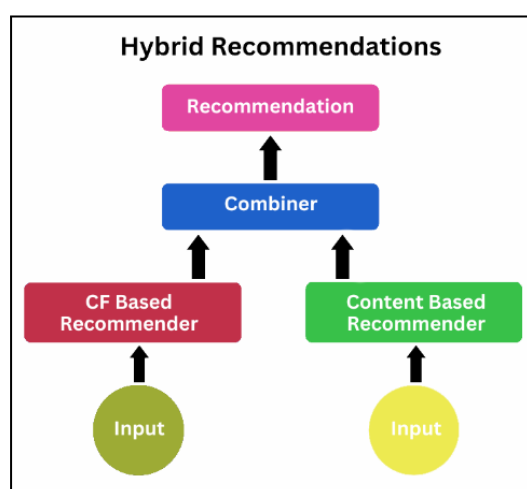
```
recommender2.recommend_on_movie('Money Train (1995)')
```

```
['Wasabi (2001)',
 'Another 48 Hrs. (1990)',
 'Metro (1997)',
 'Bad Boys (1995)',
 'Last Boy Scout, The (1991)']
```

```
recommender2.recommend_on_history()
```

```
["Twelve Tasks of Asterix, The (Les douze travaux d'Astérix) (1976)",
 'The Lego Movie (2014)',
 'Inside Out (2015)',
 'Dragonheart 2: A New Beginning (2000)',
 'TMNT (Teenage Mutant Ninja Turtles) (2007)']
```

### → Хибридно филтрирање :



Послениот чекор е комбинирање на содржинското и колаборативното филтрирање. Тоа се прави со креирање инстанци од двете класи (**Recomender\_collab** и **Recomender\_content**) и нивно комбинирање во класата **HybridRecommender**.

Овде може да се подесуваат тежините за кој препорачувач колку влијание да има на финалниот резултат.

```
movie = "Batman (1989)"

content_recommender.recommend_on_movie(movie)

['Ronin (1998)',
 'Bourne Supremacy, The (2004)',
 'U.S. Marshals (1998)',
 'Poker Night (2014)',
 'Hard Rain (1998)']

collaborative_recommender.recommend_on_movie(movie)

['Batman Forever (1995)',
 'True Lies (1994)',
 'Terminator 2: Judgment Day (1991)',
 'Fugitive, The (1993)',
 'Jurassic Park (1993)']

recommendations = hybrid_recommender.recommend(movie, n_recommend=25)
# each recommendation printed in a new line
for recommendation in recommendations:
    print(recommendation)

Beauty and the Beast (1991)
Natural Born Killers (1994)
Redemption (Hummingbird) (2013)
Judgment Night (1993)
Crimson Tide (1995)
Gangster No. 1 (2000)
Terminator 2: Judgment Day (1991)
Cliffhanger (1993)
Ronin (1998)
Brother (2000)
Speed (1994)
Dances with Wolves (1990)
Clear and Present Danger (1994)
Sin City: A Dame to Kill For (2014)
Replacement Killers, The (1998)
Next of Kin (1989)
Pulp Fiction (1994)
Equalizer, The (2014)
GoldenEye (1995)
Punisher, The (2004)
Die Hard: With a Vengeance (1995)
Aladdin (1992)
Apollo 13 (1995)
Poker Night (2014)
Hard Rain (1998)
```

## 6. Дополнителна активност

При изработка на системот за содржински филтрирање изработив матрица на сличности помеѓу филмовите со помош на Пеарсонова корелација за калкулирање сличности.



title	'71 (2014)	'Hellboy': The Seeds of Creation (2004)	'Round Midnight (1986)	'Salem's Lot (2004)	'Til There Was You (1997)	'Tis the Season for Love (2015)	'burbs, The (1989)	'night Mother (1986)	(500) Days of Summer (2009)	*batteries not included (1987)
title										
'71 (2014)	1.000000	-0.001642	-0.002324	-0.001642	-0.002254	-0.001642	-0.006407	-0.001642	0.135943	-0.004325
'Hellboy': The Seeds of Creation (2004)	-0.001642	1.000000	0.706526	-0.001642	-0.002254	-0.001642	-0.006407	-0.001642	-0.010568	-0.004325
'Round Midnight (1986)	-0.002324	0.706526	1.000000	-0.002324	-0.003191	-0.002324	0.170199	-0.002324	-0.014958	-0.006121
'Salem's Lot (2004)	-0.001642	-0.001642	-0.002324	1.000000	0.857269	-0.001642	-0.006407	-0.001642	-0.010568	-0.004325
'Til There Was You (1997)	-0.002254	-0.002254	-0.003191	0.857269	1.000000	-0.002254	-0.008797	-0.002254	-0.014510	-0.005938

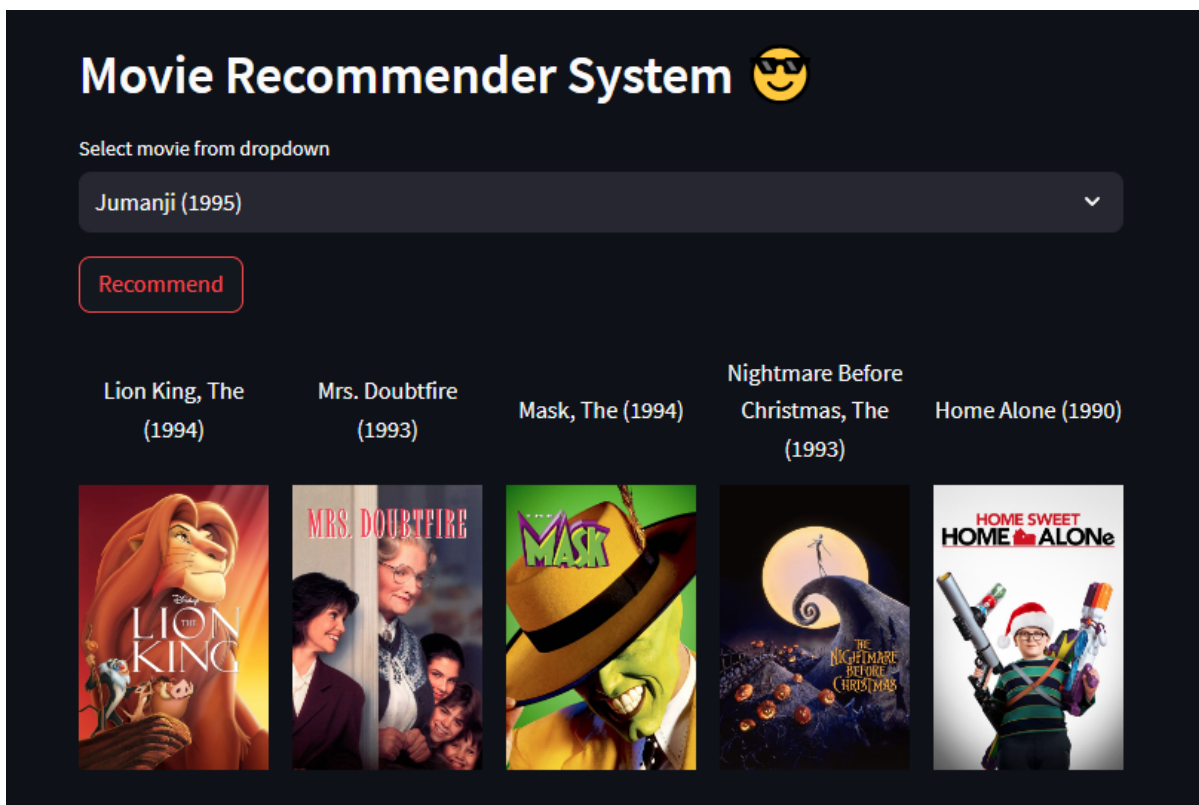
Во оваа матрица вредностите се движат од -1 до 1 и колку е повисока вредноста толку е поголема сличноста. Може да забележиме дека секој филм е со вредност 1 сличен сам на себе.

Со повик на функцијата **paerson\_rec()** за одреден наслов се листаат филмовите со најголема оцена за сличност :

```
movieLiked = 'Waiting to Exhale (1995)'
paerson_rec(movieLiked)

title
Before and After (1996)           0.544701
Georgia (1995)                   0.454782
Haunted World of Edward D. Wood Jr., The (1996) 0.454782
Sum of Us, The (1994)            0.454782
Nina Takes a Lover (1994)        0.454782
Bliss (1997)                     0.454782
Last Dance (1996)                0.454782
Federal Hill (1994)              0.454782
Jefferson in Paris (1995)        0.454782
Name: Waiting to Exhale (1995), dtype: float64
```

За ова успеав да имплементирам и интерфејс во кој со помош на API се прикажуваат и постерите од препорачаните филмови :



## 7. Заклучок

Системите за препораки како оној на Netflix се софистицирани и динамични системи кои користат многу методи и напредни алгоритми за да испорачаат персонализирани препораки за содржина на своите корисници, подобрувајќи го корисничкото искуство и зголемувајќи ја интеракцијата на корисниците.

Во практичниот пример гледаме како приближно може да се изгради еден едноставен таков систем кој може многу да се надоградува и усовршува понатаму.

Доколку би продолжила со проширување на системот би ги елиминирала филмовите со мал број на оценки (проверивме дека нема филм без оценка) бидејќи KNN може да не работи ефикасно во такви ситуации.

Исто така би пробала да направам содржинско филтрирање по тагови дадени од корисниците.

## 8. Користена литература

- [1] <https://www.nvidia.com/en-us/glossary/data-science/recommendation-system/#:~:text=A%20recommendation%20system%20is%20an,demographic%20information%2C%20and%20other%20factors.>
- [2] <https://www.kaggle.com/code/sachinsarkar/movielens-movie-recommendation-system>
- [3] <https://www.shiksha.com/online-courses/articles/movie-recommendation-system-using-machine-learning/>
- [4] <https://thingsolver.com/blog/introduction-to-recommender-systems/>
- [5] [https://eprints.ugd.edu.mk/17574/1/AleksandarKotevski\\_PhD.pdf](https://eprints.ugd.edu.mk/17574/1/AleksandarKotevski_PhD.pdf)
- [6] <https://www.bluepiit.com/blog/classifying-recommender-systems/>
- [7] <https://www.analyticsvidhya.com/blog/2020/11/create-your-own-movie-movie-recommendation-system/>
- [8] <https://www.kaggle.com/code/sachinsarkar/movielens-movie-recommendation-system>
- [9] <https://www.youtube.com/watch?v=kuC38ZCcbZI>
- [10] <https://help.netflix.com/en/node/100639#:~:text=We%20estimate%20the%20likelihood%20that,preferences%20on%20our%20service%2C%20and>