Системи за препоруку[1]

Предговор

Људи су одувек имали потребу да се посаветују у вези потешкоћа и проблема на које су наилазили. И у ово, модерно доба, подједнако се траже савети и смернице, како за превазилажење различитих препрека, тако и за остваривање зацртаних циљева. Ова одлика људске природе није уопште необична. У односу на давна времена, промениле су се једино потребе. Заједничко је то што је увек постојао висок степен поверења у субјекта који је делио савете и препоруке. Заиста, у ситуацијама прожетим неизвесностима и непознатим околностима, свака особа тражи савете од најближих пријатеља, а посебно од оних коју су већ имали слична искуства. Са друге стране, најбољи поклони се редовно добијају од најблискијих особа, јер они најбоље познају жеље, укусе и склоности појединца.

У ери убрзаног развоја технологије јавила се потреба за системима који би у одређеним ситуацијама „заменили" особе којима се верује. Како би људи стекли поверење у такве системе, неопходно је да их систем на известан начин „упозна", а затим и пружи одређену препоруку. Управо ту почиње магија различитих техника и приступа.

Људи свакодневно користе системе за препоруке несвесни њиховог постојања. Пример система за препоруку је приказивање новинских чланака кориснику на основу његових афинитета. Такође, многи интернет сајтови за продају персонализују понуду, нудећи кориснику тачно одређени скуп производа у складу са његовом историјом куповине или претраге.

Примери система за препоруку

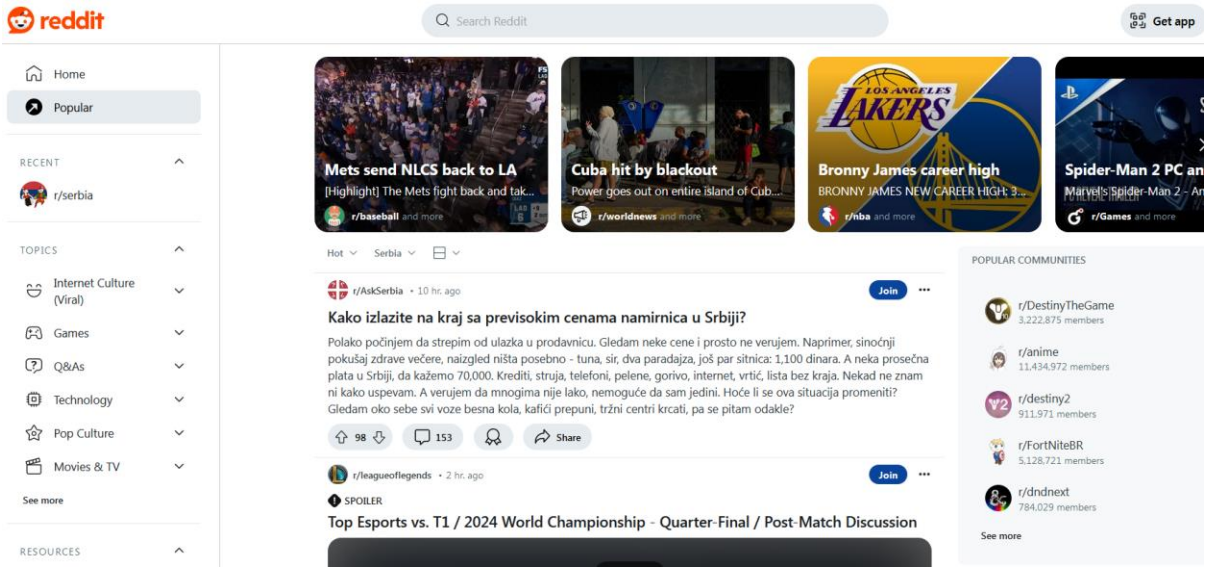
GOOGLE

PageRank (пар смерница је дато у посебном пдф фајлу) – сопствени вектори матрице
Search Engine Optimization – SEO
Једна SEO конференција је била одржана и у Крагујевцу пре пар година:
https://www.seoexpert.rs/blog/it-open/

[1] Одређени делови текста су преузети из мастер рада „Системи за препоруку", колеге Миладина Јовића.

REDDIT

https://www.reddit.com/



Hacker News и *Reddit* сервиси уместо сложених система за препоруку користе кориснички дефинисане математичке функције, које обухватају популарност и време трајања неке вести. На пример, *Reddit* користи следећу функцију:

$$score = sgn(ups - downs) \times log\big(max(1, |ups - downs|)\big) + \frac{age}{45000}$$

где су *ups* и *downs* позитивне/негативне препоруке, а *age* време трајања вести (у секундама).

Имплементација:

https://github.com/reddit-archive/reddit/blob/master/r2/r2/lib/db/_sorts.pyx#L47

Неке друге имплементације које се могу пронаћи на Интернету:

$$score = \frac{(ups - downs)}{(t+2)^G}$$

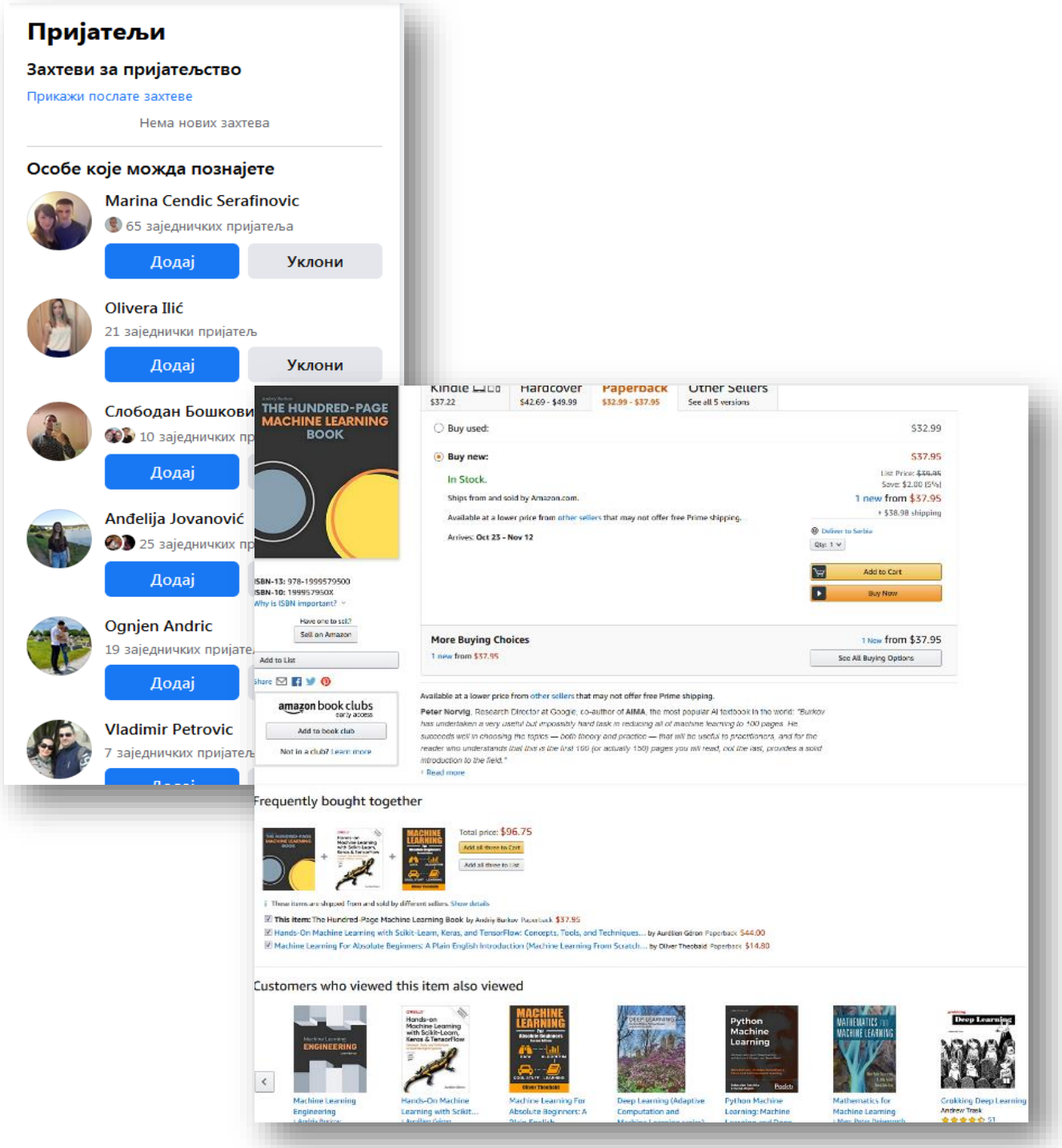
где је t време протекло од објављивања поста (у сатима), а G фактор гравитације који одређује брзину „понирања“ вести кроз време.
Претпостављам да се могу наћи и друге сличне формуле, јер без експериментисања нема добрих резултата.

FACEBOOK

BLOG: Recommending items to more than a billion people

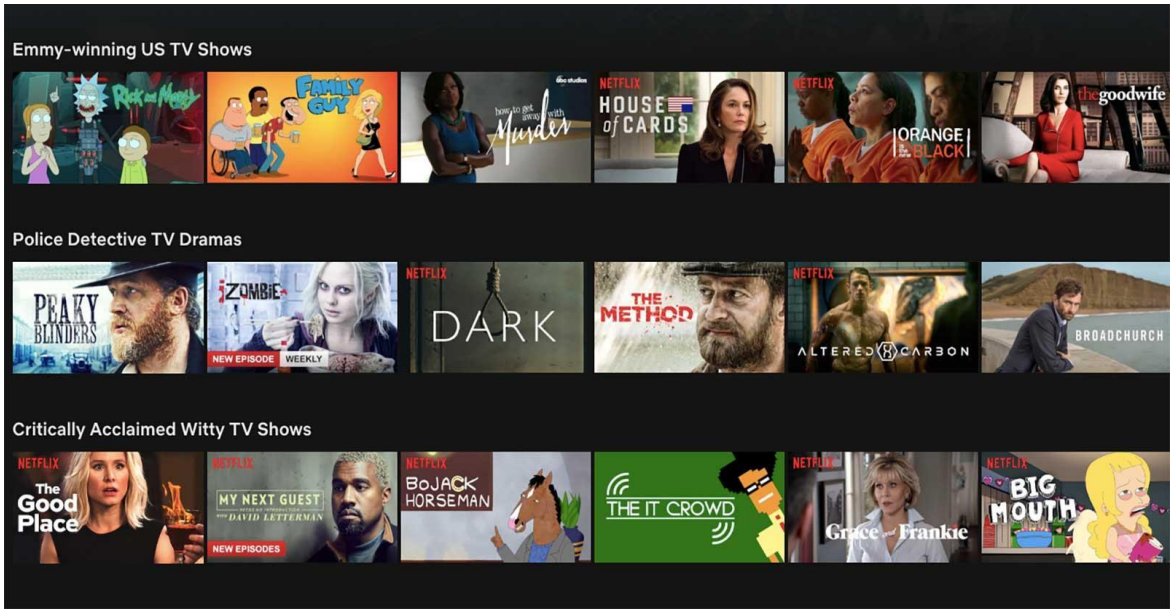
https://engineering.fb.com/core-data/recommending-items-to-more-than-a-billion-people/

AMAZON - Амазон својих 35% прихода приписује свом систему препорука.

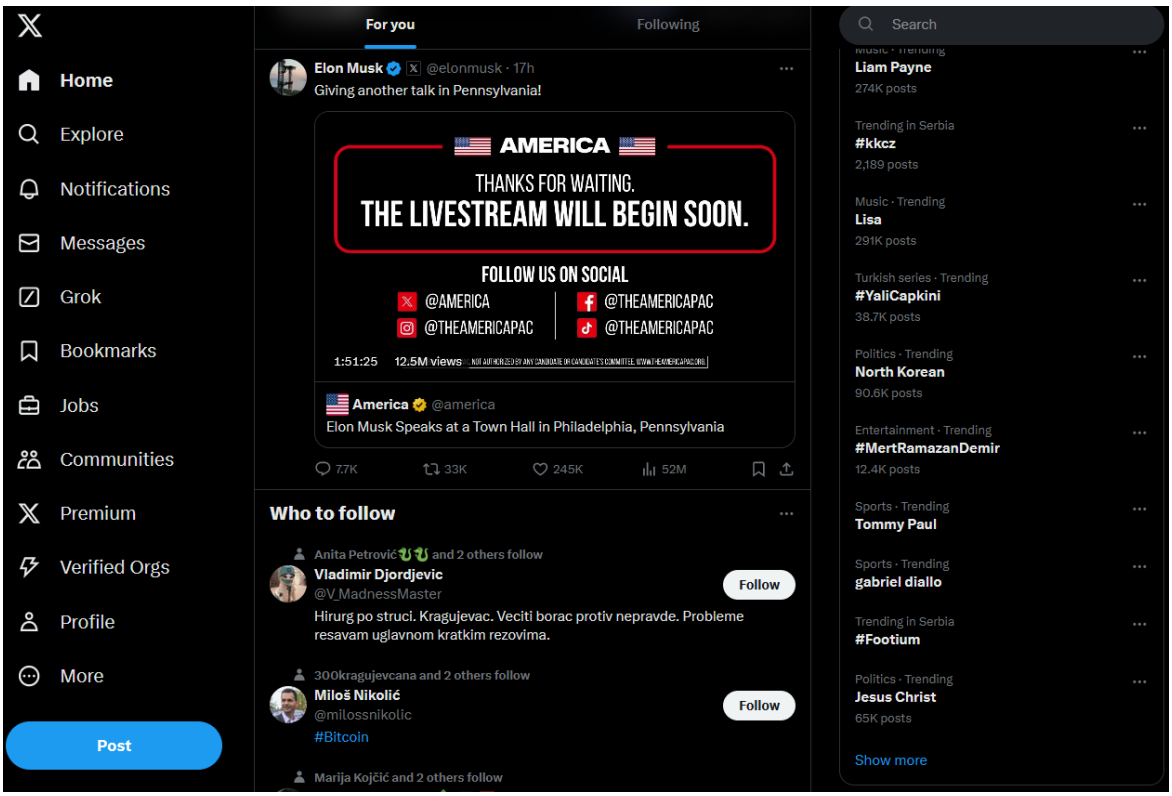


NETFLIX

Netflix Prize: https://en.wikipedia.org/wiki/Netflix_Prize



TWITTER



1. Увод

1.1. О системима за препоруку

Системи за препоруку (recommender systems) су системи чији је главни задатак да кориснику пруже препоруку о потенцијално занимљивом објекту када њихов расположив број надмашује способност корисника да их појединачно прегледа у разумном времену. Популарност ових система нарочито је порасла последњих година са повећањем количине информација које нас окружују, и појавом Интернет сервиса као што су Facebook, Google, Amazon, eBay и Netflix.

Циљ система за препоруку јесте да кориснику препоручи одређени број производа за које постоји највећа вероватноћа да ће бити заинтересован, а са којима до тада није био упознат. Ти производи сачињавају Top-*N* листу, где *N* представља број препоручених производа. Да би систем успешно генерисао поменуту листу, неопходно је да поседује податке о укусу корисника, његовим преференцама итд. Поменути подаци се прикупљају интеракцијом корисника са системом. У том смислу, разликујемо две врсте података о кориснику:

- *експлицитни* (explicit feedback)
- *имплицитни* (implicit feedback)

У експлицитне податке се убраја оцена (rating) корисника неког производа (нпр. број од 1 до 5). Под имплицитним подацима се убраја тзв. *click logs*, где се подразумева да је корисник заинтересован за артикал ако је кликнуо на исти или ако је неки производ купио или додао у корпу за куповину. Време проведено у прегледу једног производа такође може да буде имплицитна информација систему о заинтересованости корисника.

1.2. Матрица корисности

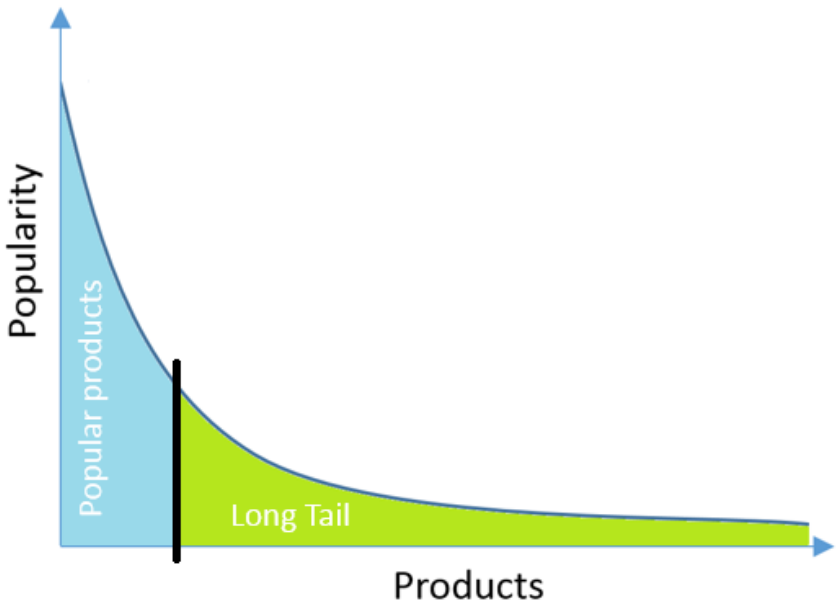
У системима за препоруку постоје два ентитета - **корисник** (*user*) и **артикал** (*item*). Интеракцијом корисника са системом формира се **матрица корисности** (*Utility Matrix*) која за сваки пар корисник-производ чува вредност која говори о степену афинитета корисника према том производу (нпр. један број на скали од 1 до 5 или пример имплицитних података - јединица ако је корисник кликнуо на артикал, у супротном нула). Врло често је ова матрица празна, тј. корисници углавном дају оцене о веома малом броју производа у односу на цео скуп. Неки алгоритми система за препоруку се управо фокусирају на то да предвиде коју би оцену дао корисник неоцењеним производима.

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Табелом 1 приказан је пример једне матрице корисности попуњене експлицитним подацима. Редови представљају кориснике (*A*, *B*, *C*, *D*) а колоне артикле (филмове) где скраћенице *HP1*, *HP2*, *HP3* означавају *Harry Potter I*, *II*, *III*, *TW Twilight*, а *SW1*, *SW2*, *SW3* се односе респективно на *Star Wars I*, *II*, *III*. Празна поља значе да корисник није оценио поменуте филмове.

1.3. Феномен дугог репа (Long Tail)

Разлика између класичних продајних објеката и онлајн продавница је позната као феномен дугог репа, што је приказано сликом 1. Вертикална оса означава популарност (број продатих примерака артикла). По хоризонталној оси су наведени производи према њиховој популарности. Класична продајна места, због ограничености простора, излажу само најпопуларније артикле који се налазе лево од вертикалне линије. Са друге стране, онлајн продавнице могу да понуде читав скуп производа, како популарних, тако и оних мање познатих. Дуги реп управо представља подручје претраге система за препоруку којом се долази до мање популарних производа који би могли у великој мери задовољити захтеве купаца. Дакле, ова област омогућава купцима да задовоље своје јединствене, профињене и истанчане укусе, а са друге стране, власницима система, да повећају стопу профита. На овај начин су сви на добитку.



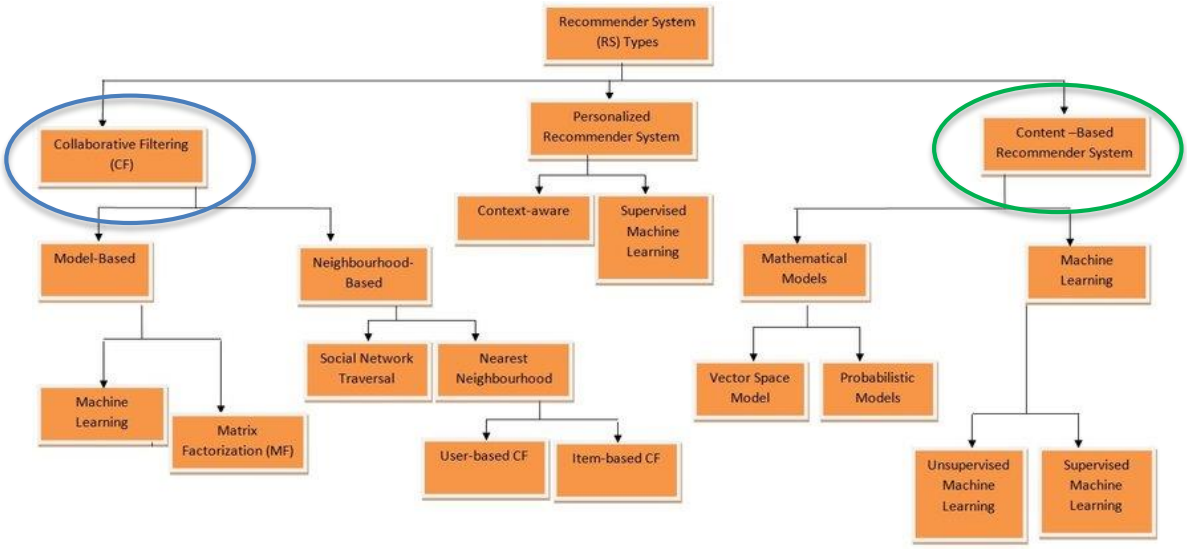
Слика 1 - Феномен дугог репа

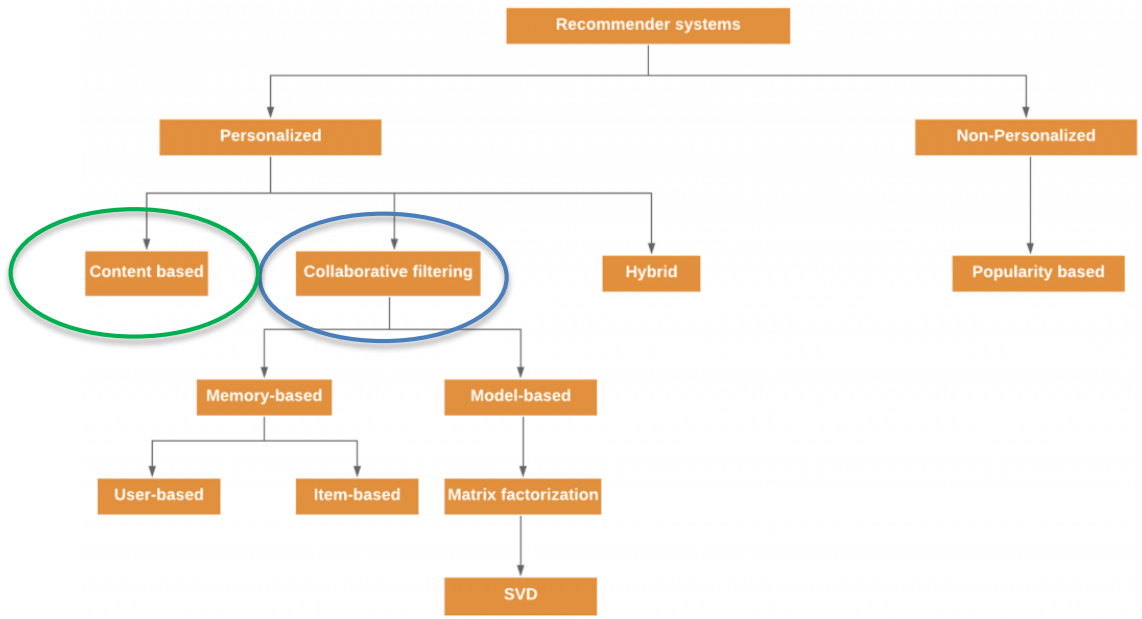
1.4. Класификација система за препоруку

Алгоритме за израду система за препоруку можемо поделити у три велике категорије:

- **Системи засновани на садржају** (*Content-Based systems*)
- **Системи засновани на сарадњи** (*Collaborative filtering systems*)
- **Системи засновани на моделу** (*Model-Based systems*)

Постоји и тзв. **хибридни** приступ који представља комбинацију прва два наведена алгоритма. У неким случајевима, овај приступ може бити доста ефикаснија од *content-based* и *collaborative filtering* приступа. Често се користи како би се превазишли неки уобичајени проблеми система за препоруку као што су **cold start** и проблем празне матрице корисности (**sparsity problem**). О поменутим изазовима биће више речи у наставку. Још неке поделе система за препоруку:



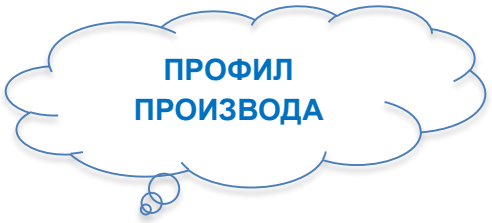


2. Теоријске основе

2.1. Системи засновани на садржају

Овај метод употребљава податке о опису и својствима производа које је корисник раније оценио како би моделовао његове преференце. Другим речима, алгоритам препоручује оне производе, који су по особинама „слични" производима који су се свидели кориснику у прошлости.

Употреба овог алгоритма подразумева конструкцију профила за сваки производ, који представља колекцију битних карактеристика тог производа. На пример, један филм карактерише скуп глумаца, година производње, жанр коме филм припада итд.



Уколико се не поседује довољно података за креирање профила, неопходно је на известан начин доћи до карактеристика тог производа. Најчешће коришћени начин код текстуалних докумената јесте употреба $TF\text{-}IDF$ мере. Наведеном мером се сваки текстуални документ описује помоћу n речи из тог документа које су добиле највећи $TF\text{-}IDF$ резултат. Тих n речи формирају вектор где свака координата одговара $TF\text{-}IDF$ резултату коју је добила та реч. $TF\text{-}IDF$ резултат описује значајност коју та реч има за наведени текст. TFIDF се може одредити према следећој формули

$$TF\text{-}IDF = TermFrequency \cdot InverseDocumentFrequency$$

где је $Term\ Frequency$ број појављивања речи у документу (неком тексту, чланку, опис филма, опис производа на Амазону, итд), док $Inverse\ Document\ Frequency$ описује колико докумената садржи одређену реч. $Inverse\ Document\ Frequency$ се **најчешће** рачуна као

$$\log \frac{\#documents}{\#documents\ with\ term}, \tag{2}$$

где $\#documents$ представља укупан број докумената у скупу, а $\#documents\ with\ term$ означава број докумената из тог скупа који садрже задату реч. Такође, често се користе и нешто другачије формуле за TF и IDF (wiki). Реч добија висок $TF\text{-}IDF$ скор ако има високу учесталост термина (често се појављује у документу), али ниску учесталост на нивоу докумената (појављује се у неколико докумената у целом корпусу).

Поред наведеног, постоји још много начина за прикупљање података о својствима објекта. Један од њих јесте остављање ознака (*tags*) од стране корисника. Ознаке представљају речи или фразе који описују производе. Велики недостатак овог приступа јесте значајна зависност од воље корисника да креирају ознаке. У случају недовољно придружених ознака производу, није могуће изградити квалитетан систем за препоруке заснованом на садржају и потребно је окренути се другим приступима.

Узмимо, на пример, систем за препоруку филмова.

Питања:

- Шта све могу да буду параметри за опис филмова?

ОДГОВОР: Глумци, режисер, година када је филм имао премијеру, жанр (IMDB) итд. Опис филма смо већ поменули.

- Како бисте моделовали глумце? Како бисте радили са описом филма?
- Како бисте почели рад у случају да сајт нема опцију за остављање рејтинга/оцена?

Основни кораци у креирању вектора за производе, у системима за препоруку заснованим на садржају (пример ако имамо опис и жанрове филмова):

1. Креирајте засебне векторе:

***TF-IDF* вектор**: Овај вектор представља опис филма. Обично има високу димензију, јер укључује много јединствених термина из корпуса (свих описа филмова).

Вектор жанрова: Ово је вектор нижих димензија који представља жанрове филма. Обично је то бинарни вектор (1 или 0), где свака димензија одговара специфичном жанру (нпр. Акција, Комедија, Драма, итд.).

Пример:

TF-IDF вектор (за опис филма): [0.5, 0.3, 0.8, 0, 0.2] (вредности представљају тежине термина за различите речи из описа).

Вектор жанрова (за Акцију и Научну фантастику): [1, 0, 1, 0, 0] (где димензије представљају Акцију, Комедију, Научну фантастику, Романсу, Драму).

2. Нормализујте векторе:

Пошто *TF-IDF* вектор и жанровски вектор могу имати значајно различите скале и димензије, важно је нормализовати оба вектора како би се могла упоређивати када се споје.

Нормализујте *TF-IDF*: Обично можете користити L2 нормализацију (дужина вектора једнака 1) за *TF-IDF* вектор. Ово скалира вектор тако да његове вредности буду у релативном односу.

$$TF\text{-}IDFnorm(v) = \frac{v}{\|v\|}$$

Нормализујте вектор жанрова: Жанровски вектори су често бинарни, али можете применити L2 нормализацију и овде, како би били на истој скали. На пример:

$$Genre\ norm(u) = \frac{u}{\|u\|}$$

3. Доделите пондер:

Сада када су оба вектора нормализована, можете доделити различите пондерисане тежине сваком скупу карактеристика на основу тога колико су вам жанрови или описи важни у препоручивању.

Дефинишите два пондера:

- $\boldsymbol{\alpha}$ за вектор жанрова
- $\boldsymbol{\beta}$ за $TF\text{-}IDF$ вектор

Ови пондери се могу прилагодити у зависности од тога колико утицаја желите да свака компонента има на крајњу препоруку. На пример, ако су жанрови важнији за ваше препоруке, можете поставити $\alpha > \beta$.

4. Комбинујте векторе:

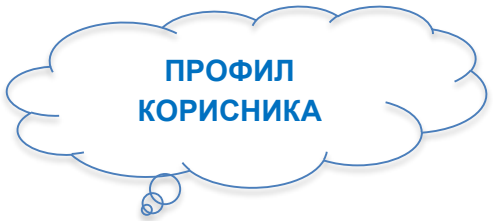
Комбинација: Спојите два вектора у један већи вектор тако што ћете их конкатенирати, што је уобичајен начин комбиновања различитих врста карактеристика.

$$Komb\_vektor = \alpha \cdot normalizovani\ vektor\ žanrova \oplus \beta \cdot normalizovani\ TF\text{-}IDF\ vektor$$

Симбол $\oplus$ означава конкатенацију вектора.

ВАЖНО: Дакле, сваки производ се описује вектором са координатама које означавају „степен присутности" неког његовог својства.

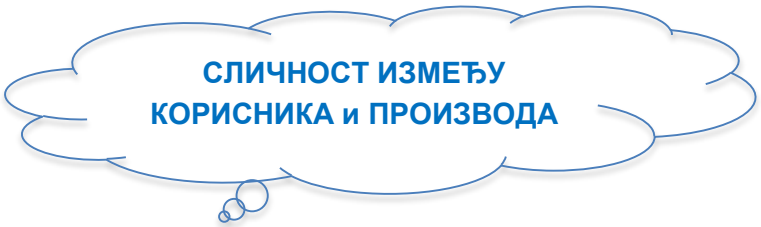
Поред тога, неопходно је креирати и вектор са истим компонентама који описује укус корисника. Како матрица корисности описује везу између корисника и производа, профил корисника се добија пондерисаним просеком профила свих производа са остављеном оценом.



Пре генерисања корисничких профила, потребно је нормализовати матрицу корисности одузимањем просечне оцене за сваког појединачног корисника од његових оцена. На тај начин ће тежински коефицијенти бити позитивни за оне артикле који се „више" свиђају одређеном субјекту, а за преостале негативни или једнаки нули.

Пример:

БЛОГ (обратити пажњу како су креирани вектори који описују преференце корисника према одређеним производима).



Као последњи корак треба измерити „сличност" између профила корисника и профила оних артикала које до тада није видео (кандидати за препоруку). Постоји више функција сличности које се користе. Најпознатије су *Jaccard similarity, Cosine similarity* и Пирсонов коефицијент корелације. Највише се користи *Cosine similarity* који се дефинише помоћу следеће формуле:

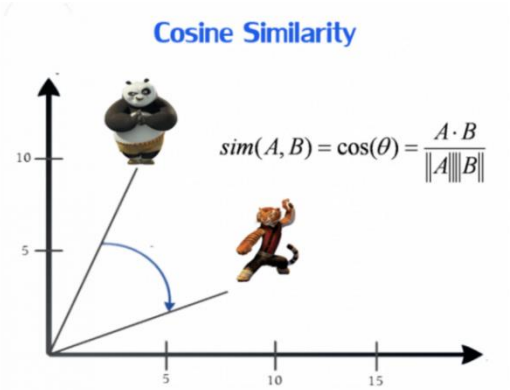
$$CosSim(\overrightarrow{up}, \overrightarrow{ip}) = \frac{\overrightarrow{up} \cdot \overrightarrow{ip}}{|\overrightarrow{up}| \cdot |\overrightarrow{ip}|}$$

где је $\overrightarrow{up}$ вектор који одговара корисничком профилу, а $\overrightarrow{ip}$ профил производа тј. потенцијалног кандидата. Ова метрика је заправо косинус угла који профил корисника заклапа са профилом производа, па стога важи:

$$-1 \leq CosSim(\overrightarrow{up}, \overrightarrow{ip}) \leq 1$$

Лако се закључује да мања вредност угла заклапања означава и већу вредност сличности између корисничког профила и профила производа.

Сви кандидати за препоруку се најпре рангирају према степену сличности са корисничким преференцама. Затим се кориснику приказује одређени број најбоље рангираних артикала у виду *Top-N* листе. Наравно, цео посао се не завршава само на овоме. Постоји много метрика којима се процењује квалитет ове листе, а о свему томе биће речи у наредним терминима. У наставку су још једном приказани кораци у овим системима за препоруку.



Још једном да се сумирају основни кораци у системима за препоруку заснованим на садржају:

1. *Нормализација матрице корисности* - одузимање просечне оцене за сваког појединачног корисника од његових оцена.
2. *Конструисање профила за сваки производ* (вектори истих димензија).
3. *Конструисање профила за сваког корисника* (вектор истих димензија као вектор производа) - профил корисника се добија пондерисаним просеком профила свих производа са остављеном оценом.
4. *Одређивање „сличности" између профила корисника и профила оних артикала* које до тада није видео (кандидати за препоруку).
5. Након сортирања добијених сличности формирати Top-*N* листу препорука за корисника.

Примере погледати на блогу и у постављеном ексел фајлу.

Погледати пример на блогу. Подаци од више предиктора је стопљено у један текст, а онда се израчунавају TF-IDF скорови.

Предности система препоруке на основу садржаја су:

- Независност од осталих корисника
- *Транспарентност* – на темељу својстава појединог објекта, могуће је јасно одредити зашто је одређени објекат препоручен.
- Могућност препоручивања нових објеката који претходно нису били оцењени.

Главни недостаци ових система су:

- *Ограничена анализа садржаја* – методе за извлачење знања довољно су добре за анализу текстуалних докумената, али не и мултимедијалног садржаја попут слика, звучних и видео записа, због чега је пожељно ручно класификовати ове објекте. Такође сама текстуална анализа, која се темељи на заступљености најважнијих речи, не може разликовати добро написане од лоше написаних чланака.
- *Претерана специјализација* – ови системи су у стању да препоруче једино објекте који су слични постојећим занимљивим објектима. У завршници то може резултовати давањем очигледних препорука и/или превидом потенцијално занимљиве препоруке. Пример система који препоручује ресторане, особи која је посећивала само кинеске ресторане неће препоручити (најбољи) италијански ресторан.
- *Проблем новог корисника* – како би препорука била што боља, потребно је прикупити што већи број оцена. Тако да у случају новог корисника, систем не може дати прецизну препоруку.

Још пар коментара...

Пример: Нека матрица корисности за оцену филмова има све попуњене ћелије са оценама 1–5. Нека се као својства која описују филмове посматрају глумци у тим филмовима. Ако 20% филмова које корисник воли има Џулију Робертс као глумицу, тада профил корисника има компоненту 0.2 за ову глумицу.

Ако корисник U има просечну оцену 3, и оценио је три филма са Џулијом Робертс оценама 3, 4, и 5, тада компонета корисника U која одговара овој глумици има вредност 1.

Објашњење: ((3-3) + (4-3) + (5-3))/3 = 3/3 = 1

Ако корисник V има просечну оцену 4, и оценио је три филма са Џулијом Робертс оценама 2, 3, и 5, тада компонета корисника V која одговара овој глумици има вредност -2/3.

Пондерисање карактеристика у системима препорука заснованим на садржају

$$Movie\_vector = 0.3 * Description \oplus 0.15 * Actors \oplus 0.25 * Genre \oplus 0.1 * Year \oplus \dots$$

Сада се поставља питање да ли је доменско знање довољно да ми сами можемо да дефинишемо тежине и тако осликамо значај одређених особина (нпр. очекивано је да главни глумац има већу тежину од глумца са споредном улогом или камермана).

Уколико приметимо да одређена сличност између корисника и производа не осликава право стање ствари, тада можемо додати још један корак:

Подешавање пондера:

- Можете експериментисати са различитим вредностима за α и β да бисте пронашли најбољу равнотежу између релевантности жанрова и описа из нашег примера. Ако су жанрови критичнији, повећајте α, а ако су описи важнији, повећајте β.
- Често можете користити *grid-search* или друге технике оптимизације да пронађете оптималну равнотежу на основу повратних информација корисника или перформанси препорука.

TIPS: Techniques for Content-based Recommender Systems

This is a kind of meta-analysis article about TF-IDF, Word Embeddings (Word2Vec), Topic Modeling (LDA, NMF, SVD), Lda2Vec (Word2Vec + LDA)