Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-182905-74277

Bc. Peter Tibenský

Personalizované odporúčanie produktov používateľom

Diplomová práca

Študijný program: Inteligentné softvérové systémy
Študijný odbor: 9.2.5 Softvérové inžinierstvo, 9.2.8 Umelá inteligencia
Miesto vypracovania: Ústav informatiky, informačných systémov a softvérového inžinierstva, FIIT STU, Bratislava
Vedúci práce: doc. Ing. Michal Kompan, PhD.

apríl 2019

Zadanie diplomovej práce

Meno študenta: **Bc. Peter Tibenský**

Študijný program: Inteligentné softvérové systémy

Študijný odbor: Softvérové inžinierstvo – hlavný študijný odbor
Umelá inteligencia – vedľajší študijný odbor

Názov práce: **Personalizované odporúčanie produktov používateľom**

Samostatnou výskumnou a vývojovou činnosťou v rámci predmetov Diplomový projekt I, II, III vypracujte diplomovú prácu na tému, vyjadrenú vyššie uvedeným názvom tak, aby ste dosiahli tieto ciele:

Všeobecný cieľ:

Vypracovaním diplomovej práce preukážte, ako ste si osvojili metódy a postupy riešenia relatívne rozsiahlych projektov, schopnosť samostatne a tvorivo riešiť zložité úlohy aj výskumného charakteru v súlade so súčasnými metódami a postupmi študovaného odboru využívanými v príslušnej oblasti a schopnosť samostatne, tvorivo a kriticky pristupovať k analýze možných riešení a k tvorbe modelov.

Špecifický cieľ:

Vytvorte riešenie zodpovedajúce návrhu textu zadania, ktorý je prílohou tohto zadania. Návrh bližšie opisuje tému vyjadrenú názvom. Tento opis je záväzný, má však rámcový charakter, aby vznikol dostatočný priestor pre Vašu tvorivosť.

Riaďte sa pokynmi Vášho vedúceho.

Pokiaľ v priebehu riešenia, opierajúc sa o hlbšie poznanie súčasného stavu v príslušnej oblasti, alebo o priebežné výsledky Vášho riešenia, alebo o iné závažné skutočnosti, dospejete spoločne s Vaším vedúcim k presvedčeniu, že niečo v texte zadania a/alebo v názve by sa malo zmeniť, navrhnite zmenu. Zmena je spravidla možná len pri dosiahnutí kontrolného bodu.

Miesto vypracovania: Ústav informatiky, informačných systémov a softvérového inžinierstva, FIIT STU v Bratislave

Vedúci práce: **Ing. Michal Kompan, PhD.**

Termíny odovzdania:

Podľa harmonogramu štúdia platného pre semester, v ktorom máte príslušný predmet (Diplomový projekt I, II, III) absolvovať podľa Vášho študijného plánu

Predmety odovzdania:

V každom predmete dokument podľa pokynov na www.fiit.stuba.sk v časti: home > Informácie o > štúdiu > harmonogram štúdia > diplomový projekt.

V Bratislave dňa 12. 2. 2018

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4
1

prof. Ing. Pavol Návrat, PhD.
riaditeľ Ústavu informatiky, informačných systémov a softvérového inžinierstva

STU FIIT
SLOVENSKÁ TECHNICKÁ
UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY
A INFORMAČNÝCH TECHNOLÓGIÍ

Návrh zadania diplomovej práce

Finálna verzia do diplomovej práce [1]

Študent:

Meno, priezvisko, tituly: Peter Tibenský, Bc.
Študijný program: Inteligentné softvérové systémy
Kontakt: peter.tibensky@icloud.com

Výskumník:

Meno, priezvisko, tituly: Michal Kompan, Ing. PhD.

Projekt:

Názov:	Personalizované odporúčanie produktov používateľom
Názov v angličtine:	Personalized product recommendation for users
Miesto vypracovania:	Ústav informatiky, informačných systémov a softvérového inžinierstva, FIIT STU, Bratislava
Oblasť problematiky:	Personalizované odporúčanie, strojové učenie, analýza dát

Text návrhu zadania[2]

Personalizované odporúčanie na webe sa v posledných rokoch rozšírilo do mnohých odvetví komerčnej sféry. Využíva sa na odporúčanie tovarov, služieb, filmov, hudby a mnohých iných produktov konkrétnym zákazníkom alebo skupine zákazníkov s podobnými charakteristikami. Vďaka odporúčacím algoritmom sa produkty odporúčajú cieľovej skupine, čo zefektívňuje predaj a zákazníkovi pomáha s výberom čo zvyšuje používateľský zážitok.

Samotné metódy odporúčania sú často optimalizované pre všetkých používateľov webovej aplikácie. Využitie rôznych metód pre rôzne skupiny však môže potencionálne priniesť zlepšenie kvality generovaných odporúčaní.

Analyzujte súčasné spôsoby a vhodné metódy odporúčania so zameraním na personalizované odporúčanie jednotlivcom alebo skupinám podobných používateľov vo vybranej aplikačnej doméne. Analyzuje spôsoby výberu optimálnej odporúčacej metódy pre používateľa s cieľom dosiahnutia kvalitatívne lepších výsledkov.

Navrhnite metódu na odporúčanie produktov na základe analýzy dostupných informácií o používateľoch a produktoch. Navrhnite spôsob adaptívneho výberu odporúčacej metódy pre konkrétneho používateľa alebo skupinu s podobnými charakteristikami. Dosiahnuté výsledky vyhodnoťte na netriviálnej dátovej sade. Preskúmajte vlastnosti navrhnutej metódy v kontexte adaptívneho výberu odporúčacích metód.

---

[1] Vytlačiť obojstranne na jeden list papiera

[2] 150-200 slov (1200-1700 znakov), ktoré opisujú výskumný problém v kontexte súčasného stavu vrátane motivácie a smerov riešenia

Literatúra[3]

- Amatriain, X., Basilico, J.: Past, Present, and Future of Recommender Systems: An Industry Perspective. In: Proceedings of the 10th ACM Conference on Recommender Systems. ACM, 2016. p. 211-214.
- Kabbur, S., Ning, X., Karypis, G.: Fism: factored item similarity models for top-n recommender systems. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2013. p. 659-667.

Vyššie je uvedený návrh diplomového projektu, ktorý vypracoval(a) Bc. Peter Tibenský, konzultoval(a) a osvojil(a) si ho Ing. Michal Kompan, PhD. a súhlasí, že bude takýto projekt viesť v prípade, že bude pridelený tomuto študentovi.

V Bratislave dňa 17.1.2018

Podpis študenta

Podpis výskumníka

Vyjadrenie garanta predmetov Diplomový projekt I, II, III

Návrh zadania schválený: áno / ~~nie~~[4]

Dňa: 12. 2. 2018

Podpis garanta predmetov

[3] 2 vedecké zdroje, každý v samostatnej rubrike a s údajmi zodpovedajúcimi bibliografickým odkazom podľa normy STN ISO 690, ktoré sa viažu k téme zadania a preukazujú výskumnú povahu problému a jeho aktuálnosť (uveďte všetky potrebné údaje na identifikáciu zdroja, pričom uprednostnite vedecké príspevky v časopisoch a medzinárodných konferenciách)

[4] Nehodiace sa prečiarknite

Čestné prehlásenie

Čestne prehlasujem, že som túto prácu vypracoval samostatne na základe konzultácií s vedúcim práce, vlastných znalostí nadobudnutých počas štúdia a odbornej literatúry, ktorá je uvedená v závere tejto práce.

V Bratislave, apríl 2019

.....

Bc. Peter Tibenský

Poďakovanie

Touto cestou sa chcem poďakovať vedúcemu tejto práce, doc. Ing. Michalovi Kompanovi, PhD. za jeho odborné rady, konzultácie a pripomienky, ktoré mi umožnili vypracovať túto prácu.

Ďalej by som sa chcel poďakovať Ing. Petrovi Gašparovi za poskytnutie prístupu k serveru Ganimedes, na ktorom som mohol vykonávať výpočty súvisiace s touto prácou.

Anotácia

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ
Študijný program: Inteligentné softvérové systémy

Autor: Bc. Peter Tibenský
Diplomová práca: Personalizované odporúčanie produktov používateľom
Vedúci diplomovej práce: doc. Ing. Michal Kompan PhD.
apríl 2019

Táto práca sa venuje problematike personalizovaného odporúčania používateľom v doméne multimediálneho obsahu so zameraním na adaptívny výber odporúčacej metódy pre konkrétneho používateľa alebo skupinu používateľov. Konvenčné realizácie odporúčacích systémov využívajú metódy, ktoré sú optimalizované pre všetkých používateľov a to vzhľadom na rôznorodosť používateľov môže limitovať výslednú presnosť odporúčania. Kvalita generovaných odporúčaní v oblasti komercie je kľúčová pre predaj produktov, ako aj používateľský zážitok.

V práci sa zameriavame na vytvorenie hybridnej metódy odporúčania, ktorá by pre používateľa, alebo skupinu používateľov s podobnými charakteristikami, vybrala optimálnu metódu odporúčania s cieľom dosiahnutia kvalitatívne lepších výsledkov, ako použitie konvenčného spôsobu odporúčania. Ukázali sme, že na základe preferencií používateľa a kontextu je možné predpovedať, ktorá metóda odporúčania bude danému používateľovi odporúčať najpresnejšie. V doméne multimediálneho obsahu sa špecializujeme na odporúčanie filmov využitím verejne dostupnej dátovej sady. V rámci práce sa venujeme analýze rôznych techník odporúčania, aktuálnemu stavu problematiky, spôsobom evaluácie odporúčacích algoritmov a návrhu a realizácii vlastnej adaptívnej odporúčacej metódy založenej na prepínacom hybridnom odporúčaní, vrátane výberu vhodných atribútov, ktoré budú vstupom pre našu metódu.

Annotation

Slovak University of Technology Bratislava
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES
Degree course: Intelligent Software Systems

Author: Bc. Peter Tibenský
Master's thesis: Personalized product recommendation for users
Supervisor: doc. Ing. Michal Kompan PhD.
2019, April

This work deals with the issue of personalized recommendations to users in the domain of multimedia content, focusing on the adaptive selection of a recommendation method for a particular user or group of users. Conventional implementations of recommender systems use methods that are optimized for all users and can, due to diversity of users, limit the resulting accuracy of the recommendation. The quality of recommendations in the field commerce is key to product sales as well as user experience.

In this work, we focus on creating a hybrid recommendation method that would choose the most appropriate recommendation method for a user or group of users with similar characteristics to achieve qualitatively better results than to use a conventional recommendation method. We show that based on user preferences and context, it is possible to predict which recommender method will provide the most precise recommendation to user. In the domain of multimedia content, we specialize in movie recommendation by using a publicly available data set. In the course of this work we analyse the various recommendations techniques, state of the art, evaluation metrics and design and implementation of our own adaptive recommendation method based on the switching hybrid recommendation, including the selection of suitable attributes that will be the input for our method.

Obsah

1 Úvod

Odporúčacie systémy zmenili spôsob, akým internetoví predajcovia ponúkajú svoje produkty zákazníkom. Kým kedysi predajcovia nemali takmer žiadne informácie o zákazníkoch, ktorým ponúkali produkty, dnes vznikajú systémy zamerané na personalizované odporúčania produktov. Takéto odporúčacie systémy zvyšujú efektívnosť ponúkania produktov zákazníkom tak, aby sa zákazníkovi zobrazovali produkty, o ktoré má záujem. Odporúčacie systémy tiež riešia problém informačného zahltenia, ktorý v súčasnosti postihuje veľké množstvo domén. Dnes sa odporúčacie systémy využívajú v rôznych odvetviach, v ktorých vznikla potreba separovať relevantné informácie pre používateľa vzhľadom na jeho preferencie. Môže ísť o odporúčanie novinových článkov, hudby, videí, kontaktov na sociálnych sieťach, dovoleniek a mnoho iného.

Vzhľadom na informácie, ktoré o používateľoch a produktoch máme, môžeme na odporúčanie využiť rôzne techniky. Kým technika odporúčania založená na obsahu využíva najmä opis produktov, kolaboratívne filtrovanie hľadá podobnosti v preferenciách medzi používateľmi. Okrem týchto dvoch základných prístupov existuje mnoho ďalších, špeciálnym prípadom je hybridné odporúčanie. Hybridné odporúčanie kombinuje viaceré techniky odporúčania s cieľom zabezpečiť presnejšie výsledky odporúčania. Kombinovaním viacerých techník dochádza k eliminovaniu nevýhod jednotlivých techník, čo umožňuje dosiahnutie lepších výsledkov.

V tejto práci sa zameriavame na vyvinutie hybridnej metódy personalizovaného odporúčania, ktorá individuálne pre každého používateľa, alebo skupinu s podobnými charakteristikami, dokáže adaptívne vybrať najpresnejšiu metódu odporúčania vzhľadom na aktuálny stav používateľa. Ukážeme, že na základe preferencií používateľa a kontextu je možné predpovedať, ktorá metóda odporúčania mu bude odporúčať nepresnejšie. V práci analyzujeme atribúty, na základe ktorých je možné túto predikciu vykonať a vyhodnotíme vlastný hybridný systém adaptívneho výberu odporúčacej metódy, ktorého výsledky porovnáme s výkonnosťou integrovaných odporúčacích metód.

Kapitola 2 sa venuje analýze aktuálneho stavu problematiky, opisuje techniky odporúčania ako aj ich využitie v praxi, v kapitole 3 definujeme cieľ práce, v kapitole 4 sa venujeme návrhu metódy na adaptívny výber odporúčacej metódy a v 0. kapitole tento návrh realizujeme. 6. kapitola sa zaoberá overením navrhnutej metódy, dôležitosti čŕt v kontextovo-preferenčnom modeli používateľa ako aj analýze nesprávne klasifikovaných používateľov.

2 Personalizované odporúčanie

Personalizované odporúčanie sa zameriava na odporúčanie produktov používateľom, pričom zohľadňuje preferencie používateľa. Používateľ môže svoje preferencie vyjadriť buď explicitne, napríklad ohodnotením produktu, alebo implicitne pozorovaním správania používateľa v aplikácii [1]. Na základe údajov, ktoré má odporúčací systém k dispozícii, môže generovať odporúčania pre používateľov. V literatúre sa používateľ, ktorému systém generuje odporúčania, nazýva aktívny, prípadne cieľový používateľ [2, 3]. V doméne personalizovaného odporúčania existuje niekoľko prístupov k odporúčaniu. Každá technika využíva rozdielne údaje o používateľoch a produktoch na generovanie odporúčaní. Nasledujúce podkapitoly sa venujú jednotlivým technikám odporúčania, ich výhodám, nevýhodám a spôsobu evaluácie presnosti odporúčania.

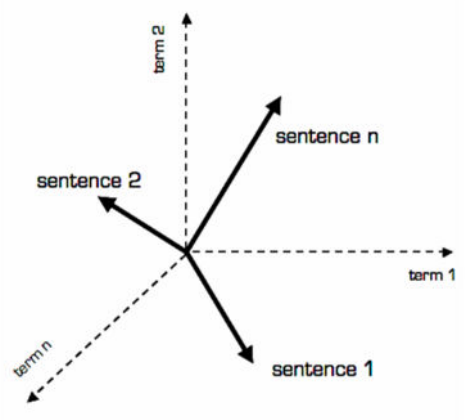
2.1 Odporúčanie založené na obsahu

Odporúčanie založené na obsahu (angl. Content-based recommendation, CB) je jednou zo základných techník odporúčania. Používateľovi sú odporúčané produkty podobné tým, ktoré v minulosti hodnotil [3]. Vstupom pre tento typ odporúčacích metód sú vlastnosti produktov a spätná väzba používateľa [3]. Pod vlastnosťami produktov sa myslia ich atribúty, ktoré môžu byť v textovej podobe, ako napríklad textový opis produktov alebo kľúčové slová. Okrem tejto informácie môže používateľ poskytnúť explicitnú spätnú väzbu v podobe hodnotenia produktov a implicitnú v podobe akcií, ktoré používateľ v systéme vykonal [4, 3].

Dáta pre odporúčanie založené na obsahu bývajú častokrát neštruktúrované a obsahujú nadbytočné informácie, ktoré pre odporúčanie nie sú podstatné. Takéto dáta sú preto predspracované a transformované do formátu vhodného na spracovanie odporúčacou metódou. Autori [3] uvádzajú niekoľko spôsobov predspracovania dát a extrahovania vlastností (angl. feature extraction). Podľa nich je najčastejším spôsobom výberu vlastností z neštruktúrovaných dát zvolenie kľúčových slov, nakoľko textový opis produktov je najbežnejší spôsob ich reprezentácie v rôznych doménach. Týmto kľúčovým slovám sú pridelené váhy vyjadrujúce ich dôležitosť v odporúčacom procese. Ďalším krokom pri spracovaní dát je ich čistenie, kde dochádza k odstráneniu slov, ktoré nemajú špecifický význam pre konkrétny produkt (angl. stop-words), ako napríklad frekventované slová daného jazyka. Vo fáze čistenia sa tiež kontrolujú variácie rovnakých slov a ponecháva sa len ich spoločný základ (angl. stemming). Pozor si tiež treba dať na synonymá a frázy, kde skupina slov nesie iný význam, ako jednotlivé slová oddelene. Na všetky vyššie spomenuté operácie existujú nástroje, ktoré túto funkcionalitu poskytujú.

Získané vlastnosti (resp. črty, angl. features) dokumentov sú následne uložené do modelu, ktorý dokáže tieto vlastnosti vhodne reprezentovať. Jedným z takýchto modelov je priestorový model vektorov (angl. vector space model). Jeho príklad je znázornený na

obrázku Obrázok 2.1. Priestorový model vektorov je multidimenzionálny model, v ktorom jeho osy predstavujú slová (angl. terms) a vektory jednotlivé dokumenty pozostávajúce zo slov [5]. Pod slovami môžeme rozumieť ľubovoľnú vlastnosť dokumentu, konkrétne pri filmoch to môže byť napríklad žáner alebo kľúčové slovo.



Obrázok 2.1: Priestorový model vektorov

Podobnosť medzi dvoma dokumentami reprezentovanými priestorovým modelom vektorov je vyjadrená vzdialenosťou vektorov týchto dokumentov. Táto vzdialenosť sa najčastejšie počíta ako kosínus uhlov vektorov, ktoré porovnávame [1]:

$$\cos(v_1, v_2) = \frac{v_1 \bullet v_2}{\|v_1\|\|v_2\|} \quad (1)$$

kde v_1 a v_2 predstavujú vektory dokumentov, ktoré porovnávame, operácia • predstavuje skalárny súčin a $\|v_1\|$ vektorovú normu. Ak pri tvorbe vektorov chceme okrem frekvencie slov v dokumente brať do úvahy aj to, ako dôležité je konkrétne slovo v konkrétnom dokumente, potrebujeme zaviesť novú techniku. Táto technika sa nazýva TF-IDF (angl. Term Frequency – Inverse Document Frequency) a ide o bežnú váhovaciu techniku, ktorá vyjadruje dôležitosť slova v dokumente [5]. Dôležitosť slova rastie úmerne s počtom výskytov v dokumente a klesá s rastom frekvencie výskytov skrz všetky dokumenty [5]. TF v tejto technike predstavuje, koľkokrát sa slovo t_i vyskytuje v dokumente d_j. Vypočíta sa nasledovne [5]:

$$TF(t_{ij}) = \frac{N(t_i, d_j)}{N(d_j)} \quad (2)$$

Pričom $N(t_i, d_j)$ vyjadruje počet výskytov slova t_i v dokumente d_j. IDF vyjadruje dôležitosť slova. Slovo je dôležitejšie, ak sa vyskytuje v menšom počte dokumentov. IDF sa počíta ako pomer celkového počtu dokumentov $N(D)$ k počtu dokumentov, v ktorých sa nachádza slovo t_i [5]:

$$IDF(t_i) = \log \frac{N(D)}{N(t_i, D)} \quad (3)$$

Z rovnice vyplýva, že čím menší je počet dokumentov, v ktorých sa vyskytuje slovo t_i, tým je slovo t_i dôležitejšie. TD-IDF sa vypočíta ako súčin oboch zložiek tejto metodiky, pričom, aby sa zachovala rovnaká dĺžka všetkých vektorov, výsledná hodnota sa ešte normalizuje do intervalu <0, 1> prostredníctvom kosínusovej normalizácie [1, 5]:

$$w_{TF-IDF}(t_{i\,j}) = \frac{TF(t_{ij}) * IDF(t_i)}{\sqrt{\sum_{j=1}^{n}\left[TF(t_{ij}) * IDF(t_i)\right]^2}} \quad (4)$$

V metóde opísanej autormi [6] je na odporúčanie športovo zameraných novinových článkov využívané obsahové odporúčanie a kolaboratívne filtrovanie. CB časť odporúčacieho systému využíva kľúčové slová z článkov, ktoré aktívny používateľ čítal. Tieto kľúčové slová vyjadrujú používateľove preferencie a sú použité na odporúčanie ďalších článkov. Algoritmus pre každý článok ukladá päť až desať kľúčových slov. Tieto slová ukladá spoločne s ich počtom výskytov v článkoch, ktoré používateľ čítal. Počty výskytov slúžia ako váhy pre kľúčové slová, z ktorých je neskôr možné určiť najrelevantnejšie články výpočtom podobnosti. Zoznam podobností je zoradený a 50 najrelevantnejších článkov slúži ako vstup pre druhú odporúčaciu metódu.

V doméne odporúčania filmov dosiahli autori článku [7] niekoľkokrát vyššiu presnosť odporúčania, ako iné CB metódy využívajúce žáner filmu. Autori z upútaviek filmov extrahovali dodatočné informácie ako osvetlenie, farebnosť, pohyb a iné vizuálne vlastnosti charakterizujúce estetickú stránku filmu. Na odporúčanie autori použili algoritmus nazývaný „Factorization Machines - FM“, ktorý je kombináciou algoritmov SVM a faktorizačných modelov. Tento algoritmus je všeobecne použiteľný na predikciu a osvedčil sa aj v doméne CB odporúčania. FM na predikciu používa kombináciu latentných faktorov a vizuálnych vlastností.

2.2 Kolaboratívne filtrovanie

Ďalšia technika personalizovaného odporúčania je založená na predpoklade, že existujú skupiny používateľov s rovnakými preferenciami. Technika kolaboratívneho filtrovania (angl. Collaborative filtering, CF) využíva na generovanie odporúčaní profil aktívneho používateľa a profily ostatných používateľov [8]. Profil používateľa typicky obsahuje osobné hodnotenia produktov, z ktorých sa aktívnemu používateľovi odporúča. Existujú tri základné prístupy používané pri kolaboratívnom filtrovaní [3]:

- Metódy založené na pamäti (angl. Memory-based methods)
  - Založené na používateľovi (angl. User-based methods)
  - Založené na produkte (angl. Item-based methods)
- Metódy založené na modeli (angl. Model-based methods)
  - Predikcia pravdepodobnosti (angl. probability prediction)
  - Predikcia hodnotenia (angl. rating prediction)
- Hybridné metódy – pozostávajú z kombinácie predošlých dvoch metód

2.2.1 Metódy založené na pamäti

Vyššie spomenuté odporúčacie metódy môžu byť využité na top-N odporúčanie. Top-N odporúčanie predstavuje algoritmy umožňujúce generovanie usporiadaného zoznamu N produktov, ktoré sú pre používateľa najrelevantnejšie [9]. Pri využití metód založených

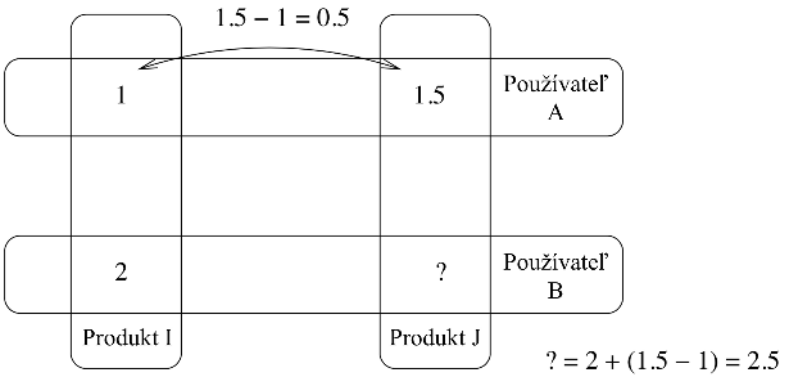
na pamäti sa buď pre konkrétneho používateľa určia podobní používatelia a následne sa odporučí top-N relevantných produktov (založené na používateľovi), alebo sa pre produkty, ktoré používateľ hodnotil, identifikujú podobné produkty, z ktorých sa potom odporučia tie najrelevantnejšie (založené na produkte) [10]. Podobnosti sú vypočítané prostredníctvom štatistických metód, v ktorých sa používa miera podobnosti medzi dvojicami používateľov [11]. Metódy odporúčania založené na pamäti na jednej strane umožňujú rýchle a jednoduché generovanie odporúčaní, na druhej neposkytujú kvalitné výsledky odporúčania a majú obmedzenú škálovateľnosť na veľkých dátových sadách [12, 2].

Jednou zo základných metód používaných pri tejto technike odporúčania je metóda *k* najbližších susedov (angl. k nearest neighbors) [13]. V metódach založených na používateľovi sa aktívnemu používateľovi hľadá používateľov s podobnou históriou hodnotení produktov. Na výpočet podobnosti môže byť použitý napríklad Pearsonov korelačný koeficient [14]. Pomocou neho je možné určiť lineárnu závislosť medzi dvoma atribútmi, v tomto prípade medzi dvoma používateľmi [2]. Spôsob výpočtu Pearsonovej korelácie medzi používateľom u a v objasňuje vzorec 5 [2].

$$w_{u,v} = \frac{\sum_{i \in I}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{i \in I}(r_{v,i} - \bar{r}_v)^2}} \tag{5}$$

V tomto vzorci $r_{u,i}$ predstavuje hodnotenie produktu i z množiny produktov I používateľom u. Priemerné hodnotenie používateľa u je označené ako $\bar{r}_u$. Pearsonov korelačný koeficient je taktiež možné použiť v metódach založených na produkte k výpočtu podobnosti dvoch produktov. V takom prípade sa vo výpočte rola používateľa a produktu zamení.

Algoritmus Slope One [11] je založený na produkte, využívajúci hodnotenia používateľov, ktorí hodnotili rovnaký produkt ako aktívny používateľ a hodnotenia produktov aktívneho používateľa. Slope One na predikciu hodnotenia využíva priemerný rozdiel medzi hodnoteniami produktov používateľov, ktorí hodnotili rovnaké produkty. Základný princíp výpočtu predikovaného hodnotenia vysvetľuje nasledujúci obrázok.



Obrázok 2.2: Princíp algoritmu Slope One [11]

Výpočet hodnotenia produktu i používateľom u objasňuje vzorec 6.

$$P(u)_i = \mu_u + \frac{1}{|R_i(u)|} \sum_{j \in R_i(u)} dev(i,j) \quad (6)$$

Pričom μ_u je priemerné hodnotenie používateľa u; $|R_i(u)|$ je počet produktov ohodnotených používateľom u, ktoré boli zároveň ohodnotené používateľmi, ktorí hodnotili produkt i; $dev(i,j)$ predstavuje priemerný rozdiel medzi hodnoteniami produktov i a j. Podľa autorov je tento algoritmus jednoduchý na implementáciu, dynamicky aktualizovateľný, efektívny v čase odporúčania a aj pri používateľoch s nízkym počtom hodnotení dokáže poskytnúť presné hodnotenia (MAE 0.752 na dátovej sade MovieLens).

Ďalší algoritmus založený na pamäti [15] využíva na predikciu hodnotení zhlukovanie matice používateľov a hodnotení. Algoritmus Co-Clustering zhlukuje ako používateľov, tak aj produkty (riadky aj stĺpce matice hodnotení) do skupín a počíta priemerné hodnotenia pre každý zhluk. Okrem zhlukov používateľov a produktov vytvára aj spoločný zhluk (tzv. co-cluster), pre ktorý vypočíta priemerné hodnotenie každého zhluku produktov každým zhlukom používateľov. Formálny zápis výpočtu hodnotenia objasňuje vzorec 7.

$$\hat{A}_{ui} = A_{gh}^{COC} + \left(A_u^R - A_g^{RC}\right) + \left(A_i^C - A_h^{CC}\right) \quad (7)$$

Premenná g reprezentuje zhluk, v ktorom sa nachádza používateľ u; h reprezentuje zhluk, v ktorom sa nachádza produkt i; A_{gh}^{COC} predstavuje priemerné hodnotenie spoločného zhluku pre zhluk používateľov g a zhluk produktov h; A_u^R a A_i^C sú priemerné hodnotenia používateľa u, respektíve produktu i; A_g^{RC} a A_h^{CC} sú priemerné hodnotenia zhlukov, do ktorých sa zaraďuje používateľ u, respektíve produkt i. Autori evaluovali algoritmus na vlastnej dátovej sade a MAE algoritmu porovnali s niekoľkými existujúcimi algoritmami. Výsledky sú na nasledujúcom obrázku.

Data	SVD	NNMF	CORR	COCLUST
Mov1	0.83 ± 0.06	0.82 ± 0.04	0.84 ± 0.04	0.84 ± 0.06
Mov2	0.83 ± 0.03	0.85 ± 0.03	0.84 ± 0.02	0.82 ± 0.03
Mov3	0.82 ± 0.03	0.83 ± 0.03	0.85 ± 0.02	0.81 ± 0.03

Obrázok 2.3: MAE algoritmu Co-Clustering v porovnaní s inými algoritmami [15]

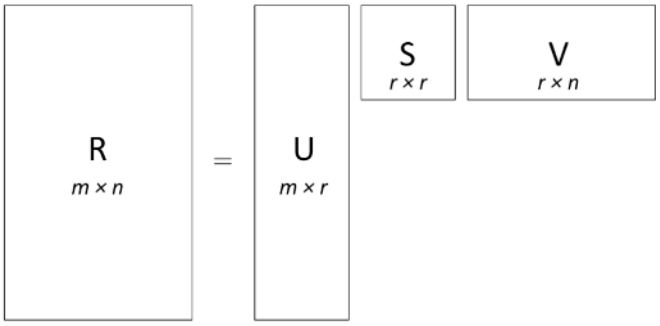
2.2.2 Metódy založené na modeli

Metódy založené na modeli vytvárajú prostredníctvom metód strojového učenia model, ktorý používajú na odporúčanie. Jedným zo spôsobov vytvorenia modelu je vytvorenie matice používateľov a produktov a jej faktorizácia na používateľskú zložku a produktovú zložku. Zložky predstavujú preferencie používateľa, prípadne charakteristiku produktu. Po vynásobení oboch zložiek matice vznikne pôvodná matica. Tento spôsob sa nazýva faktorizácia matice (angl. Matrix Factorization, MF) a pri odporúčaní sa používa

vypočítané predikované hodnotenie produktov. Existuje niekoľko algoritmov na faktorizáciu matice, napríklad:

- SVD (angl. Singular Value Decomposition) [16] a jeho variácie (SVD++ [17], PureSVD [12], TimeSVD++ [18], TrustSVD [19]),
- WRMF (angl. Weighted Regularized MF) [20],
- MMMF (angl. Max-Margin MF) [21],
- WNNMF (angl. Weighted Non-Negative MF) [22].

Algoritmus rozkladu singulárnej hodnoty, SVD, opísaný v článku [16], je jedna z techník faktorizácie matice. Tento algoritmus faktorizuje maticu používateľov a produktov R o veľkosti $m \times n$ na tri matice $R = U \cdot S \cdot V^T$. Matice U a V sú ortogonálne a majú rozmery $m \times m$, respektíve $n \times n$. Matica S je diagonálna, má rozmer $m \times n$ a na diagonále obsahuje singulárne hodnoty matice R. Počet singulárnych hodnôt na diagonálne je rovný hodnosti[1] matice R, označuje sa r. Algoritmus SVD redukuje rozmery matíc podľa r, čím vznikne matica $R_r = U_r \cdot S_r \cdot V_r^T$. Rozmery týchto matíc opisuje Obrázok 2.4. Hodnoty na diagonále matice S_r sú zoradené zostupne a predstavujú latentný faktor medzi používateľmi a produktami. V prípade odporúčania filmov používateľom si môžeme ako latentný faktor predstaviť napríklad žáner filmu [23]. Vďaka tomu dokážeme vyjadriť vzťah používateľa k filmom, nakoľko vieme závislosť medzi nimi priamo porovnávať.



Obrázok 2.4: Faktorizácia matice algoritmom SVD [43]

Faktory v matici S_r s malou hodnotou môžu byť ignorované, čím dochádza k redukcii rozmerov matice [23]. Tým je možné rozmer matice S_r zredukovať na $k \times k$, pričom $k < r$ [16]. Tieto matice sú použité na predikciu hodnotenia produktu P používateľom C, pričom predikované hodnotenie je výsledkom skalárneho súčinu týchto matíc podľa vzorcu 8 [16].

[1] Počet lineárne nezávislých riadkov a stĺpcov v matici.

$$C_{P_{pred}} = \bar{C} + U_k \sqrt{S_k}^T(c) \cdot \sqrt{S_k} \cdot V_k^T(P) \quad (8)$$

Autori [16] na vykonanie predikcie použili normalizovanú neriedku maticu hodnotení, v ktorej prázdne miesta vyplnili priemerným hodnotením. Použitie priemerného hodnotenia produktu prinieslo lepšie výsledky, ako použitie priemerného hodnotenia používateľa. Riedkosť matice, spôsobená nedostatkom explicitnej spätnej väzby, spôsobuje zhoršenú presnosť odporúčania. Algoritmus SVD++, na rozdiel od SVD, využíva okrem explicitnej aj implicitnú spätnú väzbu [24]. Implicitná spätná väzba je v tomto prípade vyjadrená Boolovou hodnotou indikujúcou, či používateľ produkt hodnotil, bez ohľadu na výšku hodnotenia [14].

SVD a SVD++ sú statické algoritmy, ktoré neberú do úvahy časové následnosti. Vzhľadom na kontext, v ktorom sa odporúčacie algoritmy využívajú, však čas zohráva dôležitú rolu, nakoľko preferencie používateľa sa môžu časom meniť [25]. V takomto prípade je vhodné využiť časovú informáciu pre presnejšie generovanie odporúčaní na základe aktuálnych preferencií používateľa. Algoritmus TimeSVD++ berie tieto zmeny do úvahy. Podľa článku [18] medzi parametre, ktoré sa časom menia, patrí bias používateľov a produktov a vektory latentných faktorov. Tento algoritmus poskytol lepšie výsledky ako SVD++, čo zobrazuje aj nasledujúca tabuľka s meraniami chyby pre rôzny počet faktorov.

Tabuľka 2.1: Porovnanie RMSE algoritmov SVD, SVD++ a TimeSVD++ [18]

Model	f **=10**	f **=20**	f **=50**	f **=100**	f **=200**
SVD	.9140	.9074	.9046	.9025	.9009
SVD++	.9131	.9032	.8952	.8924	.8911
TimeSVD++	.8971	.8891	.8824	.8805	.8799

Faktorizácia matice spoločne s ostatnými metódami založenými na modeli, na rozdiel od metód založených na pamäti, dokáže poskytnúť presnejšie odporúčania, avšak za cenu pomalého naučenia modelu [12]. Autori [12] predstavili top-N odporúčaciu metódu SLIM (angl. Sparse LInear Method), ktorá odstraňuje nevýhody predošlých dvoch metód, umožňuje rýchle a presné odporúčanie a je možné ju využiť na odporúčanie v reálnom čase. Táto metóda počíta hodnotenie doposiaľ neohodnoteného produktu t_j používateľa u_i ako riedku agregáciu (angl. sparse aggregation) produktov, ktoré používateľ u_i v minulosti hodnotil. Autori prirovnávajú model metódy SLIM k modelu metódy itemkNN, pričom rozdielom je spôsob určenia podobnosti produktov. Kým modelom itemkNN metódy je matica kosínusovej podobnosti (angl. cosine similarity matrix), SLIM na vytvorenie modelu rieši optimalizačný problém minimalizácie riedkej matice obsahujúcej agregačné koeficienty. Čo sa týka výkonu odporúčania (angl. recommendation performance) v zmysle presnosti, metóda SLIM podľa jeho autorov prekonala top-N odporúčacie metódy ako PureSVD, WRMF a BPRkNN.

V článku [26] autori opisujú metódu GLSLIM (angl. Global and Local SLIM), ktorá rozširuje pôvodnú metódu o globálne a lokálne modely. Metóda je založená na myšlienke, že existujú skupiny používateľov, ktorí sa správajú podobne. Pre každú skupinu používateľov sa vytvorí vlastný lokálny model a ten je spoločne s globálnym modelom použitý na predikciu hodnotenia produktu. Prideľovanie používateľov do skupín je dynamické a používateľ preto môže zmeniť skupinu, čo vedie k prepočítaniu lokálneho modelu tejto skupiny, ako aj globálneho modelu. Za cenu vyššej časovej zložitosti oproti metóde SLIM dosiahli autori vyššiu presnosť.

2.3 Odporúčanie založené na znalostiach

Odporúčanie založené na znalostiach (angl. Knowledge-based, KB) sa od predošlých dvoch techník líši spôsobom získavania preferencií od používateľa. Namiesto získavania spätnej väzby v podobe histórie hodnotení iných produktov, alebo implicitnej spätnej väzby, KB získava preferencie explicitným špecifikovaním produktu používateľom [3]. Je určený najmä do domén s komplexnými produktami, ktoré je možné konfigurovať alebo inak prispôsobovať a systém pri odporúčaní využíva znalosti o tejto doméne [3].

Systém na odporúčanie turistických miest opísaný v práci [27] využíva na odporúčanie kontextuálne informácie o používateľovi, atribúty turistických miest ako meno, polohu, cenu, kategóriu a znalostné modely. Používateľovi sú odporúčané miesta na základe jeho aktuálnej polohy, rozpočtu a ďalších požiadaviek závislých od typu výletu, ktorý používateľ plánuje podniknúť. Systém MIKAS [28] používa KB na odporúčanie diét, kde výstupom odporúčacej metódy je menu odpovedajúce potrebám používateľa. Ide o prípadovo založené KB (angl. case-based), čo znamená, že používateľ vyjadrí svoje preferencie a systém z bázy prípadov vyberie relevantné a zoradí ich podľa vhodnosti, pričom najvhodnejší prípad je odporučený používateľovi. Pokiaľ ani jeden prípad nie je dostatočne vhodný, ľudský expert poskytne systému novú znalosť, ktorá je pridaná do znalostnej bázy prípadov.

Okrem prípadovo založeného KB existuje aj odporúčanie založené na obmedzeniach (angl. constraint-based) [3]. V tomto type používatelia vyjadrujú svoje požiadavky obmedzeniami hodnôt atribútov produktu. Ide napríklad o určenie rozsahu hodnôt, ktoré môže atribút produktu nadobúdať. KB tohto typu obsahuje pravidlá, ktoré vyjadrujú, ako je možné hodnoty atribútov kombinovať.

2.4 Hybridné odporúčanie

Hybridné odporúčacie systémy sú také, ktoré kombinujú viacero odporúčacích techník do jednej s cieľom kompenzovať nevýhody jednej techniky inou technikou a získať tak vyšší odporúčací výkon [4]. Burke [4] definuje sedem základných spôsobov, ako môžu byť odporúčacie techniky kombinované. Tieto techniky sú predmetom nasledujúcich podkapitol.

2.4.1 Váhované odporúčanie (angl. weighted)

Výstupy odporúčacích metód sú skombinované do jedného výstupu $\hat{R}$, pričom výstup každej metódy $\hat{R}_i$ má koeficientom pridelenú váhu α_i. Vo váhovanom odporúčaní dochádza k prepočtu hodnotenia produktov na základe váh jednotlivých použitých metód odporúčania [8]. Výsledné hodnotenie produktu je počítané podľa nasledujúcej rovnice [3].

$$\hat{R} = \sum_{i=1}^{q} \alpha_i \hat{R}_i \tag{9}$$

Podľa Burke [4] je týmto prístupom možné kombinovať CF s CB, CF s KB, alebo CB s KB, avšak vo väčšine prípadov váhované odporúčanie dosahovalo horšie výsledky, ako použitie dominantnej metódy samostatne.

Váhovaný hybridný odporúčací rámec opísaný v článku [29] sa zameriava na odporúčanie zdrojov (angl. resources) na základe ich anotácií prostredníctvom tzv. štítkov (angl. tags). Štítky, ktorými môžu používatelia označovať zdroje, predstavujú spôsob, ako tento zdroj charakterizovať a odporúčať ďalším používateľom. Autori na odporúčanie použili šesť odporúčacích komponentov, pričom každý z nich sa špecializuje na určitú dimenziu v danej doméne. Váhované odporúčanie tak umožňuje matematicky nenáročným spôsobom skombinovať výsledky viacerých odporúčacích metód, ktoré túto doménu analyzujú z rôznych strán. Ako odporúčacie komponenty boli použité metódy kolaboratívneho filtrovania (userKNN, itemKNN), model podobnosti štítkov a model populárnosti. Model podobnosti štítkov je založený na predpoklade, že frekvencia používania konkrétneho štítku používateľom vyjadruje jeho záujem o tému vyjadrenú týmto štítkom. Tento model na odporúčanie zdrojov využíva kosínusovú podobnosť medzi zdrojmi. Model populárnosti odporúča tie zdroje, ktoré sú momentálne populárne a nejedná sa tak o personalizovanú metódu odporúčania. Na určenie váh odporúčacích metód je použitá horolezecká metóda (angl. hill-climbing). Autori vykonali evaluáciu algoritmu na šiestich dátových sadách a v každej z nich dosiahol hybrid lepšie výsledky, ako jednotlivé odporúčacie komponenty. Konkrétne na dátových sadách Citeulike a MovieLens boli výsledky dramaticky lepšie.

2.4.2 Prepínacie odporúčanie (angl. switching)

Na odporúčanie je použitá práve jedna odporúčacia metóda, ktorá je zvolená na základe podmienky, ktorou sa overí spoľahlivosť (angl. confidence) metódy. Prepínacie odporúčanie môže byť navrhnuté aj ako systém primárnej a sekundárnej odporúčacej metódy. Ak spoľahlivosť primárnej metódy presiahne definovanú prahovú hodnotu, je použitá táto metóda. V opačnom prípade je použitá sekundárna metóda odporúčania.

Prepínací hybridný odporúčací systém z článku [30] sa zaoberá odporúčaním webových stránok analýzou dát ich využívania. Algoritmus zvolí najpresnejšiu odporúčaciu metódu vyhodnotením troch doménovo závislých kritérií, ktorými sa určí spoľahlivosť jednotlivých metód. Autori na správne určenie najpresnejšej metódy vykonali niekoľko

experimentov nad rôznymi dátovými sadami a ich podsadami, aby dokázali definovať spoľahlivosť odporúčacích metód v rôznych kombináciách splnenia kritérií. Na odporúčanie je využitý model založený na zhlukovaní používateľských relácií, CST model (angl. click-stream tree), Markov model a model založený na objavovaní asociačných pravidiel.

2.4.3 Kaskádové odporúčanie (angl. cascade)

Odporúčacie metódy sú zostavené do hierarchie a výsledky odporúčania primárnej metódy sú vstupom pre sekundárnu metódu. Aby sa zaručila vyššia presnosť odporúčania, ako primárna metóda sa použije tá, ktorá dosahuje lepšie výsledky odporúčania. Sekundárna metóda, častokrát dosahujúca horšiu presnosť odporúčania, odporúča iba z produktov, ktoré sú výstupom primárnej metódy. Autori [3] zovšeobecnili túto definíciu a za kaskádové odporúčanie považujú akúkoľvek metódu, kde výsledok jednej odporúčacej metódy je použitý inou metódou ľubovoľným spôsobom.

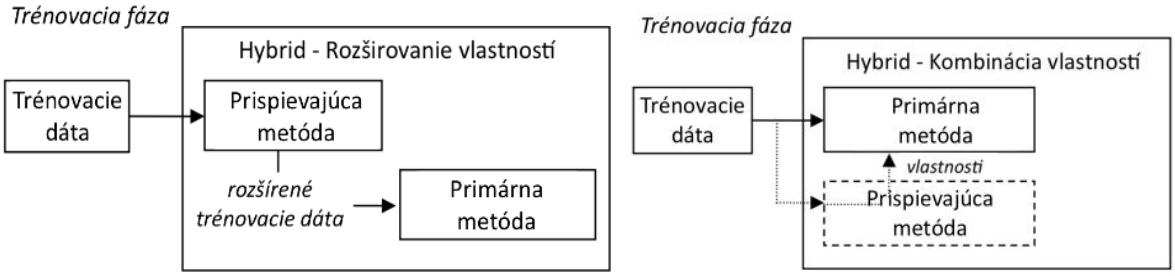
Príklad kaskádového odporúčacieho systému je opísaný v článku [31]. Tento systém vyžaduje odoslanie piesne v podobe audio súboru, ku ktorému chce používateľ získať odporúčania na ďalšie piesne. Proces odporúčania pozostáva z dvoch krokov. V prvom kroku metóda odporúčania založená na obsahu na základe audio súboru, ktorý používateľ odoslal, klasifikuje žáner piesne a vygeneruje zoznam piesní rovnakého žánru. Klasifikáciu žánru realizuje algoritmus RBF-SVM (angl. Radial Basis Function-SVM). V druhom kroku, technikou kolaboratívneho filtrovania – diagnózy osobnosti, systém nad týmto zoznamom piesní vygeneruje nový zoznam, pričom využije informáciu o hodnoteniach piesní. Hodnotenia piesní, ktoré aktívny používateľ poskytne, sa použijú na výpočet pravdepodobností vyjadrujúcich zhodu osobností medzi aktívnym používateľom a ostatnými používateľmi.

2.4.4 Rozširovanie vlastností (angl. feature augmentation)

Táto stratégia používa prispievajúcu odporúčaciu metódu a primárnu odporúčaciu metódu. Prispievajúca metóda, ako prostredník medzi dátovou sadou a primárnou metódou, obohacuje dátovú sadu o jej vlastný príspevok [8]. Táto rozšírená dátová sada je použitá primárnou odporúčacou metódou na generovanie odporúčaní.

2.4.5 Kombinácia vlastností (angl. feature combination)

Kombinácia vlastností je podobná predošlej metóde, rozdiel je však v spôsobe, akým sa prispievajúca odporúčacia metóda podieľa na obohatení vstupu pre primárnu odporúčaciu metódu. Prispievajúca metóda vlastnosti, ktoré by normálne boli použité touto metódou, posúva ako vstup primárnej metóde. Vo výsledku existuje iba jedna metóda využívajúca logiku inej odporúčacej metódy [8]. Rozdiel medzi metódou rozširovania a kombináciou vlastností objasňuje nasledujúci obrázok.



Obrázok 2.5: Porovnanie trénovacích fáz pre hybrid rozširovania a kombinácie vlastností [8]

Hybridný odporúčací systém využitý vo webovej aplikácii na odporúčanie filmov, ktorý je opísaný v článku [32], je založený na kombinácií vlastností kolaboratívneho filtrovania a odporúčacej metódy založenej na obsahu. Ako algoritmus kolaboratívneho filtrovania autori použili algoritmus najbližších susedov. Prostredníctvom kosínusovej podobnosti, prípadne Pearsonovej korelácie, vypočítali podobnosť medzi používateľmi, získali *k* najbližších susedov a na predikciu hodnotenia použili vážený priemer hodnotení týchto používateľov. Metóda založená na obsahu vytvára používateľské profily a berie do úvahy hodnotenia používateľov. Vstupom pre generovanie profilu používateľa je profil produktu v podobe vektora opisujúceho produkt a prislúchajúce hodnotenie používateľom. Používateľský profil je vypočítaný ako vážený súčet profilov produktov, kde sú ako váhy použité hodnotenia používateľov. Odporúčanie prebieha výpočtom kosínusovej podobnosti medzi profilom používateľa a profilmi produktov. Výhodou tejto hybridnej metódy je nezávislosť od profilov ostatných používateľov a schopnosť odporúčať produkty, ktoré zatiaľ neboli hodnotené.

Hybridný systém LightFM [33] používa na odporúčanie faktorizáciu matice, ktorej model reprezentuje používateľov a produkty ako lineárnu kombináciu ich latentných faktorov. Faktorizácia matice je kombinovaná s odporúčaním založeným na obsahu, ktorý redukuje vplyv studeného štartu pre nových používateľov a produkty použitím obsahových vlastností. Obsahovými vlastnosťami produktov je buď ich opis v podobe voľného textu, alebo v podobe štítkov (angl. tags), medzi ktorými hľadá podobnosť. Podľa autorov tento hybridný systém dosahuje kvalitné odporúčania v prípadoch studeného štartu a riedkej matice interakcií a dosahuje presnosť minimálne tak dobrú, ako obyčajné kolaboratívne filtrovanie založené na faktorizácii matice.

2.4.6 Meta-úroveň (angl. meta-level)

Meta-úrovňová hybridná metóda odporúčania sa podobne, ako predošlé metódy, skladá z dvoch odporúčacích metód. V tomto prípade sa naučený model jednej metódy použije ako vstup pre druhú, primárnu metódu [8, 4]. Táto metóda je podobná hybridu rozširovania vlastností, ale líši sa vstupom, ktorý prispievajúca metóda poskytne primárnej. Namiesto rozšírených trénovacích dát je ako vstup pre primárnu odporúčaciu metódu použitý samotný model prispievajúcej metódy.

2.4.7 Zmiešané odporúčanie (angl. mixed)

Zmiešané odporúčanie prezentuje odporúčania viacerých odporúčacích metód simultánne [34]. V porovnaní s váhovaným odporúčaním, kde dochádza k výpočtu hodnotenia produktov na základe váh jednotlivých metód, táto metóda kombinuje až výsledné produkty na odporúčanie, ktoré boli získané použitými metódami odporúčania.

2.5 Metódy adaptívneho výberu odporúčacej metódy

Zmiešané odporúčanie opísané v predošlej kapitole kombinuje výsledky odporúčania viacerých metód do výsledného zoznamu odporúčaných produktov. Jednotlivé stratégie odporúčania použité v takejto stratégii môžu byť rôzne efektívne v odporúčaní pre rôznych používateľov. Napríklad v doméne odporúčania filmov, používateľovi A generuje lepšie odporúčania algoritmus kolaboratívneho filtrovania, zatiaľ čo používateľovi B zameranému len na konkrétny typ filmov odporúča lepšie obsahový odporúčač. Okrem vkusu používateľa môžu pri zvolení správnej odporúčacej metódy hrať rolu aj iné faktory ako napríklad počet hodnotení a aktivita. Ďalší faktor vstupujúci do výberu vhodného algoritmu je problém nového používateľa a nového produktu, kedy sa musí odporúčací systém prispôsobovať zmenám v doméne. Takýto problém sa nazýva aj dilema preskúmavania a využívania (angl. explore-exploit dilemma). Podľa [3] v reálnom využití odporúčacích systémov do systému neustále pribúdajú noví používatelia a produkty a odporúčací systém sa preto musí prispôsobovať týmto zmenám a využívať (angl. exploit) novo získanú spätnú väzbu používateľov.

Adaptívnym výberom odporúčacej metódy sa zaoberal už Billus a Pazzani v článku [35] z roku 2000. V ňom navrhli metódu prepínacieho hybridu využívajúcu algoritmus najbližšieho suseda (NN) a naïve Bayess klasifikátor na generovanie odporúčaní na obsahové odporúčanie. Kým NN algoritmus slúži na modelovanie krátkodobého záujmu používateľa, naïve Bayess modeluje dlhodobý záujem používateľa. Model krátkodobého záujmu sa modeluje na základe n posledných hodnotení používateľa a umožňuje rýchle prispôsobenie odporúčaní reflektujúcich aktuálny záujem. V odporúčaní je NN algoritmus použitý primárne a počíta najpodobnejší produkt (článok) k produktom, ktoré používateľ hodnotil. Pokiaľ sa nepodarí nájsť adekvátnych susedov, použije sa model dlhodobého záujmu. Autori za nevýhodu tohto prístupu považujú využitie čisto algoritmov odporúčania založených na obsahu.

Ďalší z algoritmov riešiaci tento typ problému sa nazýva algoritmus viacrukého banditu (angl. multi-armed bandit, MAB) [3]. Algoritmus je v [3] vysvetlený na príklade hráča v kasíne, ktorý si musí vybrať, pri ktorom hracom automate bude hrať. Keď hráč potiahne za páku automatu, získa určitú výhru podľa pravdepodobnostného rozdelenia daného automatu. Hráč tuší, že niektoré automaty pravdepodobnostne dávajú väčšie výhry, ale nevie, ktoré z nich to sú bez toho, aby ich všetky vyskúšal. Skúšanie automatov je preskúmavaním (angl. exploration) priestoru stratégií. V kontexte odporúčania sú jednotlivé hracie automaty stratégie odporúčania a výhra automatu je ich presnosť,

prípadne iná metrika posudzujúca výkonnosť stratégie. Počas preskúmavania nemusí byť zvolená najvýkonnejšia stratégia, ale spätná väzba, ktorú získame, poslúži v ďalších iteráciách algoritmu na zvolenie vhodnejšej stratégie. Existuje niekoľko techník pre riešenie MAB problému. Jedným z nich je $\mathcal{E}$-*Greedy* algoritmus, ktorý v každej iterácii zvolí herný automat s najväčšou očakávanou výhrou s pravdepodobnosťou $1 - \mathcal{E}$ a náhodný iný herný automat s pravdepodobnosťou $\mathcal{E}$ [25]. Tento algoritmus je zameraný na nájdenie najlepšieho herného automatu (stratégie odporúčania) pri čo najmenšom počte iterácií [3].

Algoritmus viacrukého banditu využíva napríklad spoločnosť Amazon pri odporúčaní produktov v ich systéme Stream na odporúčanie ako populárnych, tak aj nových produktov [36]. Využívajú na to Thompsonove vzorkovanie. Autori [25] používajú MAB na generovanie odporúčaní na základe najaktuálnejších preferencií používateľa s detekciou náhlej zmeny preferencií používateľa. V prípade, že takúto zmenu detegujú, MAB algoritmus založený na Thompsonovom vzorkovaní začne uprednostňovať spätnú väzbu získanú po tejto zmene.

Felício et al. [37] využili algoritmus viacrukého banditu na zaradenie nového používateľa bez akejkoľvek spätnej väzby do zhluku podobných používateľov, čo im umožnilo pomocou MAB odstrániť problém studeného štartu. Ich algoritmus PdMS je možné rozdeliť do troch krokov: predikcia hodnotení, zhlukovanie preferencií a výpočet konsenzu. V prvom kroku nad trénovacou množinou obsahujúcou maticu hodnotení produktov používateľmi vykonajú faktorizáciu matice za účelom vypočítania chýbajúcich hodnotení. V druhom kroku dochádza k zhlukovaniu používateľov na základe ich hodnotení použitím dopočítanej matice hodnotení a funkcie na výpočet Euklidovskej vzdialenosti. Tretí krok, výpočet konsenzu, zahŕňa výpočet priemerného hodnotenia každého produktu každým vytvoreným zhlukom. Odporúčanie produktov prebieha nasledovne:

1. nový používateľ u je zaradený do zhluku c pomocou MAB,
2. používateľovi u sú odporučené produkty s najvyšším hodnotením pre zhluk c,
3. spätná väzba používateľa je využitá ako výška odmeny pre MAB,
4. MAB na základe spätnej väzby prepočíta svoj predikčný model.

Za nevýhodu tohto prístupu autori považujú použitie iba jednej metódy kolaboratívneho odporúčania – faktorizácie matice.

Medzi algoritmy adaptívneho odporúčania môžeme zaradiť tiež prácu [38], v ktorej autori opísali algoritmus kolaboratívneho filtrovania použitý na výber najpresnejšej metódy kolaboratívneho odporúčania pre neznámu dátovú sadu. V trénovacej fáze kolaboratívnej metódy vystupujú datasety ako používatelia a odporúčacie metódy ako produkty. Matica datasetov a metód obsahuje výkonnostné hodnotenia metód na datasetoch (v kontexte odporúčania ide o hodnotenie produktu používateľom). Vo fáze predikcie sa z datasetu,

ku ktorému sa odporúča najvhodnejší algoritmus, vypočíta N podvzorkovacích značiek (angl. subsampling landmarkers). Tie slúžia na vyriešenie problému studeného štartu, nakoľko pre nový dataset neexistujú zatiaľ žiadne výkonnostné hodnotenia. Po použití N podvzorkovacích značiek sa predikuje výkonnosť ostatných algoritmov.

Článok [39] sa zaoberá odporúčaním najvhodnejšieho algoritmu zhlukovania pre novú dátovú sadu pomocou meta-učenia (angl. meta-learning). Pod týmto pojmom autori rozumejú využitie čŕt, ktoré extrahujú z novej dátovej sady, poradia algoritmov podľa ich výkonu nad trénovacou vzorkou dátových sád, a meta-algoritmov na naučenie závislostí medzi nimi. Črty, ktoré používajú, nie sú závislé od označení objektov v dátovej sade, nakoľko využívajú informácie ako počet objektov, počet atribútov objektu, proporcie atribútov, korelácie medzi atribútmi a ich entropiu. Ako meta-algoritmy vyskúšali algoritmy KNN, naïve Bayess, viacvrstvový perceptron a rozhodovací strom, pričom KNN dosahoval najpresnejšie výsledky.

Prepínací hybrid predstavený v [40] kombinuje kolaboratívne filtrovania založené na produkte a klasifikačné prístupy (naïve Bayess, SVM) naučené na profiloch používateľov slúžiace na obsahové odporúčanie. Prepínanie prebieha na základe podmienky, do ktorej vstupujú dva parametre určujúce rizikovosť predikcie konkrétnou metódou. Riskantná predikcia znamená, že presnosť odporúčania danou metódou bude nízka. Prvým parametrom je výstup klasifikačnej metódy – pravdepodobnosť každej triedy a druhým parametrom je počet hodnotení produktu, pričom ak je odporučený málo hodnotený produkt, riziko je vyššie ako pri viac hodnotenom produkte.

2.6 Problémy odporúčacích techník

Každá odporúčacia technika má svoje výhody a limitácie, ktoré je dôležité zvážiť pri výbere vhodnej techniky pre konkrétnu doménu. V nasledujúcej časti sú uvedené niektoré problémy, ktoré postihujú odporúčacie systémy. Tabuľka 2.2 zobrazuje výhody a nevýhody jednotlivých techník odporúčania.

Tabuľka 2.2: Výhody a nevýhody odporúčacích techník

Technika	Výhody	Nevýhody
Odporúčanie založené na obsahu (CB)	Schopnosť odporúčať aj bez dostupného hodnotenia produktov	Obmedzená analýza obsahu
	Prispôsobenie sa zmeneným preferenciám používateľa	Studený štart (problém nového používateľa)
	Nezávislosť používateľov	Úzka špecializácia
	Schopnosť odporúčať nové produkty	Riedkosť dát

Technika	Výhody	Nevýhody
Kolaboratívne filtrovanie (CF)	Nezávislosť od metadát produktov Nezávislosť od domény	Studený štart Riedkosť dát Škálovateľnosť Problém sivej ovce
Odporúčanie založené na znalostiach (KB)	Žiaden studený štart Spoľahlivejšie odporúčania	Potreba získavania znalostí

2.6.1 Studený štart

Problém studeného štartu spočíva v ťažkosti odporúčania nových produktov, ktoré zatiaľ nikto nehodnotil, alebo odporúčania novým používateľom, o ktorých systém neeviduje žiadne preferencie v podobe explicitnej alebo implicitnej spätnej väzby, alebo eviduje nedostatočný počet [14, 41]. Tento problém zastihuje najmä systémy založené na technike kolaboratívneho filtrovania, ktoré vyžadujú maticu hodnotení produktov. Niektoré systémy tento problém riešia dotazníkom, ktorým od používateľa získa potrebné preferencie. Iné systémy obchádzajú studený štart hybridnými metódami, napríklad s využitím obsahového odporúčania [1].

2.6.2 Riedkosť dát

Riedkosť (angl. sparsity) sa v odporúčacích systémoch vyskytuje najmä v podobe riedkosti matice produktov a používateľov, ktorá je používaná v metódach kolaboratívneho filtrovania [41]. Riedkosť matice spočíva v prítomnosti veľkého množstva prázdnych (nulových) hodnôt. Jedným z dôvodov riedkosti matice produktov a používateľov je fakt, že väčšina používateľov nehodnotila väčšinu produktov a tak v takejto matici chýbajú hodnotenia. Okrem toho sa tento problém vyskytuje v počiatočných fázach systému, kedy hodnotenia nie sú k dispozícii.

2.6.3 Škálovateľnosť

S rastúcim počtom produktov a používateľov potrebuje systém viac prostriedkov pre generovanie odporúčaní pri zachovaní rovnakej rýchlosti spracovania dát. Čas natrénovania modelu, čas predikcie a pamäťové požiadavky sú základné miery, ktoré slúžia na určenie škálovateľnosti [3].

2.6.4 Úzka špecializácia

Úzka špecializácia (angl. overspecialization) zabraňuje používateľovi objavovať nové produkty, pretože mu odporúča iba produkty podobné tým, ktoré v minulosti hodnotil. Tento problém postihuje systémy založené na obsahovom odporúčaní [14]. Riešením pre tento typ problému je skombinovanie s kolaboratívnym filtrovaním v podobe hybridného prístupu.

2.6.5 Obmedzená analýza obsahu

Obmedzená analýza obsahu (angl. limited content analysis) je problém vyskytujúci sa pri obsahovom odporúčaní. V ňom presnosť a kvalita odporúčania závisí na metadátach produktov a s rastúcou kvalitou opisu produktu rastie aj kvalita odporúčania [42]. Ak odporúčací systém nemá dostatočne veľa informácií o produkte, nedokáže tento produkt správne odporučiť [14].

2.6.6 Sivá ovca

Problém sivej ovce, ktorý zasahuje najmä metódy kolaboratívneho filtrovania, vyjadruje situáciu, kedy nie je možné používateľovi nájsť používateľov s podobnými preferenciami, pretože preferencie aktívneho používateľa nekorelujú so žiadnou skupinou používateľov [2].

2.7 Evaluácia metód odporúčania

Aby bolo možné metódy odporúčania medzi sebou porovnávať, bolo zavedených niekoľko metodológií a metrík výpočtu presnosti odporúčacích metód. Evaluácia metód odporúčania je dôležitá v prípade, ak potrebujeme vybrať najvhodnejšieho kandidáta do odporúčacieho systému pre konkrétnu doménu, alebo vyhodnotiť výkon metódy. Vhodnosť kandidáta možno posudzovať na základe rôznych požiadaviek, ktoré sú kladené v doméne na odporúčacie metódy, či už ide o presnosť, alebo časovú zložitosť. Pri výbere vhodnej metódy musíme najprv určiť, ktoré vlastnosti metód je pre túto doménu dôležité sledovať [43]. Medzi tieto vlastnosti patrí napríklad presnosť, robustnosť, alebo škálovateľnosť. Autori [43] definujú tri spôsoby, ako môžu byť experimenty pre evaluáciu odporúčacích metód vykonané:

1. *Off-line evaluácia*. V tomto type experimentu nie je vyžadovaná žiadna interakcia s používateľom. Na vykonanie experimentu sa používa vopred pripravená dátová sada používateľov, produktov a ich hodnotení. Tieto dáta sú použité na simuláciu správania používateľov interagujúcich s odporúčacím systémom pri predpoklade, že správanie používateľov bude dostatočne podobné aj po nasadení systému do prevádzky. Off-line experimenty, v porovnaní s nasledujúcimi dvoma typmi, sú najjednoduchšie na vykonanie.
2. *Používateľské štúdie*. Vstupné dáta niektorých odporúčacích systémov zahŕňajú interakciu používateľa so systémom. Z tohoto dôvodu je off-line evaluácia zložitá na simulovanie a do experimentu je potrebné zahrnúť niekoľko skutočných používateľov. Počas takéhoto experimentu sú testovacie subjekty požiadané o vykonanie rôznych úloh, pričom sú zaznamenávané rôzne vlastnosti potrebné na evaluáciu odporúčacieho systému.
3. *On-line evaluácia*. Tento typ experimentu sa zameriava na porovnanie interakcie skutočných používateľov s rôznymi odporúčacími systémami. Jeden zo spôsobov realizácie on-line evaluácie v reálnom nasadení je presmerovanie časti

používateľov na testovaný odporúčací systém a porovnanie interakcie používateľov naprieč porovnávanými systémami. On-line evaluácia umožňuje priamo sledovať splnenie cieľov, pre ktoré bol odporúčací systém navrhnutý.

Nasledujúca časť tejto podkapitoly sa zaoberá spôsobmi merania presnosti odporúčacích systémov. Presnosť je možné merať buď overením správnosti predpovedaného hodnotenia, alebo overením, že si odporučený produkt používateľ naozaj vybral.

2.7.1 Meranie presnosti predikcie hodnotenia

V odporúčacích systémoch, ktoré pri odporúčaní predpovedajú hodnotenie produktu používateľom, je možné využiť niektorú z metrík na meranie presnosti predikcie hodnotenia. Medzi tieto metriky patrí RMSE (angl. Root Mean Squared Error), MAE (angl. Mean Absolute Error) a ich variácie. Na výpočet RMSE a MAE potrebujeme okrem predpovedaných hodnotení poznať aj skutočné hodnotenia produktov, aby sme dokázali overiť úspešnosť predikcie.

RMSE predstavuje výpočet chyby v predikcii, kde rozdiel medzi skutočným a predpovedaným hodnotením je umocnený. Umocnenie rozdielov v hodnoteniach spôsobí väčšiu penalizáciu tých chýb, ktoré sú väčšie [43]. Vzorec na výpočet RMSE je nasledovný [43]:

$$RMSE = \sqrt{\frac{1}{|T|}\sum_{(u,i)\in T}\left(\hat{r}_{u,i} - r_{u,i}\right)^2} \tag{10}$$

T predstavuje testovaciu množinu dát, u používateľa, i produkt, $\hat{r}_{u,i}$ je predpovedané hodnotenie a $r_{u,i}$ je skutočné hodnotenie. Nevýhodou RMSE je, že aj malé množstvo zle predpovedaných hodnotení môže výrazne ovplyvniť celkové meranie [3]. *MAE,* na rozdiel od predošlej metriky, chybu neumocňuje, ale získa jej absolútnu hodnotu, čím zachováva vyváženosť medzi jednotlivými chybami. MAE sa vypočíta nasledovne [43]:

$$MAE = \frac{1}{|T|}\sum_{(u,i)\in T}|\hat{r}_{u,i} - r_{u,i}| \tag{11}$$

Jednou z variácií RMSE a MAE je ich normalizovaná hodnota vzhľadom na rozsah hodnotenia. Sú označované *NRMSE* a *NMAE* a vypočítajú sa takto [3]:

$$NRMSE = \frac{RMSE}{r_{max} - r_{min}} \tag{12}$$

$$NMAE = \frac{MAE}{r_{max} - r_{min}} \tag{13}$$

Normalizované chyby ležia vždy na intervale (0, 1), čo umožňuje ich lepšiu interpretáciu, napríklad pri porovnávaní s algoritmami pracujúcimi nad dátovými sadami s inými rozsahmi hodnotení [3]. *Priemerné RMSE* a *priemerné MAE* sú ďalšou metrikou, používanou v prípade nevyrovnanej distribúcie produktov, kde dochádza k ovplyvneniu

celkového merania najfrekventovanejšími produktami [43]. V takom prípade je vhodné vypočítať chybu pre každý produkt zvlášť a potom vypočítať priemernú chybu [43].

2.7.2 Meranie presnosti zoznamu odporúčania

V niektorých prípadoch odporúčacie systémy nepredpovedajú hodnotenie produktov, ale to, či si používateľ produkt vyberie, určujú iným spôsobom [43]. Výber produktu môžeme reprezentovať ako binárnu hodnotu vyjadrujúcu, či používateľ film hodnotil alebo nie. Medzi takéto metriky patrí presnosť (angl. precision), úplnosť (angl. recall), ROC krivka (angl. Receiver Operating Characteristics) [2, 43], AUC krivka (angl. Area Under the ROC Curve) [2, 12] a ich variácie.

Presnosť je vyjadrená ako pomer počtu pravdivo pozitívnych výsledkov k celkovému počtu pozitívnych výsledkov (pravdivo aj falošne pozitívnych), vo vzorci [16]:

$$Presnosť = \frac{počet\ správne\ odporučených\ \ produktov}{počet\ všetkých\ odporučených\ produktov} \tag{14}$$

Vyjadruje tak, akú časť z produktov, ktoré boli odporučené, si používateľ skutočne vybral. Pri *top-N* odporúčacích systémoch ide o počet z *N* produktov, ktoré si používateľ vybral. V publikáciách sa presnosť pri top-N odporúčacích systémoch označuje aj ako *precision@N* [43, 44]. *Úplnosť* predstavuje pomer počtu pravdivo pozitívnych k súčtu počtov pravdivo pozitívnych a falošne negatívnych výsledkov. Vyjadruje, akú časť z produktov, ktoré si používateľ vybral, boli používateľovi odporučené prostredníctvom odporúčacieho systému. Pri off-line evaluácii môžeme úplnosť vyjadriť vzorcom uvedeným nižšie [16].

$$Úplnosť = \frac{počet\ správne\ odporučených\ \ produktov}{počet\ správne\ odporučených\ + počet\ nesprávne\ neodporučených\ produktov} \tag{15}$$

Presnosť a ani úplnosť neberú do úvahy poradie produktov, v akom boli odporučené. Metrika NDCG – normalizovaný znížený kumulatívny zisk (angl. Normalized Discounted Cumulative Gain) sa používa v prípadoch, kde je potrebné vyhodnotiť poradie odporučených produktov. NDCG je možné vypočítať nasledovne [45]:

$$NDCG_p = \frac{\sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}}{IDCG_p} \tag{16}$$

V tomto vzorci p vyjadruje dĺžku zoznamu, rel_i je ohodnotená relevancia výskytu na konkrétnom poradí a $IDCG_p$ vyjadruje DCG ideálne zoradeného zoznamu.

2.7.3 Top-N odporúčanie a evaluácia chyby predikcie

V predošlých podkapitolách sme si predstavili základné techniky odporúčania. Zatiaľ čo niektoré metódy sú optimalizované na odporúčanie zoznamov produktov, veľa metód miesto toho predikuje hodnotenie produktu používateľom a negeneruje žiaden zoradený zoznam. Avšak v praxi sú používateľovi odporúčané produkty v podobe zoradeného

zoznamu určitej dĺžky, pričom používateľ venuje vyššiu pozornosť produktom umiestneným na prvých priečkach zoznamu [3]. Bežne používané metriky založené na výpočte chyby, ako je napríklad RMSE, tak nie sú vhodné na vyhodnotenie takéhoto zoznamu. Tieto metriky navyše nemusia korelovať s metrikami na vyhodnotenie presnosti a pri znížení chyby nemusí nutne nastať nárast presnosti top-N odporúčania [46]. Autori článku [46] vykonali experiment, v ktorom porovnali presnosť metódy kolaboratívneho filtrovania využívajúcej SVD s metódou nepersonalizovaného odporúčania založeného na odporúčaní populárnych produktov a ukázali, že bez ohľadu na RMSE, nepersonalizovaná metóda dosahovala porovnateľnú presnosť so sofistikovaným SVD algoritmom.

Okrem poukázania na rozdiel medzi evaluáciou chyby a presnosti, autori článku tiež navrhli spôsob evaluácie presnosti metód, ktorý je efektívny nad dátovými sadami obsahujúcimi vysoké množstvo produktov. Výpočet presnosti rozdelili do nasledujúcich krokov:

1. Z datasetu sa náhodne vyberie malá podmnožina hodnotení (autori z datasetu od Netflixu obsahujúcom 100 miliónov hodnotení vybrali 1.4 % hodnotení).
2. Z tejto množiny vybrali pozitívne hodnotenia (5 hviezdičkové).
3. Pre každý produkt, ktorý používateľ hodnotil pozitívne, vybrali náhodne 1000 ďalších neohodnotených produktov z pôvodnej množiny.
4. Z týchto produktov zhotovili zoradený zoznam produktov, pričom zoznam obsahoval N produktov, ktorých predikované hodnotenie bolo najvyššie
5. Ak sa pozitívne hodnotený produkt nachádza v tomto zozname, autori to považujú za zásah.

Výslednú presnosť je potom možné vypočítať nasledovne:

$$Presnosť\ (N) = \frac{počet\ zásahov}{N * |T|} \tag{17}$$

Pričom N vyjadruje dĺžku zoznamu odporučených produktov a $|T|$ veľkosť testovacej sady.

3 Cieľ práce

V predchádzajúcich kapitolách sme uviedli niekoľko metód odporúčania a predstavili ich využitie v reálnych doménach. V týchto doménach sa využívajú také techniky odporúčania, ktoré sú považované za najvhodnejšie. Na odporúčacie systémy sú v každej doméne kladené rôzne nároky a požiadavky, ktoré vyplývajú z podstaty problematiky, ktorú má odporúčací systém riešiť. Či už ide o odporúčanie v doméne, v ktorej je potrebná rýchla adaptácia na pribúdajúce produkty, respektíve používateľov, alebo je potrebná vysoká presnosť aj na úkor časovej náročnosti, prípadne sa vyžaduje explicitná väzba od používateľa v podobe formuláru, bez ktorej by nebolo možné odporúčať. Vzhľadom na všetky tieto faktory sú niektoré odporúčacie metódy v konkrétnej doméne vhodnejšie, než iné.

Väčšina súčasných riešení zvolenú metódu odporúčania aplikuje rovnako na všetkých používateľov a každému používateľovi sú generované odporúčania na základe jednej metódy (alebo kombinácii metód v prípade hybridného odporúčania). Používatelia sa ale aj v rámci jednej domény od seba líšia, a to nielen vkusom, ale aj svojim správaním. Navyše, ich vkus a správanie sa časom môže meniť a to sa môže odraziť aj na presnosti odporúčacej metódy.

Cieľom práce je navrhnúť odporúčací systém v doméne odporúčania filmov, ktorý bude využívať algoritmus na adaptívny výber najpresnejšej odporúčacej metódy pre konkrétneho používateľa na základe jeho charakteristík a kontextu. Adaptívnym výberom metódy, ktorou budú používateľovi odporúčané ďalšie produkty, chceme docieliť väčšiu personalizovanosť a zvýšenie presnosti odporúčania oproti konvenčným riešeniam založeným na jednom algoritme. Tento cieľ vieme rozložiť na niekoľko fáz:

- Zvolenie metód odporúčania, ktoré budú v hybridnom systéme použité,
- Vytvorenie kontextovo-preferenčného modelu používateľa, ktorý bude použitý pri výbere najvhodnejšej odporúčacej metódy,
- Výber a prípadne výpočet atribútov, ktoré sa použijú v kontextovo-preferenčnom modeli používateľa,
- Navrhnutie a implementácia algoritmu na adaptívny výber odporúčacej metódy,
- Overenie navrhnutého algoritmu adaptívneho výberu odporúčacej metódy a porovnanie s presnosťami jednotlivých odporúčacích metód.

V tejto práci sa zaoberáme niekoľkými výskumnými otázkami:

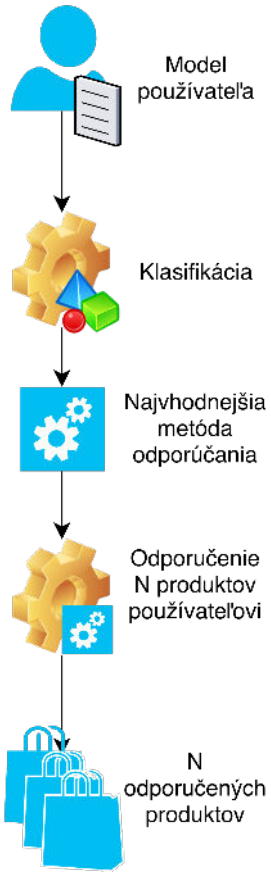
- V1: Existuje pre každú odporúčaciu metódu skupina používateľov, u ktorých metóda odporúčania dosahuje najvyššiu presnosť v porovnaní s ostatnými?
- V2: Na základe akých charakteristík je možné rozlíšiť jednotlivé skupiny používateľov?

- V3: Pri akej kombinácii odporúčacích metód poskytuje metóda adaptívneho výberu najvyššiu presnosť?
- V4: Aké zlepšenie presnosti odporúčania dokáže priniesť adaptívny výber odporúčacej metódy z množiny dostupných metód pre používateľa oproti využitiu klasických prístupov?
- V5: Existuje vzťah medzi technikami odporúčania a charakteristikami skupín používateľov?

Vo výskumnej otázke *V4* pod využitím klasických prístupov rozumieme využitie odporúčacích metód, ktoré sú integrované v našej hybridnej metóde, samostatne. Venujeme sa hypotéze, že v kontexte jednej domény pre rôznych používateľov presnejšie odporúčajú rôzne metódy odporúčania. V hypotéze predpokladáme, že existujú skupiny používateľov s určitými charakteristikami, ktoré môžu byť použité na výber odporúčacej metódy, ktorej presnosť bude pre danú skupinu najvyššia v porovnaní s ostatnými metódami. Napríklad, zatiaľ čo používateľovi, ktorý má málo hodnotení a sleduje iba historické filmy, generuje najpresnejšie odporúčacia metóda založená na obsahu, inému používateľovi, ktorý sleduje rôzne žánre, lepšie generuje odporúčacia metóda kolaboratívneho filtrovania.

4 Návrh adaptívnej metódy odporúčania

V tejto kapitole predstavíme návrh metódy, ktorej podstata spočíva v rozhodnutí o výbere optimálnej metódy odporúčania pre používateľa vzhľadom na kontext, v ktorom sa nachádza. Optimálnosť metódy môže byť interpretovaná ako presnosť odporúčania, chyba predikcie hodnotenia, alebo iná metrika vhodná na vyhodnotenie odporúčacieho systému. Výber najpresnejšej metódy interpretujeme ako klasifikačný problém, v ktorom používateľa u potrebujeme zaradiť do triedy c, pričom trieda c reprezentuje odporúčaciu metódu. Klasifikátor, ktorý na riešenie tohto problému použijeme, obsahuje na vstupe kontextovo-preferenčný model používateľa, ktorému chceme odporučiť produkty. Naučený klasifikátor na základe čŕt používateľa vyhodnotí, ktorá metóda odporúčania (trieda) z množiny dostupných metód (tried) sa javí ako najvhodnejšia pre aktívneho používateľa. Táto metóda je následne použitá na vygenerovanie n produktov, ktoré sú odporučené používateľovi. Nasledujúci obrázok objasňuje základný princíp odporúčacieho systému s využitím adaptívneho výberu odporúčacej metódy.



Obrázok 4.1: Diagram zobrazujúci aktivity v navrhovanom odporúčacom systéme

V kapitole 4.1 sa venujeme podrobnejšiemu opisu kontextovo-preferenčného modelu používateľa a v kapitole 4.2 opisujeme princíp fungovania klasifikátora ako nástroja na určenie najvhodnejšej metódy odporúčania.

4.1 Kontextovo-preferenčný model používateľa

Kontextovo-preferenčný model používateľa je vektor obsahujúci črty, ktoré charakterizujú používateľa pre účely jeho klasifikácie do triedy odporúčania. V tejto práci sa zaoberáme odporúčaním v doméne multimediálneho obsahu, čo súvisí aj s výberom čŕt, ktoré budú v modeli obsiahnuté. Črty v modeli rozdeľujeme na tri typy:

a) Kontextové informácie
b) Preferenčné informácie
c) Demografické informácie

Autori [47] ukázali, že kontextové informácie zohrávajú v odporúčaní dôležitú rolu a je možné ich využiť v rôznych fázach odporúčania vrátane fázy pred odporúčaním (angl. pre-filtering). V našom prípade využijeme kontextové informácie, ktoré extrahujeme z dátovej sady, ako aj z aktuálneho kontextu používateľa, na zatriedenie používateľa do najvhodnejšej triedy odporúčania. Medzi kontextové informácie zaraďujeme informácie ako:

- celkový počet hodnotených produktov používateľom,
- histogram hodnotení produktov používateľom,
- variancia atribútov produktov, ktoré používateľ hodnotil,
- časová pečiatka (angl. timestamp) hodnotenia produktov,
- počet unikátnych kategórií produktov, ktoré používateľ hodnotil.

Medzi preferenčné informácie používateľa zaraďujeme vektory týkajúce sa informácií o produktoch, ktoré hodnotil. Tieto informácie vyjadrujú vkus používateľa. V doméne odporúčania filmov, ktorej sa venujeme, ide o nasledujúce atribúty:

- histogram žánrov filmov, ktoré používateľ hodnotil,
- histogram kľúčových slov filmov, ktoré používateľ hodnotil,
- dĺžky trvania filmov, ktoré používateľ hodnotil,
- ďalšie informácie o filmoch.

Pod demografickými informáciami rozumieme osobné informácie používateľa ako vek, pohlavie, zamestnanie a krajina pôvodu. Výber vhodných čŕt, ktoré budú zastúpené v kontextovo-preferenčnom modeli používateľa, závisí od toho, či ich klasifikátor využije na naučenie sa modelu, podľa ktorého bude používateľom vyberať metódu odporúčania a ich použitie prinesie kvalitatívne lepšie výsledky. Analýzou a výberom vhodných čŕt pre konkrétnu dátovú sadu, nad ktorou realizujeme túto prácu, sa budeme zaoberať v ďalších kapitolách.

4.2 Adaptívny výber odporúčacej metódy ako klasifikačný problém

Pod adaptívnym výberom odporúčacej metódy rozumieme taký spôsob výberu metódy, pri ktorej sú zohľadnené charakteristiky používateľa, pre ktorého výber uskutočňujeme. Používateľ je charakterizovaný kontextovo-preferenčným modelom, ktorý obsahuje rôzne črty, ktoré vyjadrujú jeho správanie, ale aj jeho preferencie v danej doméne a časový aspekt.

Výber najpresnejšej metódy uskutočníme prostredníctvom natrénovaného modelu klasifikátora. Pre túto problematiku sme zvolili klasifikačnú metódu Náhodný les (angl. random forest). Túto metódu sme vybrali pre jej efektívnosť pri práci s veľkými dátovými sadami, dobrou presnosťou spomedzi podobných metód a jednoduchou možnosťou overenia dôležitosti čŕt (angl. features importance), ktoré sa použili na natrénovanie modelu. Okrem klasifikátora Náhodný les porovnávame jeho výkonnosť s klasifikátorom XGB (angl. eXtreme Gradient Boosting)[2].

Na naučenie modelu klasifikátora najprv potrebujeme získať trénovacie dáta, ktoré pozostávajú z kontextovo-preferenčných modelov používateľov a informácie, do ktorej triedy každý z modelov patrí. V kontexte výberu najpresnejšej metódy odporúčania pre používateľa práve ony reprezentujú triedy klasifikácie. Na zaradenie modelov používateľov, ktoré budú použité na trénovanie modelu klasifikátora, do tried, potrebujeme získať optimálnu metódu pre každý z nich. Proces získania najlepšej metódy odporúčania objasňuje Obrázok 4.2.

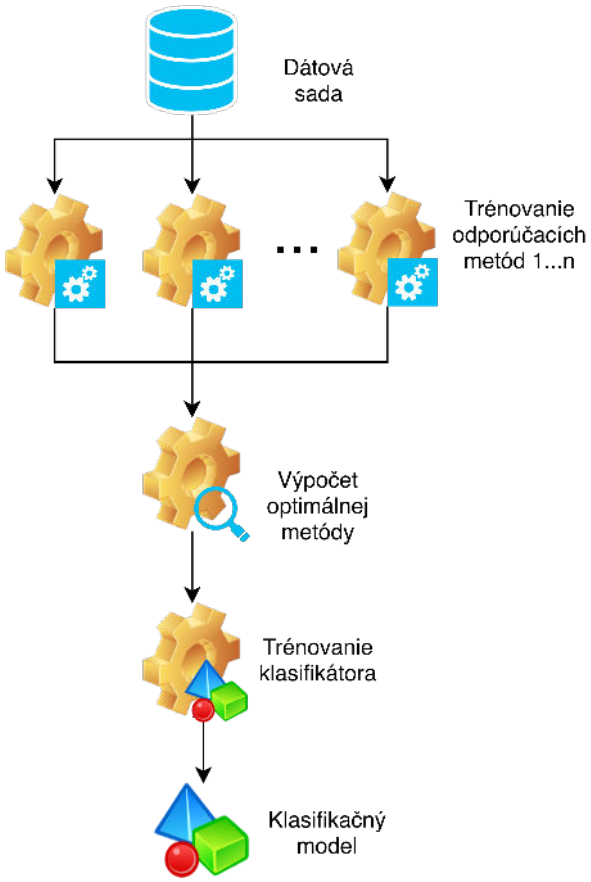
4.2.1 Trénovacie dáta pre odporúčacie metódy

Na začiatku dátovú sadu rozdelíme na trénovaciu a testovaciu časť. Trénovacie dáta slúžia na natrénovanie všetkých odporúčacích metód, ktoré sú vo vyvíjanom odporúčacom systéme aplikované. Štruktúra týchto dát závisí od techniky odporúčania, ktorú konkrétna metóda využíva. Pri metódach kolaboratívneho filtrovania obsahujú tieto dáta informácie o hodnoteniach, v prípade obsahových techník vstupujú do trénovania aj charakteristiky produktov.

4.2.2 Testovacie dáta pre odporúčacie metódy

Testovacie dáta, ktorých štruktúra opäť závisí od použitých techník odporúčania, sú použité v procese výpočtu najpresnejšej odporúčacej metódy pre jednotlivých používateľov. Po označení každej inštancie v tejto sade najpresnejšou metódou odporúčania sú z týchto inštancií vytvorené kontextovo-preferenčné modely používateľov, ktoré sú použité na natrénovania klasifikátora.

[2] xgboost.readthedocs.io



Obrázok 4.2: Proces prípravy dát pre naučenie modelu klasifikátora

4.2.3 Trénovanie odporúčacích metód

V tejto fáze sa vykoná trénovanie modelov odporúčacích metód. Nakoľko odporúčacie metódy nie sú navzájom nijako závislé, je možné tento proces paralelizovať a urýchliť tak celý proces predprípravy dát pre naučenie modelu klasifikátora.

4.2.4 Výpočet optimálnej metódy

Tento krok zahŕňa evaluáciu natrénovaných odporúčacích metód testovacími dátami. Na evaluáciu je možné použiť niektorú z metrík vyhodnocovania úspešnosti odporúčania. Vzhľadom na využitie rôznych techník odporúčania sme sa rozhodli pre metriku *NDCG*, ktorej výhoda oproti presnosti *P@k* spočíva v schopnosti vyhodnotiť poradie odporučených produktov. Keďže potrebujeme získať najpresnejšiu metódu pre každého používateľa z testovacej sady, vykonáme výpočet *NDCG* nad agregovanými dátami podľa ID používateľa.

4.2.5 Trénovanie klasifikátora a klasifikačný model

Vo fáze trénovania klasifikátora máme na vstupe modely používateľov a ku každému modelu priradenú triedu. Ako klasifikátor je použitý Náhodný les. Naučený model klasifikátora je použitý na klasifikovanie používateľov do tried odporúčania. Na základe výsledku klasifikácie sú používateľovi generované odporúčania predikovanou metódou.

4.2.6 Testovacia fáza

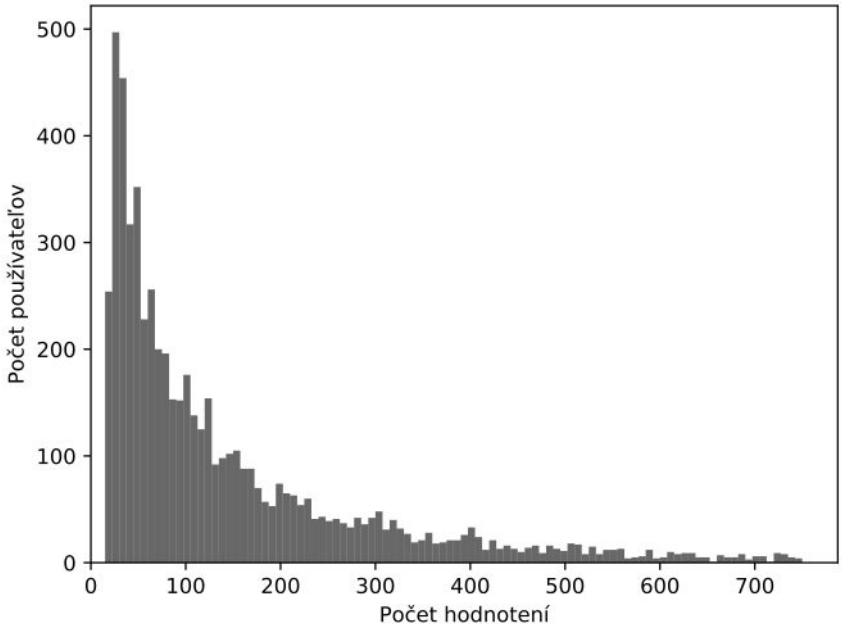
Vo fáze testovania (resp. prevádzky) je na základe aktuálnych informácií o aktívnom používateľovi zostrojený kontextovo-preferenčný model. Ten sa použije v kroku klasifikácie v zmysle diagramu na Obrázok 4.1 na predikciu najpresnejšej metódy odporúčania. Predikovaná metóda odporúčania je následne natrénovaná na modeloch používateľov a použitá na odporučenie produktov aktívnemu používateľovi.

5 Realizácia

V tejto kapitole opíšeme realizáciu odporúčacieho systému využívajúceho klasifikátor na adaptívny výber odporúčacej metódy. Navrhnutú metódu realizujeme v doméne multimediálneho obsahu, konkrétne v doméne odporúčania filmov.

5.1 Dátová sada

V tejto práci pracujeme s dátovou sadou MovieLens obsahujúcou 1 milión hodnotení (ML-1m)[3]. Táto sada obsahuje informácie o hodnoteniach 3706 filmov 6040 používateľmi. Každý používateľ v tejto sade hodnotil aspoň 20 filmov. Sada okrem hodnotení obsahuje aj informácie o filmoch, ako *názov*, *rok vydania* a *žánre*. Tiež obsahuje informácie o používateľoch ako *vek*, *pohlavie* a zamestnanie. Obrázok 5.1 zobrazuje histogram hodnotení. Dá sa z neho vyčítať, že väčšina používateľov má pod 100 hodnotení, pričom najviac je takých používateľov, ktorí majú 21 hodnotení. Hodnotenia sú na škále od 1 do 5 s nárastom jeden bod. Sada tiež obsahuje časovú pečiatku každého hodnotenia. Distribúcia hodnotení (počet hviezdičiek) je zobrazená na Obrázok 5.2. V dátovej sade existuje najviac hodnotení vo výške 4, zatiaľ čo najmenej hodnotení vo výške 1.



Obrázok 5.1: Histogram hodnotení filmov

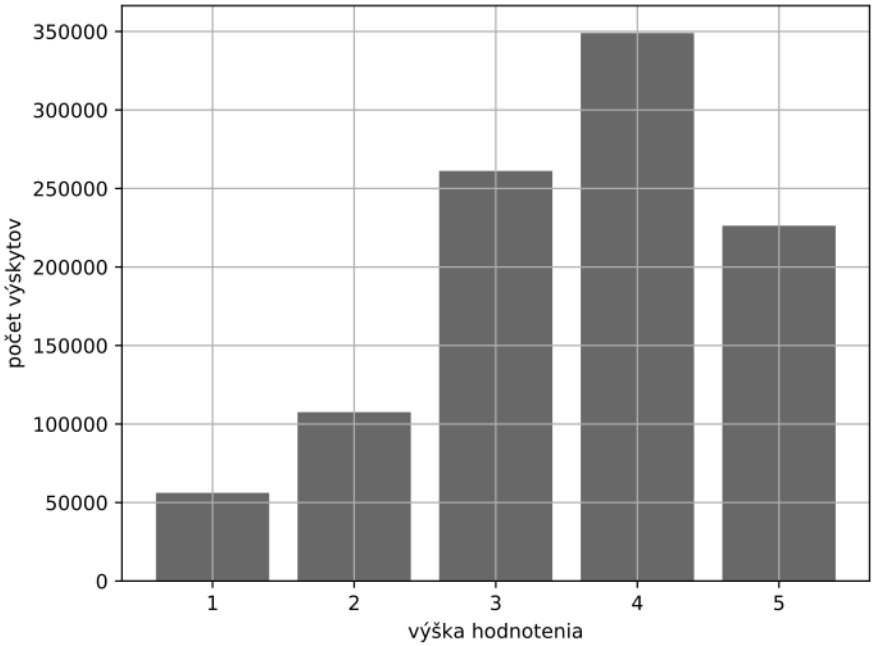
Okrem ML-1m používame aj dáta získané zo stránky The Movie DB[4] prostredníctvom ich API rozhrania, ktoré obsahujú dodatočné informácie o filmoch. Do JSON štruktúry sme pre každý film z dátovej sady uložili nasledujúce informácie:

- názov
- žánre

[3] grouplens.org/datasets/movielens/1m

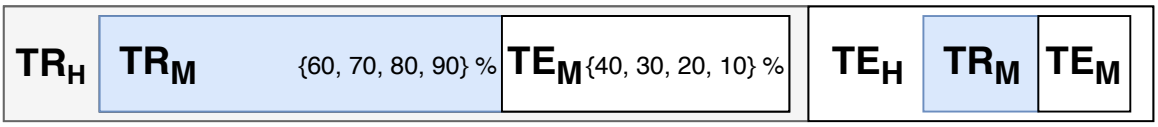
[4] themoviedb.org

- popis
- dátum vydania
- kľúčové slová
- rozpočet a zisk
- jazyk
- herci
- hodnotenie



Obrázok 5.2: Distribúcia hodnotení na základe výšky hodnotenia

V tejto sade existuje celkom 7881 kľúčových slov, 19 žánrov a 41546 hercov. Pretože implementovanú metódu budeme evaluovať off-line, rozdeľujeme dátovú sadu na viac častí. Dátovú sadu delíme na trénovaciu sadu pre hybridný systém TR_H a testovaciu sadu pre hybridný systém TE_H podľa Obrázok 5.3.



Obrázok 5.3: Ukážka delenia dátovej sady vo fáze vyhodnocovania. TR_H a TE_H sú trénovacie, resp. testovacie dáta pre hybridný systém; TR_M a TE_M sú trénovacie, resp. testovacie dáta pre odporúčacie metódy.

5.1.1 Trénovacia sada pre hybridný systém TR_H

Táto sada o veľkosti 70 % pôvodnej sady slúži na vygenerovanie relácií a následne vytvorenie kontextovo-preferenčných modelov používateľov, ktoré sú použité na trénovanie klasifikátora. Sada je ďalej dynamicky delená na trénovaciu časť pre odporúčacie metódy TR_M a testovaciu časť TE_M, nakoľko postupným zväčšovaním trénovacej sady simulujeme aktivitu používateľov. V našich experimentoch TR_M nadobúda veľkosti z množiny {60 %, 70 %, 80 %, 90 %}. TE_M slúži na evaluáciu odporúčacích metód a vytvorenie kontextovo-preferenčných modelov používateľov, ktoré sú vstupom pre naučenie modelu klasifikátora.

5.1.2 Testovacia sada pre hybridný systém TE_H

Táto sada o veľkosti 30 % pôvodnej sady slúži na kompletné otestovanie a vyhodnotenie implementovaného hybridného odporúčacieho systému využívajúceho adaptívny výber metódy. Vo fáze vyhodnocovania sa podobne ako vo fáze trénovania dynamicky delí za účelom simulovania aktivity.

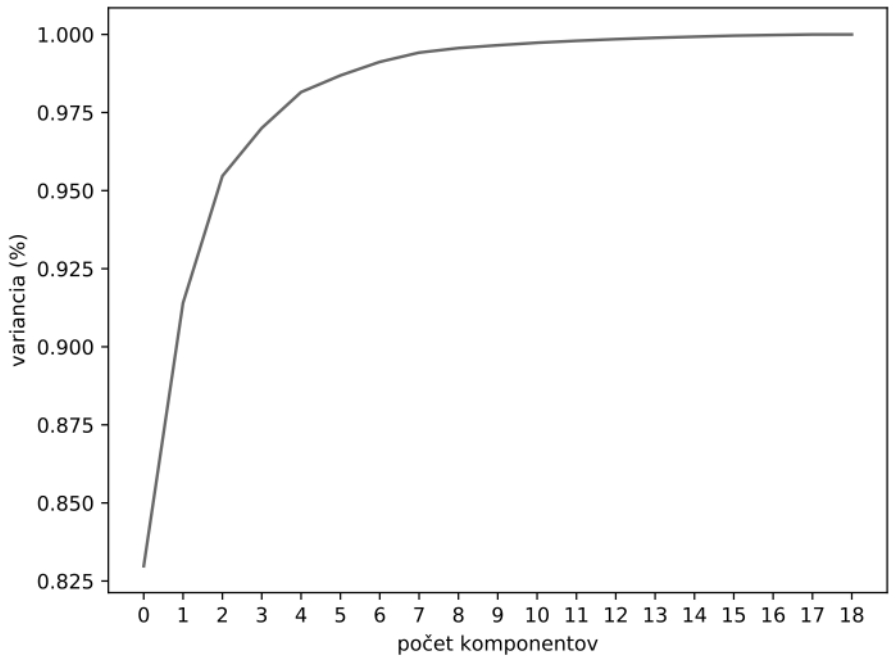
5.2 Kontextovo-preferenčný model používateľa

Kontextovo-preferenčný model používateľa obsahuje:

a) celkový počet hodnotení používateľa,
b) histogram hodnotení,
c) histogram žánrov,
d) počet najfrekventovanejších žánrov,
e) variancia žánrov,
f) vek používateľa
g) pohlavie používateľa,
h) zamestnanie používateľa,
i) aktuálny deň v týždni a mesiac,
j) histogram kľúčových slov.

Bod *d* vyjadruje počet žánrov, ktorých zastúpenie v histograme predstavuje aspoň 30 % oproti zastúpeniu najpočetnejšieho žánra. Bod *e* predstavuje varianciu σ^2 počtov žánrov.

Črty reprezentujúce histogram žánrov a kľúčových slov sú viacrozmerným atribútom, preto sme na zníženie dimenzionality použili analýzu hlavných komponentov PCA, ktorej predchádzala normalizácia. Histogramu žánrov sme dimenzionalitu zredukovali na 10 komponentov. Vychádzali sme pri tom z analýzy zachovania najvyššej variancie podľa grafu na Obrázok 5.4. Histogram kľúčových slov sme prostredníctvom PCA zredukovali do 15 komponentov.



Obrázok 5.4: Závislosť zachovania variancie a počtu komponentov PCA pre histogram žánrov

Aktuálny deň v týždni a mesiac sme získali prostredníctvom dynamického delenia trénovacej sady, ako je opísané v kapitole 5.1.1. Aktuálny čas predstavuje čas hodnotenia prvého produktu v testovacej sade aktívnym používateľom. Tieto atribúty sú cyklické a preto ich reprezentácia na intervale <0, 6>, resp. <1, 12> nie je dobrá pre zachovanie informácie o cyklicite. Na jej zachovanie sme tieto informácie transformovali na sínusovú a kosínusovú hodnotu podľa článku [48], ktorý sa zaoberá kódovaním časových údajov. Každú cyklickú črtu f sme transformovali do dvoch nových čŕt f_{sin} a f_{cos} nasledovne [48]:

$$f_{sin} = \sin\left(f * \frac{2\Pi}{T}\right) \tag{18}$$

$$f_{cos} = \cos\left(f * \frac{2\Pi}{T}\right) \tag{19}$$

kde T je konštanta, ktorá v prípade dňa v týždni nadobúda hodnotu 7 a v prípade mesiaca hodnotu 12.

Pohlavie používateľa sme transformovali na binárnu hodnotu 1 predstavujúcu muža a 0 predstavujúcu ženu. Ostatné kategorické črty sme transformovali na binárne indikátory, aby ich bolo možné použiť v procese klasifikácie.

5.3 Metóda adaptívneho výberu odporúčacej metódy

Ako metódy odporúčania sme použili metódy kolaboratívneho filtrovania implementované v Python knižnici Surprise[7], Hybridnú metódu LightFM implementovanú v rovnomennej Python knižnici[8] a odporúčanie založené na obsahu, ktorej implementácia je opísaná v kapitole 5.3.1. Využívame tieto metódy odporúčania:

- Baseline Only,
- Co-Clustering,
- Slope One,
- SVD,
- KNN Basic,
- LightFM,
- CB.

Tabuľka 5.1 zobrazuje parametre metód odporúčania použité pri trénovaní. Metódy odporúčania knižnice Surprise neposkytujú implementáciu top-N odporúčania, preto sme túto metódu implementovali. Táto metóda pre každého aktívneho používateľa u získa prostredníctvom kosínusovej podobnosti zoznam produktov I od 50 najpodobnejších používateľov. Následne použije metódu z knižnice Surprise na predikciu hodnotenia $r_{u,i}$

[7] surpriselib.com

[8] lyst.github.io/lightfm/docs/home.html

pre každý produkt $i \in I$. Zoznam produktov s prideleným predikovaných hodnotením zoradí zostupne podľa hodnotenia a aktívnemu používateľovi odporučí N najlepšie hodnotených produktov. Na výpočet kosínusovej podobnosti sú použité binárne vektory s dĺžkou rovnou celkovému počtu filmov v dátovej sade, kde hodnoty indikujú, či používateľ daný produkt hodnotil.

Tabuľka 5.1: Parametre metód odporúčania

Algoritmus	**Názov parametra**	**Hodnota parametra**
Baseline Only	metóda odhadu základných línií	SGD (angl. Stochastic Gradient Descent)
	založené na používateľovi	Áno
Co-Clustering	# zhlukov (používatelia)	7
	# zhlukov (produkty)	5
	# epoch	30
SVD	# faktorov	20
	# epoch	30
KNN Basic	# susedstiev (max)	50
	metóda podobnosti	kosínusová
	založené na používateľovi	Áno
LightFM	# komponentov	30
	stratová funkcia	WARP (angl. Weighted Approximate-Rank Pairwise)

Výber kombinácie spomedzi týchto metód do našej realizácie je predmetom kapitoly Overenie. Pri procese výberu najvhodnejšej metódy odporúčania používame klasifikačnú metódu Náhodný les. Tento klasifikátor sme nakonfigurovali hodnotami parametrov zobrazenými v tabuľke Tabuľka 5.2. Parametre predstavujú argumenty konštruktora triedy *RandomForestClassifier* knižnice scikit-learn. V niektorých prípadoch dosahoval vyššiu presnosť klasifikátor XGB, ktorého parametre sú spísané v Tabuľka 5.3. Na nájdenie kombinácie parametrov, pri ktorej dosahuje klasifikátor najvyššiu presnosť, sme použili techniku ladenia parametrov pomocou náhodného hľadania *RandomizedSearchCV* knižnice scikit-learn.

Tabuľka 5.2: Parametre klasifikátora Náhodný les

Názov parametra	**Hodnota parametra**
bootstrap	True
criterion	gini
max_depth	None
min_samples_split	3
min_samples_leaf	2

Názov parametra	Hodnota parametra
max_features	$\sqrt{počet\ čŕt}$
n_estimators	500

Tabuľka 5.3: Parametre klasifikátora XGB

Názov parametra	Hodnota parametra
max_depth	150
n_estimators	500
learning_rate	0.005
reg_alpha	0.05

5.3.1 Metóda odporúčania založená na obsahu

Súčasťou navrhovaného hybridného odporúčacieho systému je aj metóda odporúčania založená na obsahu. Táto metóda predstavuje jednu z množiny metód, ktorá je použitá na odporúčanie produktov používateľom. Na nájdenie podobností medzi produktami využíva kosínusovú podobnosť, ktorú počíta váhovaným súčtom matíc podobností. Každá matica podobností predstavuje kosínusovú podobnosť produktov na základe niektorej charakteristiky produktu. V implementácii používame črty *názov filmu*, *kľúčové slová*, *opis filmu*, *žáner* a *mená hercov*. Podrobný návrh, realizácia a vyhodnotenie tejto metódy je uvedené v prílohe B.

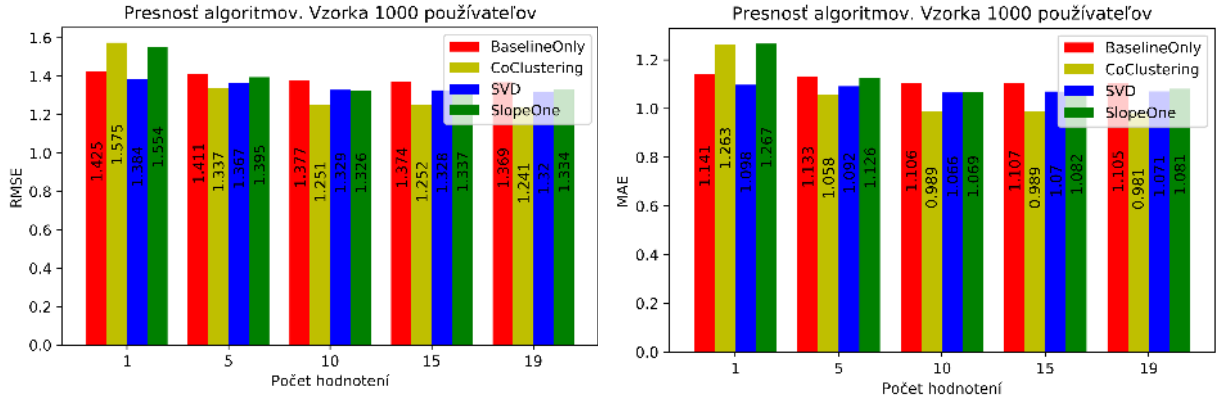
6 Overenie

V tejto kapitole sa venujeme overeniu implementovanej hybridnej metódy ako aj zodpovedaniu výskumných otázok z kapitoly 3. Overenie sme rozdelili na niekoľko častí. V každej časti overujeme naše riešenie na inej kombinácií odporúčacích metód, nakoľko existuje závislosť medzi výkonnosťou realizovaného odporúčacieho systému a metódami, ktoré predikuje. Tiež sa venujeme overeniu dôležitosti čŕt v kontextovo-preferenčnom modeli a analýze skupín používateľov, ktorým jednotlivé metódy odporúčajú najpresnejšie. Kapitola 6.7 obsahuje sumarizáciu a komentár k jednotlivým výskumným otázkam.

6.1 Evaluácia metód odporúčania

V tejto časti sa venujeme analýze odporúčacích metód a výkonu, aký dosahujú na dátovej sade MovieLens 1M. Zároveň sa venujeme výskumným otázkam *V1* a *V2*. V tejto podkapitole demonštrujeme, že pre každú z integrovaných odporúčacích metód existuje skupina používateľov, nad ktorými dosahuje daná metóda najvyššiu presnosť v porovnaní s ostatnými metódami.

Na zodpovedanie výskumnej otázky *V1* sme zisťovali, ako presne dokážu vybrané algoritmy predpovedať hodnotenie produktov s obmedzeným množstvom evidovaných hodnotení produktov používateľmi. Presnosť sme merali postupne nad používateľmi s jedným, piatimi, desiatimi, a devätnástimi hodnoteniami v trénovacej množine dát. Zo všetkých algoritmov knižnice Surprise sme vybrali tie, ktoré poskytovali podľa autorov knižnice najpresnejšie výsledky. Výsledky sú znázornené v nasledujúcich grafoch v metrikách RMSE a MAE.



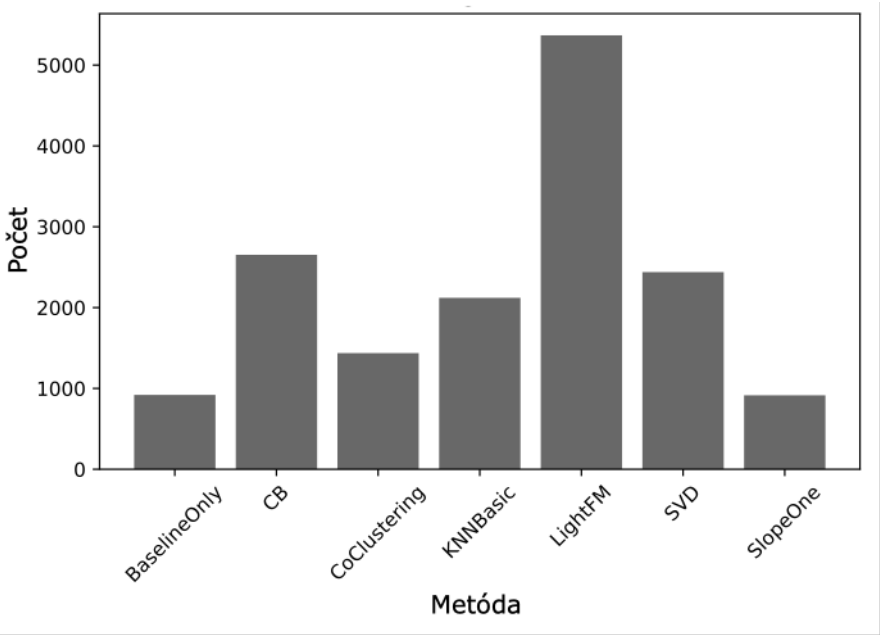
Obrázok 6.1: Presnosť algoritmov knižnice Surprise

Z vyššie uvedených grafov vyplýva, že algoritmus SVD dokáže v porovnaní s ostatnými algoritmami najlepšie predpovedať hodnotenie filmu, ak evidujeme iba jedno hodnotenie používateľa. S rastúcim počtom evidovaných hodnotení rastie presnosť algoritmu

spoločného zhlukovania Co-Clustering, ktorý poskytol najpresnejšie predpovede hodnotení pri evidencii viac ako jedného hodnotenia používateľa.

Predošlá analýza ukázala, že s rastom počtu hodnotení sa menia aj presnosti odporúčacích algoritmov. Ďalej sme preto chceli zistiť najpresnejšiu metódu spomedzi tých, ktoré v našej realizácii používame, pre každého používateľa v dátovej sade. To nám umožní analyzovať skupiny používateľov, u ktorých podáva konkrétny algoritmus lepšie výsledky v porovnaní s ostatnými. Na zistenie najpresnejšieho algoritmu vzhľadom na metriku NDCG sme nad dátovou sadou vykonali výpočet, ktorý nad 75 % dátovej sady natrénoval všetky metódy odporúčania a následne na zvyšných 25 % dátovej sady vypočítal odporúčania a NDCG nad množinou hodnotení každého používateľa. Výsledkom bol zoznam hodnôt NDCG pre každého používateľa. Vzhľadom na to, že dátová sada ML-1m obsahuje iba používateľov s minimálne dvadsiatimi hodnoteniami, dátovú sadu sme pred trénovaním normalizovali, aby sme pokryli aj situácie, kedy má používateľ minimálny počet hodnotení. Preto sme každému používateľovi odobrali konštantne 18 hodnotení, aby v dátovej sade existovali aj používatelia s jedným hodnotením v trénovacej a jedným v testovacej sade.

Zo získaných predikcií, ktoré sme agregovali podľa identifikátora používateľa, sme vypočítali NDCG a pre každého používateľa určili algoritmus, ktorý mu odporučil najpresnejšie. Histogram na Obrázok 6.2 zobrazuje vzťah medzi algoritmami a počtom používateľov, u ktorých dosiahli najvyššiu presnosť.



Obrázok 6.2: Početnosti používateľov, ktorým jednotlivé algoritmy odporúčali najpresnejšie

Ako môžeme pozorovať, aj napriek prevahe algoritmu LightFM majú aj ostatné algoritmy výrazné zastúpenie a sú schopné odporúčať lepšie pre 65.7 % používateľov oproti LightFM. Tento graf zároveň indikuje, že existujú skupiny používateľov, pre ktoré konkrétna metóda odporúča presnejšie, ako pre iné skupiny.

Tiež sme pre algoritmy určili ich odchýlku presnosti *P@10 (NDCG)* od najpresnejšieho algoritmu, ktorá dosiahla v priemere hodnotu *0.158 (0.194)* a vypočítali odchýlku druhého najpresnejšieho algoritmu od najpresnejšieho, ktorá dosiahla v priemere hodnotu *0.072 (0.110)*. Intuitívne môžeme túto odchýlku interpretovať ako priemernú stratu v presnosti odporúčania, ak by sme neodporúčali najpresnejšou dostupnou metódou. Vzhľadom na použitú metriku *P@10* tak ide približne o jeden až dva nesprávne odporučené produkty naviac.

Tabuľka 6.1 zobrazuje výkonnosť odporúčacích metód na dátovej sade Movielens-1m v Top-N odporúčaní. Presnosti boli vypočítané 5 zložkovou krížovou validáciou.

Tabuľka 6.1: Výkonnosť Top-N odporúčania

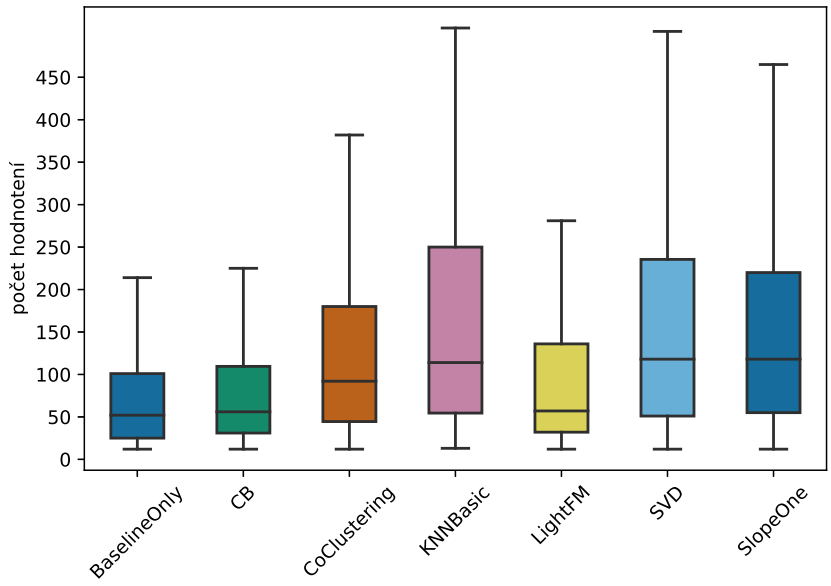
Metrika/Metóda	Presnosť			Úplnosť			NDCG
	P@3	P@5	P@10	R@3	R@5	R@10	
Baseline Only	0.1345	0.1349	0.1335	0.0099	0.0173	0.0354	0.3165
CB	0.2079	0.1994	0.1971	0.0202	0.0310	0.0488	0.4360
Co-Clustering	0.1520	0.1518	0.1505	0.0127	0.0215	0.0433	0.3363
KNN Basic	0.1452	0.1449	0.1447	0.0117	0.0199	0.0397	0.3210
LightFM	0.4400	0.4141	0.3705	0.0506	0.0772	0.1321	0.6719
Slope One	0.1453	0.1413	0.1354	0.0107	0.0176	0.0342	0.3104
SVD	0.1605	0.1546	0.1455	0.0125	0.0204	0.0392	0.3474

6.2 Evaluácia čŕt v kontextovo-preferenčnom modeli

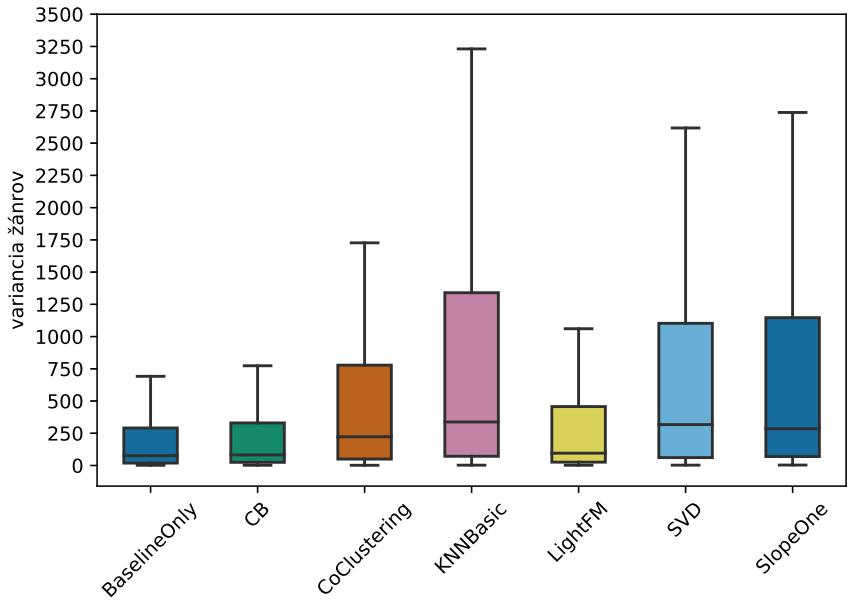
V tejto časti sa venujeme výskumnej otázke *V2*. Ukážeme, že existujú skupiny používateľov pre každú metódu odporúčania, v ktorej tieto metódy dosahujú najvyššiu presnosť v porovnaní od ostatných a je možné ich rozlíšiť na základe čŕt získaných z preferencií používateľa a kontextu.

Deskriptívnou štatistikou niektorých čŕt z kontextovo-preferenčného modelu používateľa sme objavili niekoľko indikátorov nasvedčujúcich, že medzi týmito skupinami existujú rozdiely vyjadriteľné modelom používateľa. Napríklad na krabicovom diagrame na obrázku Obrázok 6.3 je zobrazená závislosť medzi počtami hodnotení a najlepším algoritmom odporúčania. Z grafu vyplýva, že odporúčacie metódy *Baseline Only*, *CB* a *LightFM* dosahujú najvyššiu presnosť vtedy, keď má používateľ priemerne 50 hodnotení. Naopak metódy ako *KNN* a *SVD* sú najpresnejšie, keď má používateľ priemerne 100 hodnotení. Tejto jav pripisujeme podstate týchto metód, nakoľko metódy kolaboratívneho filtrovania bývajú počas studeného štartu menej presné oproti obsahovým, prípadne hybridným metódam.

Podobne ako počet hodnotení sme analyzovali aj varianciu žánrov a vek používateľa. Krabicové diagramy pre tieto črty sú zobrazené na Obrázok 6.4 a Obrázok 6.5.

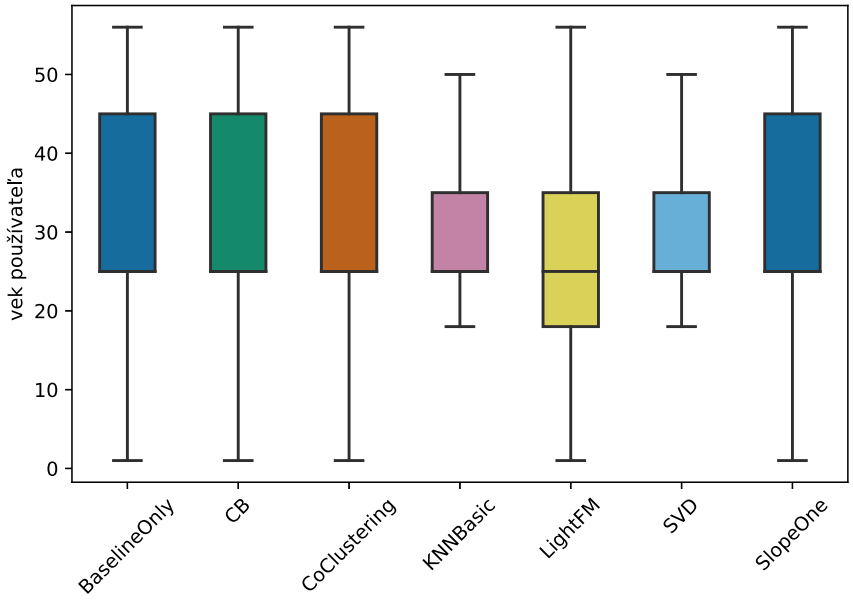


Obrázok 6.3: Analýza črty „počet hodnotení“



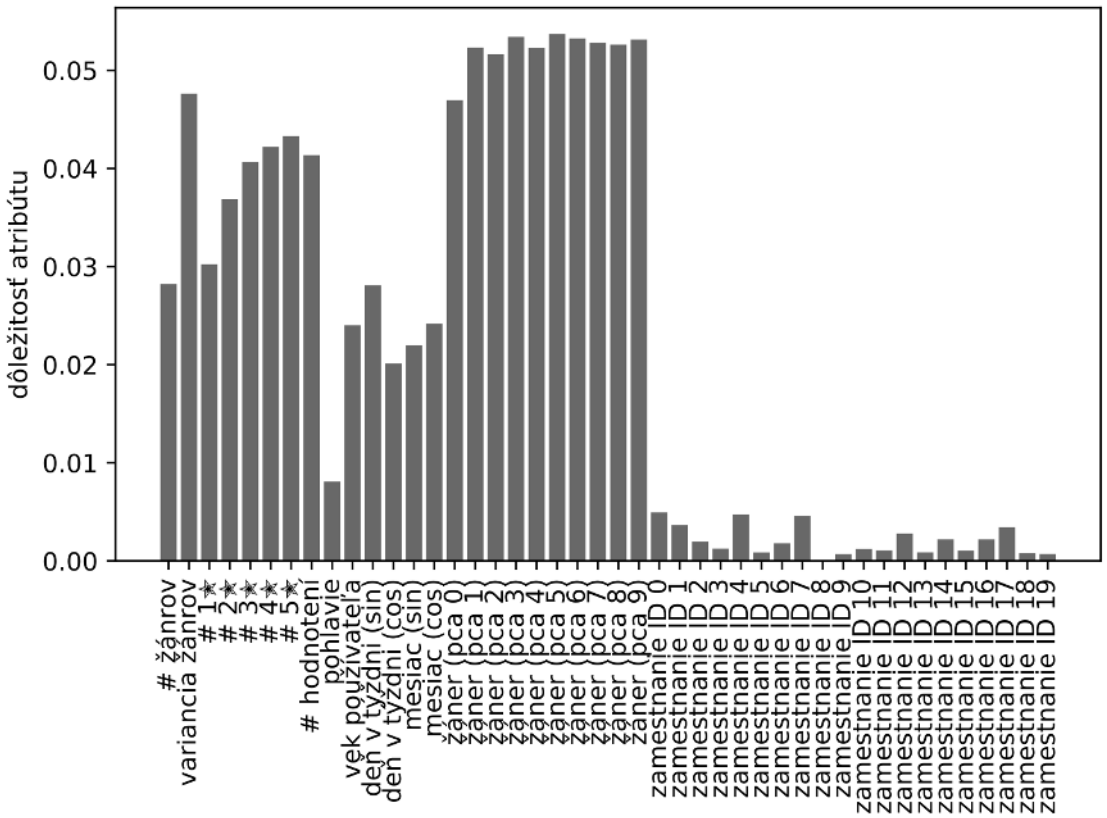
Obrázok 6.4: Analýza črty "variancia žánrov"

Z analýzy variancie žánrov vyplýva, že skupina, pre ktorú najlepšie odporúčajú metódy *KNN*, *SVD* a *Slope One* majú najvyššiu varianciu žánrov. Rozdiel medzi týmito metódami a metódou založenou na obsahu ukazuje, že *CB* najpresnejšie odporúča vtedy, keď je používateľ orientovaný na špecifickú skupinu žánrov, zatiaľ čo *CF* metódy presne odporúčajú aj vtedy, keď používateľ sleduje široké spektrum žánrov. Vek používateľa tiež zohráva rolu pri rozlíšení jednotlivých skupín používateľov.



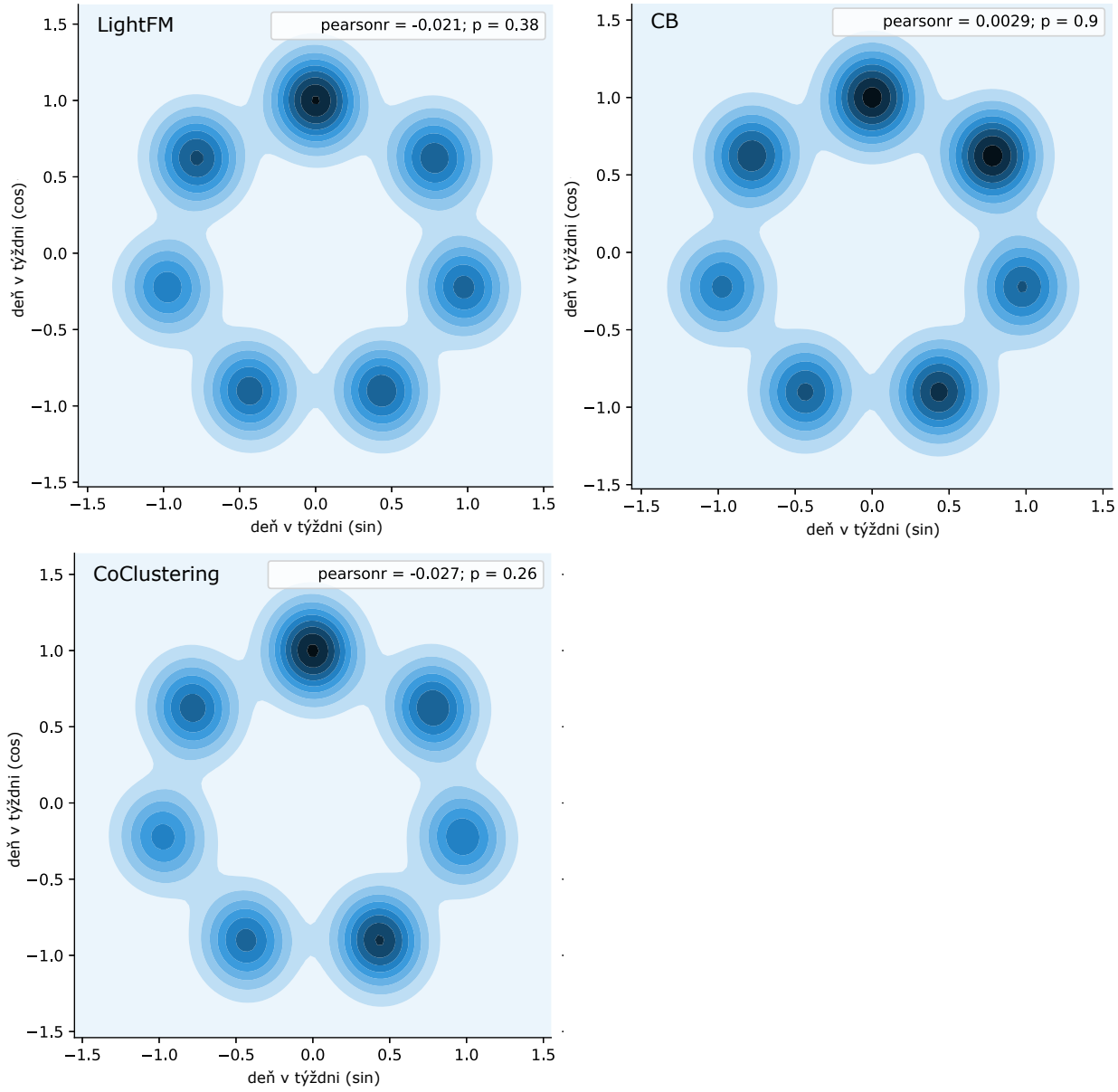
Obrázok 6.5: Analýza črty „vek používateľa“

Črty ako pohlavie používateľa a jeho zamestnanie neprejavovali na krabicových diagramoch žiadne výrazné rozdiely naprieč metódami a preto usudzujeme, že pre účel klasifikácie nie sú dôležité. Ako dôležitosť čŕt na obrázku Obrázok 6.6 ukazuje, tieto atribúty sú zanedbateľné pri trénovaní modelu klasifikátora Náhodný les.



Obrázok 6.6: Dôležitosť atribútov pri učení modelu klasifikátora

Cyklické údaje ako deň v týždni a mesiac sú ďalšími črtami, ktoré využívame v kontextovo-preferenčnom modeli používateľa. Každá z týchto čŕt je reprezentovaná dvoma atribútmi (sínusová a kosínusová hodnota) na základe definície z kapitoly 5.2 a preto sme na vizualizáciu vzťahu cyklických čŕt a metód odporúčania zvolili inú vizualizáciu. Grafy na Obrázok 6.7 zobrazujú rozdiel medzi metódou obsahového odporúčania, hybridnej metódy LightFM a metódy CoClustering vzhľadom na frekvenciu dňa v týždni. Na ukázanie rozdielu sme použili tieto tri, nakoľko každá z týchto metód patrí do inej techniky odporúčania.



Obrázok 6.7: Analýza črty „deň v týždni“

Ako môžeme na týchto grafoch vidieť, odporúčanie založené na obsahu (CB) má v porovnaní so zvyšnými dvoma metódami najfrekventovanejšiu oblasť v prvom (pravom hornom) kvadrante. Vyššia frekvencia je na grafoch zvýraznená tmavšou farbou. Tiež pozorujeme rozdiely vo štvrtom (pravom dolnom) kvadrante, v ktorom je CB

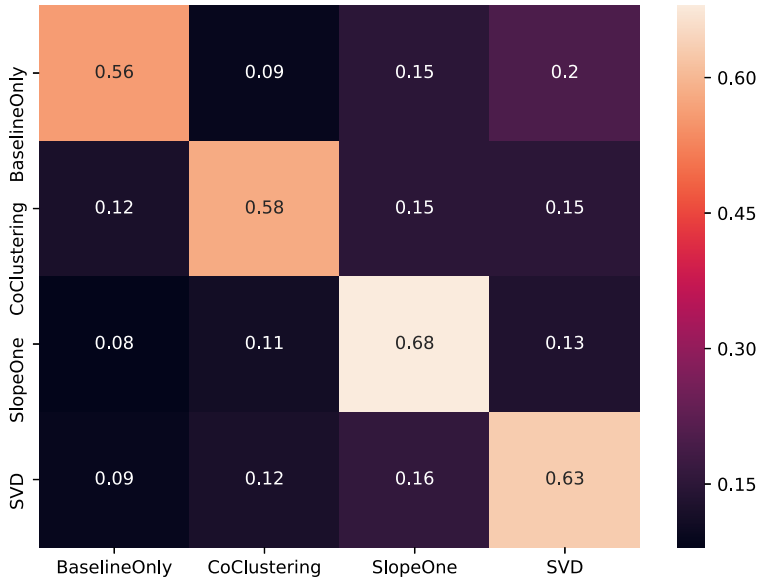
najfrekventovanejší, po ňom nasleduje metóda Co-Clustering a najmenej frekventovaná v tejto oblasti je hybridná metóda LightFM. Črta vyjadrujúca aktuálny mesiac mala podobné špecifiká naprieč metódami.

Niektoré ďalšie črty ako histogram hodnotení, žánrov a kľúčových slov, vzhľadom na ich viacrozmernosť, nedokážeme rozumne vizualizovať. Pri overení dôležitostí týchto čŕt natrénovaním modelu klasifikátora sme zistili, že použitie histogramu žánrov zvýšilo presnosť klasifikácie a histogram kľúčových slov nemal žiaden vplyv na výslednú presnosť klasifikácie.

6.3 Evaluácia hybridného odporúčania – integrované iba CF metódy

V tejto podkapitole sa venujeme výskumnej otázke *V3*. Vyhodnocujeme presnosť klasifikácie a hybridného odporúčania na konfigurácii, v ktorej vystupujú iba metódy kolaboratívneho filtrovania. Vybrali sme metódy BaselineOnly, Co-Clustering, Slope One a SVD, pretože podľa predošlej analýzy vhodnosti čŕt existujú medzi nimi rozdiely, ktoré by sa mohli v klasifikácii prejaviť.

Nasledujúci experiment ukazuje, že je možné zostrojiť klasifikátor, ktorý na základe modelu používateľa dokáže určiť optimálnu metódu odporúčania z vybranej podmnožiny. Najprv sme klasifikátor naučili na modeloch, ktoré obsahovali výlučne histogram hodnotení a celkový počet hodnotení používateľa. Ako klasifikačný algoritmus sme použili náhodný les s parametrami z kapitoly 5.3. Klasifikátor sme potom testovali na nezávislej vzorke používateľov, ktorí neboli prítomní v trénovacej sade klasifikátora a ani v trénovacej sade odporúčacích metód. Matica zámen (angl. confusion matrix) na nasledujúcom obrázku zobrazuje percentuálnu presnosť (v desatinnom zápise) predikcie jednotlivých metód odporúčania.

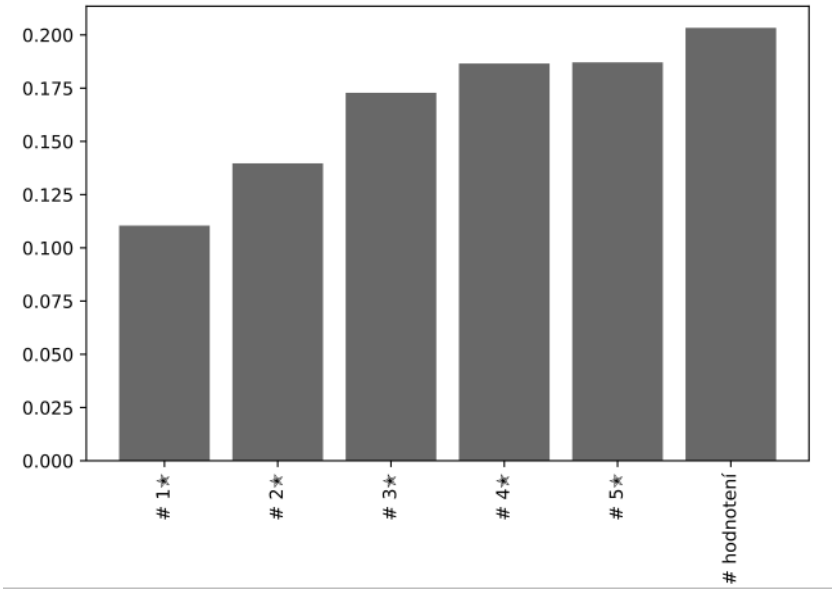


Obrázok 6.8: Matica zámen pre model klasifikátora naučený podľa počtov hodnotení

Celková presnosť klasifikátora je 57.15 %. Túto presnosť sme počítali nasledovne:

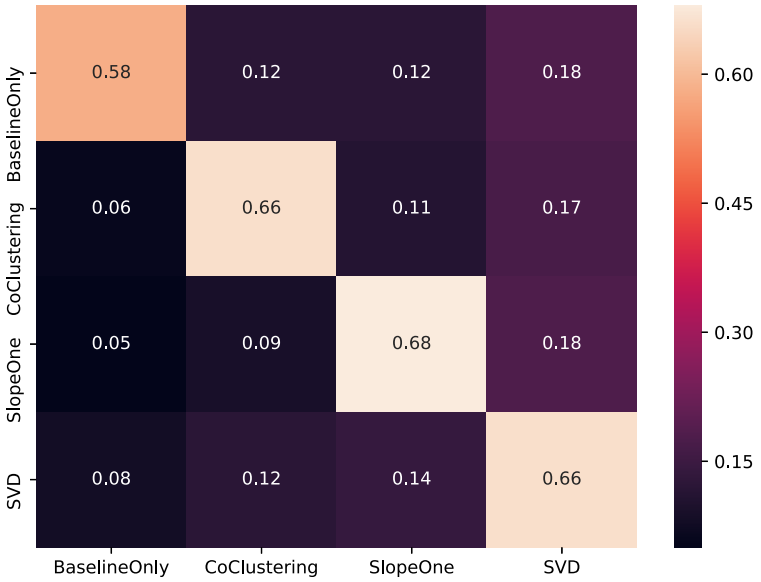
$$presnosť = \frac{\|správne\ klasifikovaní\ používatelia\|}{\|všetci\ používateľa\ v\ testovacej\ sade\|} \quad (20)$$

Graf na Obrázok 6.9 zobrazuje dôležitosť jednotlivých čŕt pre naučený model klasifikátora. Z grafu vyplýva, že celkový počet hodnotení a počet hodnotení, pri ktorých používateľ ohodnotil film na 4 a 5 hviezdičiek, sú pre klasifikáciu najpodstatnejšie.



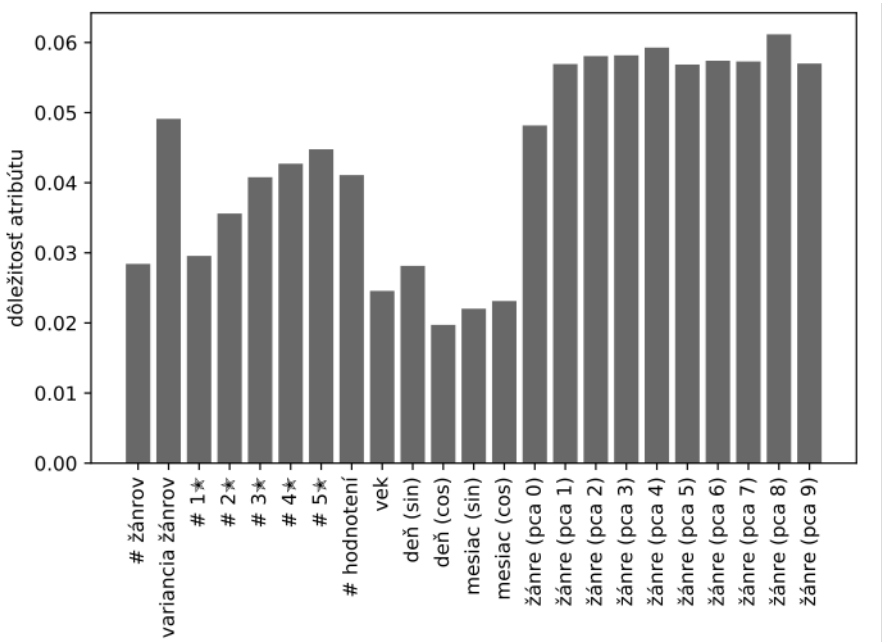
Obrázok 6.9: Dôležitosť histogramu hodnotení a počtu hodnotení pre naučený model klasifikátora pri použití výlučne týchto čŕt

Keď sme klasifikátor s rovnakými parametrami naučili na modeloch používateľov, ktoré obsahovali črty podľa kapitoly 5.2, klasifikátor dosiahol presnosť 64.72 %, teda vyššiu, ako pri využití výlučne iba počtov hodnotení. Matica zámen je zobrazená na obrázku Obrázok 6.10 a dôležitosť čŕt na Obrázok 6.11.



Obrázok 6.10: Matica zámen pre model klasifikátora naučený podľa viacerých čŕt

Pri porovnaní oboch matíc zámen môžeme sledovať nárast presnosti klasifikácie jednotlivých metód odporúčania. Z týchto výsledkov môžeme usudzovať, že kontextové a preferenčné informácie dokážu byť využité na určenie najvhodnejšej metódy odporúčania pre používateľa.



Obrázok 6.11: Dôležitosti čŕt pre naučený model klasifikátora využívajúci informácie o žánroch, hodnoteniach a časových čŕt

Pri porovnaní presností nášho hybridného riešenia a odporúčacích metód integrovaných v hybridnom systéme v Tabuľka 6.2 vidíme, že navrhnutá metóda neposkytuje vyššiu presnosť od metódy Slope One, ktorá dosiahla najvyššiu presnosť. Túto nižšiu presnosť hybridného odporúčania pripisujeme nedostatočnej úspešnosti klasifikácie, nakoľko

optimálna presnosť hybridu dosahuje v metrike *P@3* až o 5.3 % vyššiu presnosť od druhej najpresnejšej metódy.

Tabuľka 6.2: Porovnanie presnosti a úplnosti hybridu a integrovaných metód. $DIFF_{P@10}$ je priemerný rozdiel v presnosti medzi najpresnejšou a druhou najpresnejšou metódou odporúčania použitou v hybridnom systéme. Vyjadrené v P@10.

Metrika/Metóda	Presnosť			Úplnosť			NDCG	$DIFF_{P@10}$
	P@3	P@5	P@10	R@3	R@5	R@10		
Baseline Only	0.0861	0.0868	0.0866	0.0065	0.0111	0.0236	0.2078	0.0439
Co-Clustering	0.0933	0.0910	0.0864	0.0081	0.0127	0.0229	0.2155	
Slope One	0.1036	0.0998	0.0933	0.0088	0.0138	0.0255	0.2373	
SVD	0.0868	0.0839	0.0836	0.0067	0.0110	0.0211	0.1964	
Hybrid	0.0993	0.0961	0.0901	0.0081	0.0129	0.0235	0.2240	
Opt. hybrid	0.1566	0.1357	0.1119	0.0151	0.0212	0.0345	0.3312	

K zmene najlepšej metódy vplyvom zmeny kontextu a preferencií používateľa došlo v 64.46 % prípadoch, pričom v 39.87 % prípadoch sa používateľovi zmenila predikovaná metóda raz. V 20 % prípadoch sa stalo, že vplyvom zmeny kontextovo-preferenčného modelu mu bola vybraná každá metóda z množiny aspoň raz.

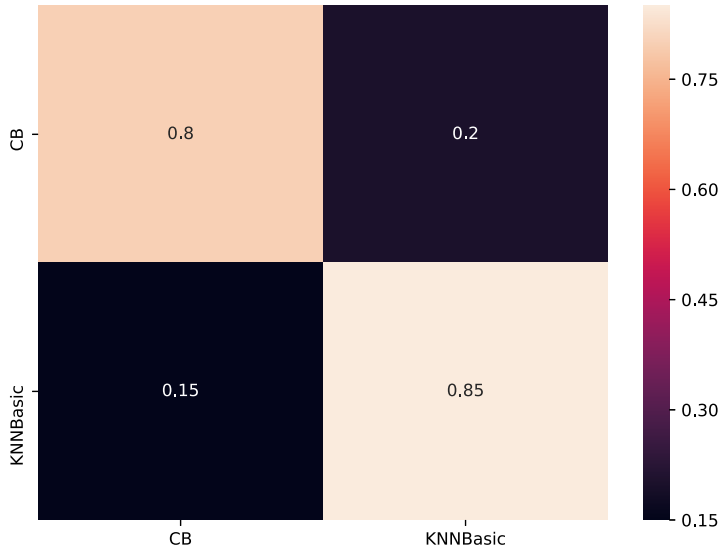
Pri analýze chybne klasifikovaných inštancií sme zistili, že najčastejšie sa nesprávne klasifikovala trieda *Baseline Only* a najčastejšie sa nesprávne klasifikovalo do triedy *SVD*. V prípadoch, kde došlo k nesprávnej klasifikácii, bola potenciálna presnosť odporúčania *P@10 (NDCG)* rovná *0.1028 (0.3088)* pričom zlou klasifikáciou bola dosiahnutá presnosť *0.0739 (0.165)*.

Z doterajších výsledkov vyplynula ďalšia výskumná otázka, akú presnosť klasifikácie a odporúčania dosahuje navrhnutá metóda, keď klasifikujeme z menšej množiny tried. Pre zvýšenie rozdielov medzi jednotlivými technikami v zmysle analýzy z kapitoly 6.2 sme vyskúšali predikovať medzi metódou obsahového odporúčania a metódou kolaboratívneho filtrovania.

6.4 Evaluácia hybridného odporúčania – integrované CB a CF

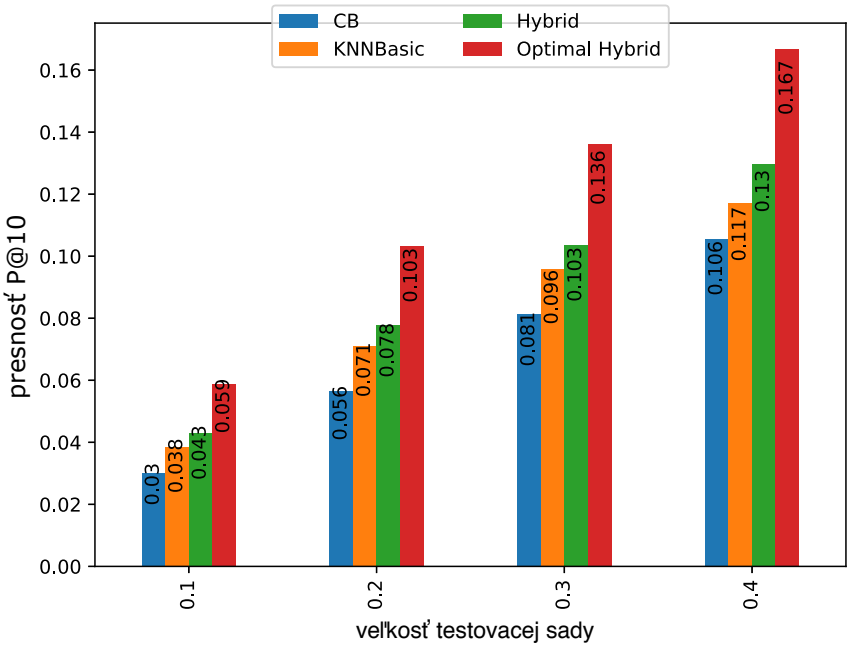
V zmysle výskumnej otázky *V3* a *V5* sme v nasledujúcom experimente zvolili kombináciu metód, pri ktorých rozdiel v dôsledku odlišnej techniky odporúčania je oveľa razantnejší, ako rozdiely medzi metódami kolaboratívneho filtrovania. Vykonali sme experiment, v ktorom sme do navrhovaného systému integrovali metódu obsahového odporúčania a metódu kolaboratívneho filtrovania. Rozdiel týchto techník sa prejavil aj v presnosti klasifikácie a celkovej presnosti odporúčania prostredníctvom realizovanej hybridnej metódy. Presnosť klasifikácie je zobrazená na matici zámen na obrázku Obrázok 6.12. V tomto experimente sme vybrali metódu kolaboratívneho filtrovania *KNN*

Basic, pretože v štatistickom porovnaní z kapitoly 6.2 bola od metódy obsahového odporúčania najviac odlišná.



Obrázok 6.12: Matica zámen pre klasifikáciu medzi CB a CF metódou

Presnosť hybridného odporúčacieho systému v tejto konfigurácii prevýšila presnosti jednotlivých odporúčacích metód. Porovnanie je na Obrázok 6.13. Zhrnutie všetkých veličín meraných pri tomto experimente je v Tabuľka 6.3.



Obrázok 6.13: Porovnanie presností hybridu a integrovaných metód

Tento experiment ukazuje existenciu rozdielu medzi obsahovým odporúčaním a kolaboratívnym filtrovaním z pohľadu klasifikácie za použitia kontextovo preferenčného modelu používateľa. Tento rozdiel sa prejavil v celkovej presnosti hybridnej metódy, ktorý dosiahla vyššiu presnosť, ako druhá najpresnejšia metóda *KNN Basic*. Optimálna presnosť hybridného systému dosahuje približne o 32 % lepšiu

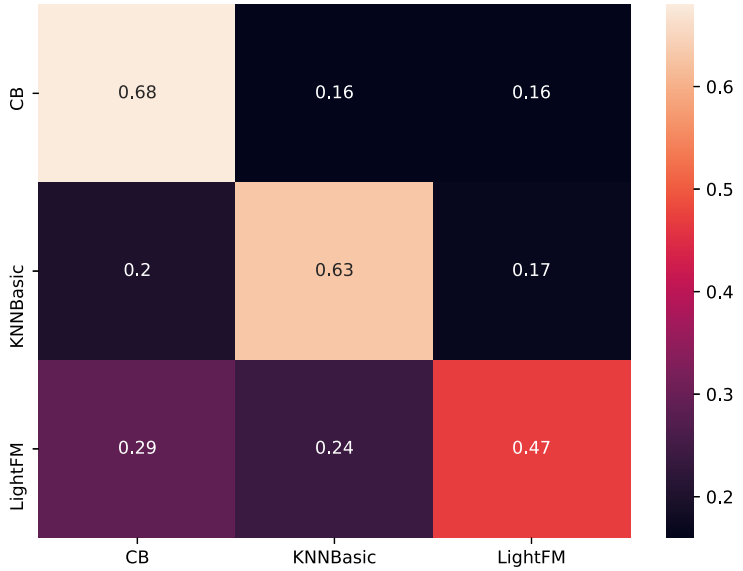
presnosť v porovnaní so skutočnou presnosťou. Táto strata je spôsobená nedokonalou klasifikáciou. V porovnaní s predošlým experimentom je presnosť všeobecne nižšia, čo je spôsobené absenciou tých metód kolaboratívneho filtrovania, ktoré túto presnosť zvyšovali (napr. Slope One). Z analýzy vyplýva, že 77.45 % používateľom bola odporučená iná metóda po aktualizácii ich kontextovo-preferenčného modelu vplyvom nových hodnotení a zmeny času čo znamená, že vplyvom zmeny preferencií používateľa a kontextu došlo k zmene najpresnejšej metódy pre používateľa.

Tabuľka 6.3: Porovnanie presnosti a úplnosti hybridu a integrovaných metód

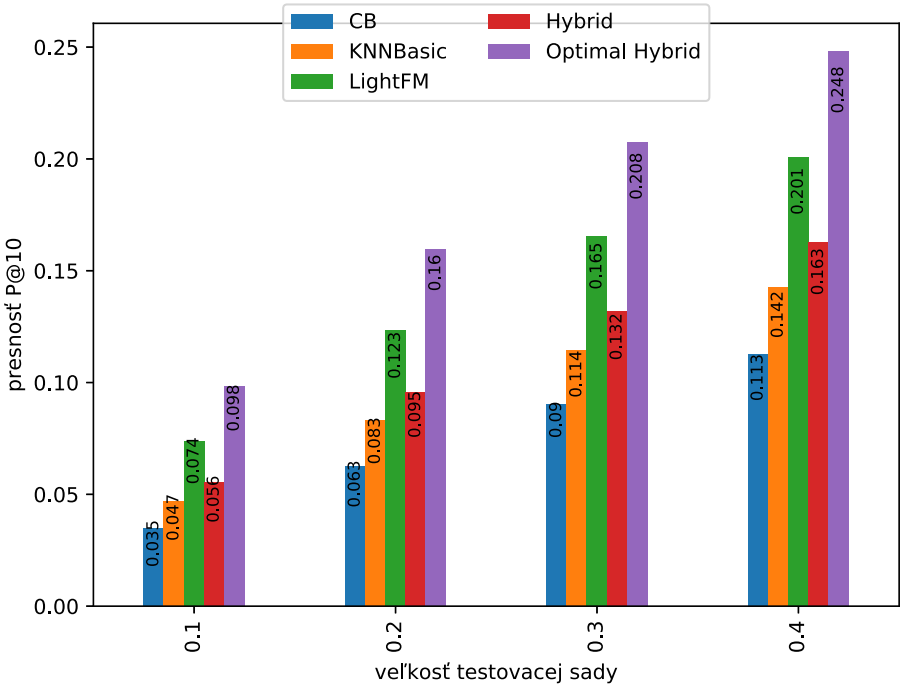
Metrika/Metóda	Presnosť			Úplnosť			NDCG	$DIFF_{P@10}$
	P@3	P@5	P@10	R@3	R@5	R@10		
CB	0.0672	0.0727	0.0740	0.0093	0.0155	0.0280	0.2177	0.1003
KNNBasic	0.0805	0.0781	0.0777	0.0063	0.0103	0.0202	0.1998	
Hybrid	0.0925	0.0924	0.0887	0.0098	0.0160	0.0288	0.2371	
Opt. hybrid	0.1303	0.1257	0.1170	0.0145	0.0233	0.0411	0.3285	

6.5 Evaluácia hybridného odporúčania – integrovaných viac metód

V ďalšom experimente sme integrovali metódu odporúčania založenú na obsahu, metódu kolaboratívneho filtrovania KNN Basic a hybridnú metódu LightFM. Tieto tri metódy sme sa rozhodli kombinovať z dôvodu ich rozličnosti, nakoľko každá z metód využíva inú techniku odporúčania. V tomto experimente sa venujeme výskumnej otázke *V5* a overujeme, či vlastnosti techník je možné rozlíšiť na základe kontextovo-preferenčného modelu používateľa. Na klasifikáciu sme v tomto experimente použili klasifikátor XGB, nakoľko dosahoval vyššiu presnosť klasifikácie a tiež vyššiu presnosť odporúčania v porovnaní s klasifikátorom Náhodný les. Porovnanie s klasifikátorom Náhodný les je uvedené v nasledujúcej kapitole.



Obrázok 6.14: Matica zámen klasifikátora použitého v analyzovanej konfigurácii



Obrázok 6.15: Porovnanie presností odporúčania v analyzovanej konfigurácii

Na matici zámen na Obrázok 6.14 môžeme vidieť presnosť klasifikátora a na Obrázok 6.15 celkovú presnosť odporúčacieho systému. Ako z grafu zobrazujúci presnosti vyplýva, optimálna presnosť hybridu prevyšuje presnosti odporúčacích metód, pričom LightFM dosahuje vyššiu presnosť, ako je skutočná dosiahnutá presnosť našej hybridnej metódy. Nízka presnosť hybridu je zapríčinená situáciami, kedy klasifikátor nesprávne predikuje metódu odporúčania a táto metóda dosahuje výrazne nižšiu presnosť, ako skutočná trieda. Priemerný rozdiel v presnosti v situáciách, kde došlo k nesprávnej klasifikácii, je *P@10 16.29 %*, priemerný rozdiel NDCG je *0.39* a recall *5.23 %*.

Z vykonaného experimentu ďalej vyplýva, že k zlej klasifikácii dochádza najčastejšie v prípade klasifikácie metódy LightFM. Až *27.19 %* zo všetkých klasifikácií tvorí nesprávna klasifikácia triedy LightFM. Najčastejšie sú triedy nesprávne klasifikované do triedy CB (*34.15 %*) a KNN Basic (*19.67 %*). Do LightFM sa nesprávne klasifikovalo 14.24 *%* klasifikácií.

Nesprávna klasifikácia závisí aj od množiny dát, z ktorej boli generované vektory atribútov. V implementovanom odporúčacom systéme dochádza ku generovaniu sady pre klasifikátor viacnásobným delením dátovej sady v rôznych úsekoch definovaných percentuálnym pomerom trénovacej a testovacej sady a následným trénovaním a testovaním metód odporúčania nad týmito sadami. Z analyzovanej konfigurácie vyplýva, že čím bola trénovacia sada menšia, tým častejšie dochádzalo k nesprávnej klasifikácii. V prípadoch, kedy sada obsahuje *10 %* používateľovej spätnej väzby, dochádza k nesprávnej klasifikácii takmer v *55 %* prípadov. Ak je sada tvorená *90 %* používateľovej spätnej väzby, k nesprávnej klasifikácii dochádza v *38.74 %* prípadov. Tabuľka 6.4 zobrazuje porovnanie dosiahnutých presností.

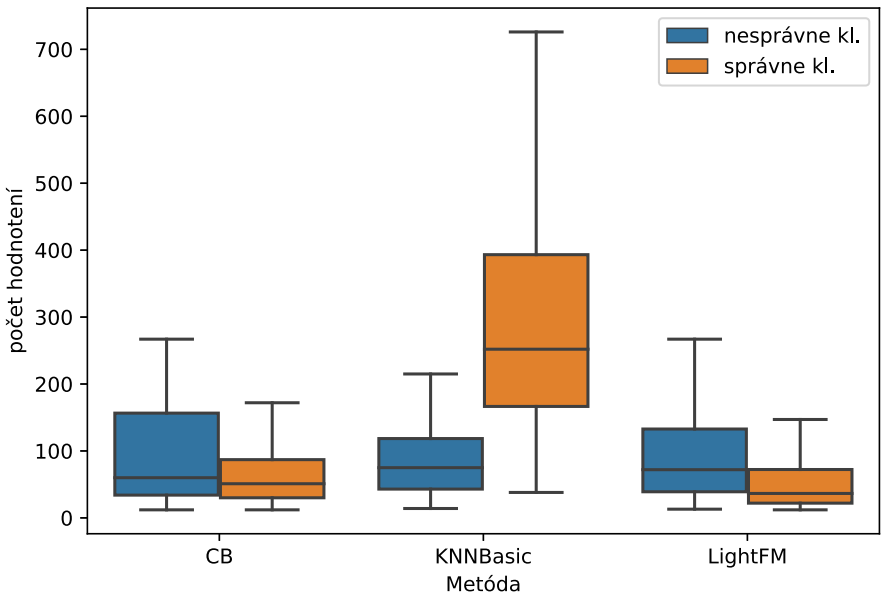
Tabuľka 6.4: Porovnanie metrík hybridu a integrovaných metód

Metrika/Metóda	Presnosť			Úplnosť			NDCG	$DIFF_{P@10}$
	P@3	P@5	P@10	R@3	R@5	R@10		
CB	0.0703	0.0744	0.0751	0.0094	0.0156	0.0285	0.2210	0.1133
KNN Basic	0.1012	0.0989	0.0967	0.0082	0.0132	0.0265	0.2405	
LightFM	0.1579	0.1506	0.1407	0.0200	0.0318	0.0582	0.3594	
Hybrid	0.1216	0.1187	0.1131	0.0133	0.0216	0.0396	0.2882	
Opt. hybrid	0.2343	0.2112	0.1785	0.0299	0.0448	0.0744	0.4933	

Z experimentu vyplýva, že klasifikátor naučený na kontextovo-preferenčných modeloch používateľov dokáže presnejšie rozlíšiť metódu obsahového odporúčania od kolaboratívneho filtrovania KNN Basic, ako hybridnú metódu LightFM kombinujúcu oba prístupy týchto techník. V nasledujúcej časti analyzujeme inštancie, ktoré klasifikátor nedokázal správne klasifikovať.

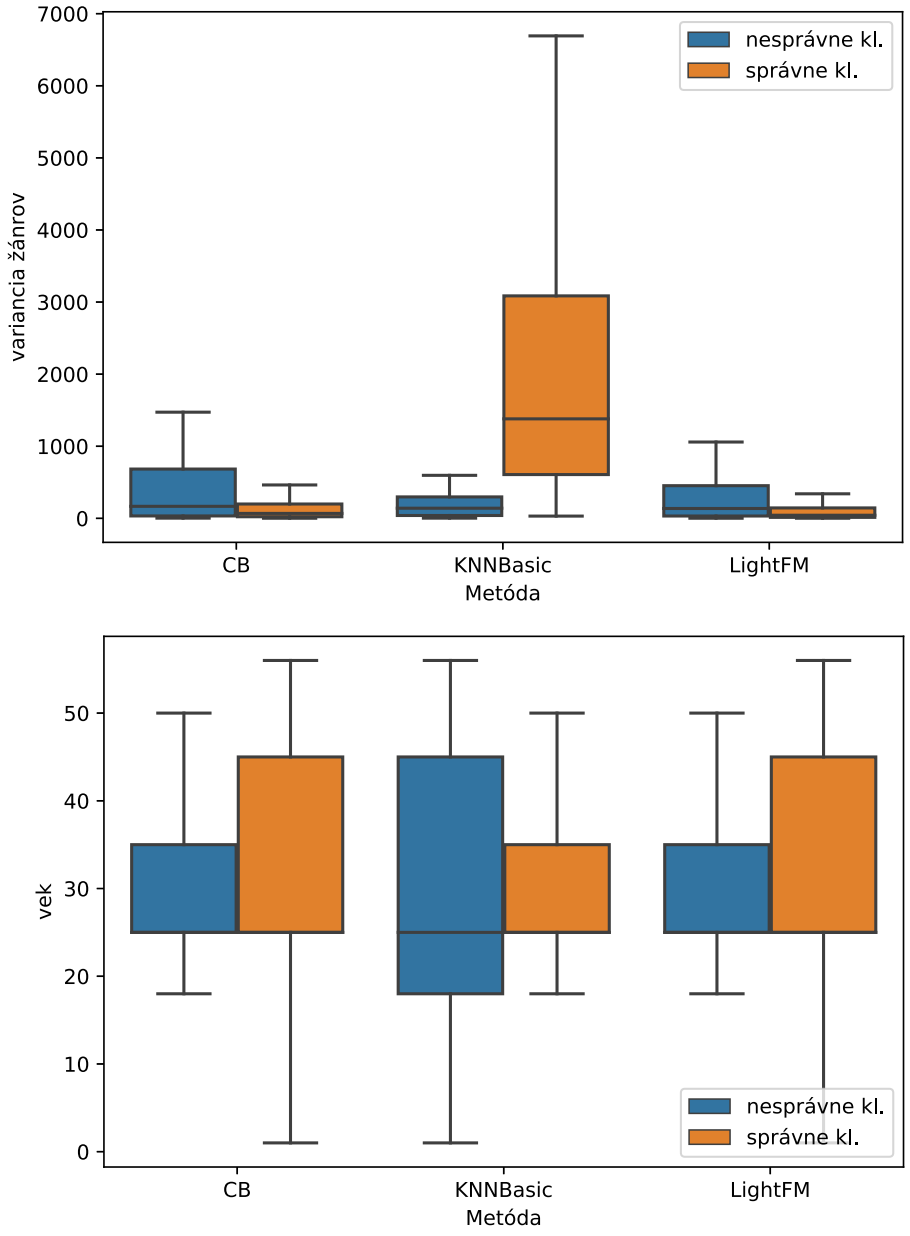
6.5.1 Analýza nesprávne klasifikovaných inštancií

Na analýzu nesprávne klasifikovaných inštancií sme použili kontextovo-preferenčné modely tých používateľov, ktorých nesprávna klasifikácia spôsobila razantné zníženie presnosti hybridného odporúčania. Ide o prípady, v ktorých nesprávnou klasifikáciou došlo k poklesu presnosti NDCG o viac ako *0.5*. Tieto inštancie sme porovnali so správne klasifikovanými prípadmi. Obrázok 6.16 ukazuje rozdiel v počte hodnotení medzi správne a nesprávne klasifikovanými inštanciami. Zistili sme, že nesprávne klasifikované inštancie predstavujú odľahlé body (angl. outliers) vymykajúce sa intervalu definovaným horným a dolným kvantilom. Odľahlé body sú inštancie, ktoré sa svojimi hodnotami výrazne líšia od väčšiny inštancií z danej kategórie. Tieto inštancie vzhľadom na ich diverzitu spôsobujú nesprávnu klasifikáciu, pretože svojimi vlastnosťami sa nehodia do triedy, do ktorej patria.



Obrázok 6.16: Porovnanie správne a nesprávne klasifikovaných inštancií - počet hodnotení

Na obrázku Obrázok 6.17 sú zobrazené rozdiely vo variancii žánrov a veku používateľa. Môžeme tu tiež pozorovať výkyvy v hodnotách horného kvantilu, pričom rozdiely sú najväčšie medzi inštanciami triedy KNN Basic, korej nesprávne klasifikované inštancie majú vlastnosťami bližšie k triedam CB a LightFM. Práve tieto rozdiely spôsobujú nesprávnu klasifikáciu. Pri ostatných črtách z dôvodu ich viacrozmernosti nedokážeme vizualizovať ich rozdiely.



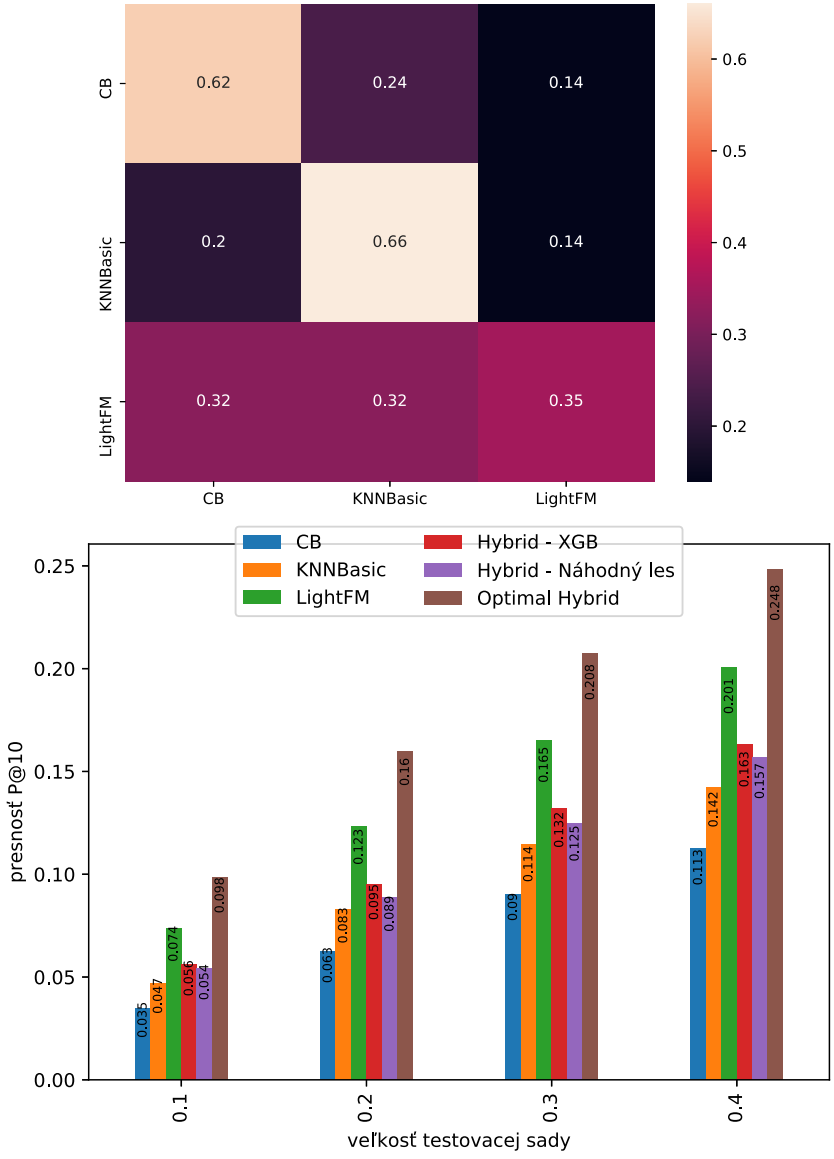
Obrázok 6.17: Porovnanie správne a nesprávne klasifikovaných inštancií - variancia žánrov a vek používateľa

6.6 Porovnanie klasifikátorov Náhodný les a XGB

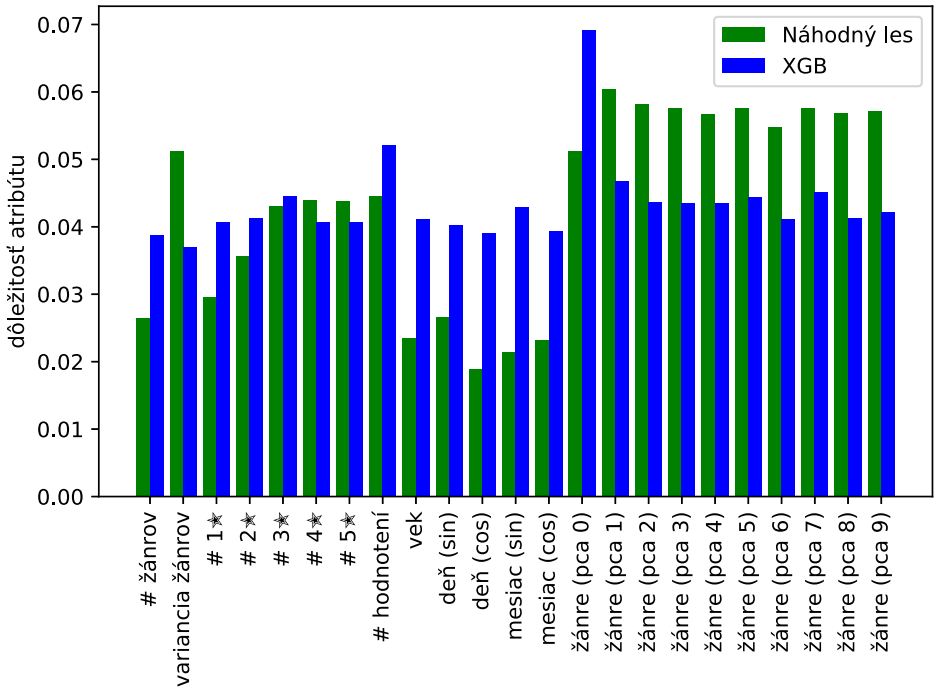
Klasifikátor náhodný les poskytoval presnejšiu klasifikáciu vo väčšine experimentov. V experimente z kapitoly 6.5 dosiahol väčšiu presnosť klasifikácie algoritmus XGB, čo tiež viedlo k zvýšeniu presnosti odporúčania hybridnej metódy. Na porovnanie klasifikátora Náhodný les a XGB sme preto použili trénovaciu a testovaciu sadu z tejto

kapitoly. Obrázok 6.18 na matici zámen zobrazuje presnosť klasifikátora Náhodný les. V porovnaní s presnosťou XGB na obrázku Obrázok 6.14 pozorujeme nárast presnosti odporúčania obsahovej metódy a metódy LightFM, ktorá vzhľadom na najvyššiu presnosť v porovnaní so zvyšnými dvoma metódami je kľúčová na zvýšení presnosti odporúčania hybridnej metódy.

Diagram na Obrázok 6.19 ukazuje, že kontextové časové črty boli na trénovanie klasifikátora viac využité algoritmom XGB, zatiaľ čo preferenčné informácie o žánroch boli viac využité Náhodným lesom. Obrázok 6.18 na grafe dole ukazuje, že použitím Náhodného lesa na predikciu najpresnejšej metódy odporúčania dosiahlo hybridné odporúčanie nižšiu presnosť, ako keď bol na klasifikáciu použitý XGB. Nižšiu presnosť prisudzujeme menej efektívnejšiemu využitiu čŕt pri trénovaní rozhodovacích stromov, ktoré mohlo byť spôsobené aj výskytom odľahlých bodov v zmysle predošlej analýzy nesprávne klasifikovaných inštancií.



Obrázok 6.18: Matica zámen klasifikátora Náhodný les a porovnanie presností odporúčania



Obrázok 6.19: Porovnanie dôležitostí čŕt pre Náhodný les a XGB

6.7 Sumarizácia

Z vyššie vykonaných experimentov a analýz vyplýva niekoľko záverov, ktoré zhrnieme v tejto podkapitole v podobe odpovedí na výskumné otázky.

V1: *Existuje pre každú odporúčaciu metódu skupina používateľov, u ktorých metóda odporúčania dosahuje najvyššiu presnosť v porovnaní s ostatnými?*

Ukázali sme, že pre každú odporúčaciu metódu existuje takáto skupina používateľov. Používatelia v jednotlivých skupinách majú podobné vlastnosti, ktoré je možné využiť pri zaradení nového používateľa do jednej zo skupín.

V2: *Na základe akých charakteristík je možné rozlíšiť jednotlivé skupiny používateľov?*

Z viacerých analyzovaných čŕt, ktoré tvoria kontextovo-preferenčný model používateľa sme zistili, že počet hodnotení hrá dôležitú rolu pri klasifikácii používateľa do triedy odporúčania (ktorú môžeme interpretovať ako spomínanú skupinu používateľov v zmysle *V1*). Medzi ďalšie črty, ktoré sa prejavili ako užitočné patrí variancia žánrov, histogram žánrov a ďalšie informácie extrahované zo štatistiky týkajúcej sa žánrov, ktoré používateľ sleduje. Kontextové informácie ako deň v týždni a mesiac sú taktiež v každej skupine odlišné. Z demografických údajov sa ako jediný užitočný prejavil vek používateľa.

V3: *Pri akej kombinácii odporúčacích metód poskytuje metóda adaptívneho výberu najvyššiu presnosť*

Presnosť metódy adaptívneho výberu závisí od presností metód, ktoré sú v hybridnom systéme integrované. Nakoľko najvyššiu presnosť z množiny nami použitých metód dosahuje metóda LightFM, metóda adaptívneho výberu dosiahne najvyššiu presnosť pri použití tejto metódy.

V4: *Aké zlepšenie presnosti odporúčania dokáže priniesť adaptívny výber odporúčacej metódy z množiny dostupných metód pre používateľa oproti využitiu klasických prístupov?*

Pri tejto otázke je dôležité rozlišovať potenciálnu (optimálnu) presnosť a presnosť, ktorú sa nám podarilo dosiahnuť. V prípade optimálnej presnosti dochádza k nárastu 20 až 50 % v porovnaní s najpresnejšou integrovanou metódou odporúčania, čo je veľmi vysoký nárast v presnosti hodný ďalšieho skúmania. V prípade reálnej presnosti sme dosiahli nárast presnosti iba v jednom z experimentov. Ako sme v analýze ukázali, nízka reálna presnosť súvisí s odľahlými bodmi, ktoré spôsobujú nízku presnosť klasifikácie. Intuitívne povedané, skupiny používateľov sa z časti prekrývajú a to spôsobuje pre odľahlé inštancie ich nesprávnu klasifikáciu.

V5: *Existuje vzťah medzi technikami odporúčania a charakteristikami skupín používateľov?*

Ukázali sme, že medzi technikou odporúčania založenou na obsahu a kolaboratívnym filtrovaním existuje rozdiel, ktorý je možné vyjadriť kontextovo-preferenčným modelom používateľa. Okrem krabicových diagramov bol rozdiel viditeľný aj na matici zámen, kde sa klasifikovalo do týchto tried. Z toho usudzujeme, že charakteristiky používateľov, ktorým najpresnejšie odporúča technika obsahového odporúčania, sú odlišné od charakteristík používateľov, ktorým najpresnejšie odporúčajú metódy kolaboratívneho filtrovania.

7 Zhodnotenie

Personalizované odporúčanie produktov používateľom má široké uplatnenie v rôznych odvetviach čeliacich informačnému zahlteniu. V tejto práci sme analyzovali techniky odporúčania, opísali spôsob ich fungovania, uviedli výhody a obmedzenia každej z nich.

Ďalej sme v tejto práci navrhli metódu adaptívneho výberu odporúčacej metódy, ktorá v rámci jednej domény pre každého používateľa určí, ktorá metóda z množiny integrovaných metód odporúčania mu poskytne optimálne odporúčania. Túto problematiku riešime ako klasifikačný problém, kde používateľa klasifikujeme do triedy odporúčania. V kapitole 6 - Overenie sme vykonali evaluáciu realizovaného riešenia a ukázali sme, že použitím kontextových a preferenčných informácií o používateľovi je možné predikovať optimálnu metódu. Ukázali sme, že existujú skupiny používateľov, pre ktoré určité algoritmy odporúčajú presnejšie a na vybranej množine čŕt sme vizualizovali rozdiely naprieč metódami. Zistili sme, že črty ako počet hodnotení, histogram žánrov, variancia žánrov, vek používateľa, aktuálny deň v týždni a mesiac majú vplyv na výber najpresnejšej odporúčacej metódy. Tiež sme zistili, že črty ako pohlavie používateľa, jeho zamestnanie a histogram kľúčových slov filmov, ktoré používateľ sledoval, nemali vplyv na zlepšenie procesu klasifikácie.

Overením adaptívnej hybridnej metódy odporúčania nad rôznymi množinami metód odporúčania sme zistili, že pri kombinácii metód obsahového odporúčania a metódy kolaboratívneho filtrovania KNN Basic naša metóda prevyšuje presnosť týchto metód. Každopádne, zvýšenie presnosti bolo minimálne a v prípade inej kombinácie metód naša metóda nedosiahla vyššiu presnosť. Túto situáciu pripisujeme nedostatočnej presnosti klasifikácie pri použití zvolených čŕt nad dátovou sadou Movielens-1m. Vyskúšali sme viacero klasifikačných metód s rôznymi nastaveniami parametrov. Náhodný les dosahoval najpresnejšie výsledky klasifikácie, pričom klasifikačná metóda XGBoost tiež dosahovala dobré výsledky a v špecifických prípadoch lepšie, ako Náhodný les.

Aj napriek výsledkom našej metódy vidíme v tomto smere potenciál, nakoľko v prípade optimálnej klasifikácie je možné dosiahnuť až 50 % nárast v presnosti. Ďalšie smerovanie tejto práce vidíme vo vyskúšaní adaptívneho výberu metódy nad inou dátovou sadou, ktorá obsahuje dostatočné množstvo informácií o používateľovi, produktoch a kontexte. Tiež má zmysel uvažovať také metódy odporúčania, ktoré sú technikou odporúčania odlišné a každá rieši niektorý z problémov ako studený štart, úzka špecializácia, alebo sivá ovca. V ďalšom smerovaní má zmysel vyskúšať črty, ktoré redukujú vplyv odľahlých bodov a zvýšia presnosť klasifikácie pre inštancie vymykajúce sa charakteristikám triedy, do ktorej patria, prípadne filtrovať inštancie vykazujúce anomálne vlastnosti už vo fáze trénovania klasifikátora, čo môže viesť k neskoršej presnejšej klasifikácii.

Literatúra

[1] X. Amatriain a J. Basilico, „Past, present, and future of recommender systems: An industry perspective," rev. *Proceedings of the 10th ACM Conference on Recommender Systems*, Boston, ACM, 2016, pp. 211--214.

[2] X. Su a T. M. Khoshgoftaar, „A survey of collaborative filtering techniques," *Advances in artificial intelligence,* p. 4, 9 February 2009.

[3] C. C. Aggarwal, Recommender systems, Springer International Publishing, 2016.

[4] R. Burke, „Hybrid systems for personalized recommendations," rev. *Intelligent Techniques for Web Personalization*, Chicago, Springer, 2005, pp. 133--152.

[5] K. Ma, Content-based Recommender System for Movie Website, Stockholm, 2016.

[6] P. Lenhart a D. Herzog, „Combining Content-based and Collaborative Filtering for Personalized Sports News Recommendations.," rev. *CBRecSys@ RecSys*, Boston, ACM, 2016, pp. 3--10.

[7] Y. Deldjoo, M. Elahi a P. Cremonesi, „Using visual features and latent factors for movie recommendation," rev. *CBRecSys '16: Proceedings of the 3rd Workshop on New Trends on Content- Based Recommender Systems*, Boston, CEUR-WS, 2016, p. 15–18.

[8] R. Burke, "Hybrid Web Recommender Systems," in *The Adaptive Web: Methods and Strategies of Web Personalization*, P. Brusilovsky, A. Kobsa and W. Nejdl, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 377--408.

[9] G. Mustafa and I. Frommholz, "Performance comparison of top N recommendation algorithms," in *Future Generation Communication Technology (FGCT), 2015 Fourth International Conference on*, IEEE, 2015, pp. 1--6.

[10] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Transactions on Information Systems (TOIS),* pp. 143--177, 2004.

[11] D. Lemire a A. Maclachlan, „Slope one predictors for online rating-based collaborative filtering," rev. *Proceedings of the 2005 SIAM International Conference on Data Mining*, Newport Beach, SIAM, 2005, pp. 471--475.

[12] X. Ning a G. Karypis, „Slim: Sparse linear methods for top-n recommender systems," rev. *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, IEEE, 2011, pp. 497--506.

[13] P. Aditya, I. Budi a Q. Munajat, „A comparative analysis of memory-based and model-based collaborative filtering on the implementation of recommender system for E-commerce in Indonesia: A case study PT X," rev. *Advanced Computer Science and Information Systems (ICACSIS), 2016 International Conference on*, IEEE, 2016, pp. 303--308.

[14] F. Ricci, L. Rokach a B. Shapira, „Introduction to recommender systems handbook," rev. *Recommender systems handbook*, New York, Springer, 2011, pp. 1--35.

[15] T. George a S. Merugu, „A scalable collaborative filtering framework based on co-clustering,“ rev. *Data Mining, Fifth IEEE international conference on*, Austin, IEEE, 2005, pp. 4--pp.

[16] B. Sarwar, G. Karypis, J. Konstan a J. Riedl, „Application of dimensionality reduction in recommender system-a case study,“ Minnesota, 2000.

[17] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2008, pp. 426--434.

[18] Y. Koren, „Collaborative filtering with temporal dynamics,“ *Communications of the ACM,* %1. vyd.53, pp. 89--97, 2010.

[19] G. Guo, J. Zhang and N. Yorke-Smith, "A novel recommendation model regularized with user trust and item ratings," *ieee transactions on knowledge and data engineering,* no. 28, pp. 1607--1620, 28 Júl 2016.

[20] Y. Hu, Y. Koren a C. Volinsky, „Collaborative filtering for implicit feedback datasets,“ rev. *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, IEEE, 2008, pp. 263--272.

[21] N. Srebro, J. Rennie a T. S. Jaakkola, „Maximum-margin matrix factorization,“ rev. *Advances in neural information processing systems*, 2005, pp. 1329--1336.

[22] V. Sindhwani, S. S. Bucak, J. Hu a A. Mojsilovic, „One-class matrix completion with low-density factorizations,“ rev. *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, IEEE, 2010, pp. 1055--1060.

[23] R. Mehta a K. Rana, „A review on matrix factorization techniques in recommender systems,“ rev. *Communication Systems, Computing and IT Applications (CSCITA), 2017 2nd International Conference on*, Surat, IEEE, 2017, pp. 269--274.

[24] R. Kumar, B. Verma a S. S. Rastogi, „Social popularity based SVD++ recommender system,“ *International Journal of Computer Applications,* zv. 87, %1. vyd.14, pp. 33--37, February 2014.

[25] N. Hariri, B. Mobasher a R. Burke, „Adapting to User Preference Changes in Interactive Recommendation.,“ rev. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, zv. 15, Chicago, 2015, pp. 4268--4274.

[26] E. Christakopoulou a G. Karypis, „Local item-item models for top-n recommendation,“ rev. *Proceedings of the 10th ACM Conference on Recommender Systems*, Minneapolis, ACM, 2016, pp. 67--74.

[27] W. Ahmed, F. Nur, N. B. Hema a N. Ahmed, An interactive knowledge based recommender system for tourism, Bangladesh: BRAC University, 2017.

[28] A. S. Khan a A. Hoffmann, „Building a case-based diet recommendation system without a knowledge engineer,“ *Artificial Intelligence in Medicine,* %1. vyd.27, pp. 155--179, 2003.

[29] J. Gemmell, T. Schimoler, B. Mobasher a R. Burke, „Resource recommendation in social annotation systems: A linear-weighted hybrid approach,“ *Journal of computer and system sciences,* zv. 4, %1. vyd.78, pp. 1160--1174, July 2012.

[30] M. Göksedef a Ş. Gündüz-Öğüdücü, „Combination of Web page recommender systems,“ *Expert Systems with Applications,* zv. 4, %1. vyd.37, pp. 2911--2922, 2010.

[31] A. S. Lampropoulos, P. S. Lampropoulou a G. A. Tsihrintzis, „A cascade-hybrid music recommender system for mobile services based on musical genre classification and personality diagnosis,“ *Multimedia Tools and Applications,* zv. 1, %1. vyd.59, pp. 241--258, 2012.

[32] R. Nayak, A. Mirajkar, J. Rokade a G. Wadhwa, Hybrid Recommendation System For Movies, Maharashtra: IRJET, 2018.

[33] M. Kula, „Metadata embeddings for user and item cold-start recommendations,“ *arXiv preprint arXiv:1507.08439,* 2015.

[34] R. Burke, „Hybrid recommender systems: Survey and experiments,“ *User modeling and user-adapted interaction,* zv. 12, %1. vyd.4, pp. 331--370, 2002.

[35] D. Billsus a M. J. Pazzani, „User modeling for adaptive news access,“ *User modeling and user-adapted interaction, Springer,* pp. 147--180, 2000.

[36] C. H. Teo, H. Nassif, D. Hill, S. Srinivasan, M. Goodman, V. Mohan a S. Vishwanathan, „Adaptive, personalized diversity for visual discovery,“ rev. *Proceedings of the 10th ACM Conference on Recommender Systems*, Boston, ACM, 2016, pp. 35--38.

[37] C. Z. Felício, K. V. Paixao, C. A. Barcelos a P. Preux, „A multi-armed bandit model selection for cold-start user recommendation,“ rev. *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, Bratislava, ACM, 2017, pp. 32--40.

[38] T. Cunha, C. Soares a A. C. de Carvalho, „CF4CF: Recommending Collaborative Filtering algorithms using Collaborative Filtering,“ rev. *arXiv preprint arXiv:1803.02250*, Vancouver, 2018.

[39] D. G. Ferrari a L. N. de Castro, „Clustering algorithm recommendation: a meta-learning approach,“ rev. *International Conference on Swarm, Evolutionary, and Memetic Computing*, São Paulo, Springer, 2012, pp. 143--150.

[40] M. A. Ghazanfar, „Experimenting switching hybrid recommender systems,“ *Intelligent Data Analysis,* pp. 845--877, 2015.

[41] P. Melville a V. Sindhwani, „Recommender systems,“ rev. *Encyclopedia of machine learning*, New York, Springer, 2011, pp. 829--838.

[42] F. Isinkaye, Y. Folajimi a B. Ojokoh, „Recommendation systems: Principles, methods and evaluation,“ *Egyptian Informatics Journal,* zv. 3, %1. vyd.16, pp. 261--273, 2015.

[43] G. Shani a A. Gunawardana, „Evaluating recommendation systems,“ rev. *Recommender systems handbook*, Springer, 2011, pp. 257--297.

[44] A. Said, B. Kille, B. J. Jain a S. Albayrak, „Increasing diversity through furthest neighbor-based recommendation,“ 12 February 2012.

[45] K. Järvelin a J. Kekäläinen, „Cumulated gain-based evaluation of IR techniques,“ *ACM Transactions on Information Systems (TOIS),* pp. 422--446, 2002.

[46] P. Cremonesi, Y. Koren a R. Turrin, „Performance of recommender algorithms on top-n recommendation tasks,“ rev. *Proceedings of the fourth ACM conference on Recommender systems*, ACM, 2010, pp. 39--46.

[47] G. Adomavicius a A. Tuzhilin, „Context-aware recommender systems,“ rev. *Recommender systems handbook*, Springer, 2011, pp. 217--253.

[48] D. Kaleko, „Feature Engineering - Handling Cyclical Features,“ 30 October 2017. [Online]. Available: http://blog.davidkaleko.com/feature-engineering-cyclical-features.html. [Cit. 20 April 2019].

[49] S. Bouraga, I. Jureta, S. Faulkner a C. Herssens, „Knowledge-based recommendation systems: a survey,“ *International Journal of Intelligent Information Technologies (IJIIT),* zv. 2, %1. vyd.10, pp. 1--19, 2014.

[50] M. E. Wall, A. Rechtsteiner a L. M. Rocha, „Singular value decomposition and principal component analysis,“ rev. *A practical approach to microarray data analysis*, Springer, 2003, pp. 91--109.

Príloha A: Vyhodnotenie plánu práce

V tejto prílohe uvádzame vyhodnotenie plánu práce *DP III*, ktorý je uvedený v tabuľke *A.1*. Tento plán sa v konečnom dôsledku podarilo splniť aj napriek časovým posunom pri určitých častiach ako optimalizácia klasifikátora, ktorá zabrala viac ako dva týždne. Komplikácie pri plnení plánu nastali aj pri zmene metriky na *precision@n*, ktorej zmena si vyžadovala dodatočnú optimalizáciu odporúčacích metód. Optimalizácia klasifikátora a vykonávanie experimentov, ktorých výstup slúžil na ladenie parametrov a zlepšovanie odporúčania, trval približne 5 týždňov semestra simultánne s ďalšími činnosťami definovanými v pláne.

Tabuľka A.1: Plán práce DP III

Týždeň semestra	*Plán*
1	Pokračovanie na implementácii obsahového odporúčania; výber ďalších vlastností do modelu používateľa a overenie ich dôležitosti
2	Pokračovanie na implementácii obsahového odporúčania, vyhodnotenie tejto metódy; výber ďalších metód odporúčania na použitie v našej metóde
3	Dokončovanie obsahového odporúčania; zapojenie nových metód do našej metódy; zmena metriky úspešnosti z RMSE na precision@n
4	Zapojenie nových metód do našej metódy, ich otestovanie; zmena metriky úspešnosti z RMSE na precision@n
5	Zmena metriky z RMSE na precision@n, testovanie a vyhodnotenie metód nad novou metrikou, prípadná zmena a optimalizácia
6	Vyhodnotenie zapojených metód vzhľadom na presnosť, porovnanie s ostatnými metódami; ďalšie úpravy a opravy chýb
7 – 8	Overovanie dôležitosti vybraných vlastností pre klasifikátor, porovnanie presnosti klasifikátora s inými metódami klasifikácie; výber ďalších vlastností
9 – 10	Optimalizácia klasifikátora a celej metódy odporúčania, vyhodnocovanie
11 – 12	Dokončovanie dokumentu, vykonanie dodatočných overení

Príloha B: Dokumentácia k metóde obsahového odporúčania

Táto príloha opisuje návrh, realizáciu a vyhodnotenie metódy odporúčania založenej na obsahu, ktorá bola integrovaná v navrhovanom odporúčacom systéme. Táto metóda je rozšírená a aktualizovaná verzia založená na vlastnej implementácii realizovanej na predmete Vyhľadávanie informácií.

Návrh a realizácia odporúčacej metódy

Navrhovanú metódu môžeme rozdeliť do dvoch fáz: trénovacia fáza a fáza odporúčania. Vo fáze trénovania dochádza k výpočtu podobností medzi každou dvojicou produktov z dátovej sady. Táto podobnosť je vypočítaná na základe atribútov, ktorými produkty disponujú. Môže ísť o číselné, kategorické, alebo textové atribúty charakterizujúce konkrétny produkt. V doméne, v ktorej realizujeme túto prácu, ide o charakteristiky filmov získané zo servera The Movie DB. Konkrétne ide o tieto atribúty:

- popis filmu
- kľúčové slová
- zoznam hercov
- žánre
- rok vydania

Na určenie podobnosti medzi produktami je potrebné jednotlivé atribúty previesť do vektorovej formy. Vzhľadom na typ atribútu sme použili adekvátny vektorizér. Zatiaľ čo žánre a kľúčové slová môžu byť vo vektore reprezentované binárnym reťazcom (One-hot), popis filmu obsahuje neštrukturalizovaný text. Na vektorizáciu popisov sme použili metódu TF-IDF implementovanú vo vedeckej knižnici scikit-learn.

Po vytvorení vektorov metóda vypočíta pre každý atribút kosínusovú podobnosť medzi všetkými dvojicami vektorov a vytvára tak priestorový vektorový model vyjadrujúci podobnosť medzi filmami. Výpočtom kosínusových podobností získame *n* matíc, kde *n* vyjadruje počet atribútov použitých na nájdenie podobností. V ďalšom kroku tieto matice zjednotíme váhovaním jednotlivých matíc nasledovne:

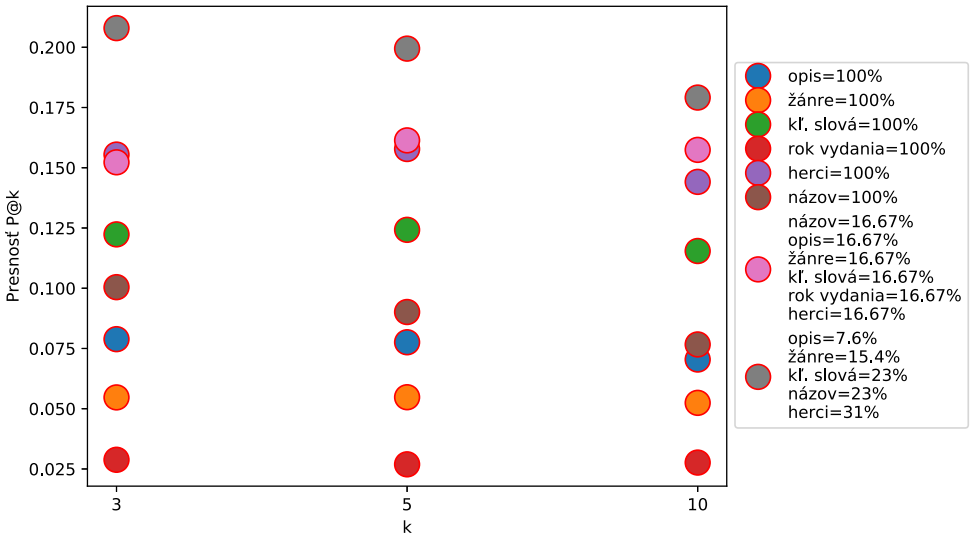
$$M = \sum_{i=1}^{n} \alpha_i m_i$$

Kde m_i vyjadruje maticu podobností, α_i je váha matice m_i a M je konečná matica podobností zohľadňujúca všetky použité atribúty. Po vypočítaní matice metóda uloží pre každý produkt zoznam *N* najpodobnejších.

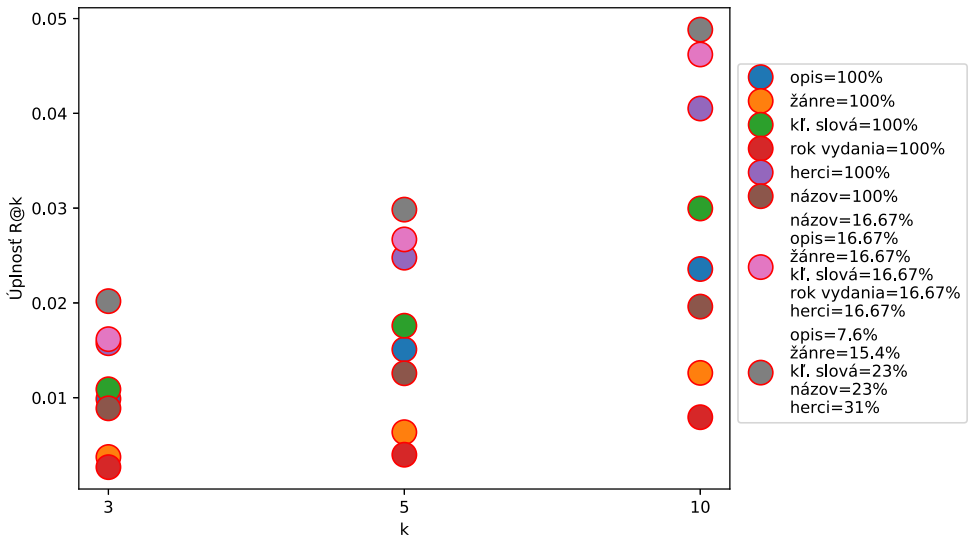
Vo fáze odporúčania dochádza ku generovaniu zoznamu *N* produktov zoradeného podľa relevantnosti. Metóda ku každému produktu, ktorý používateľ hodnotil, získa zoznam jemu podobných produktov spoločne so skóre vyjadrujúcim podobnosť. Tieto produkty následne zoradí vykonaním agregačnej funkcie nad všetkými zoznamami agregujúc na základe *ID* produktu, pričom metóda sčíta skóre podobností pre jednotlivé filmy. Tým docielime, že viac frekventované filmy budú na konečnom zozname umiestnené vyššie.

Vyhodnotenie obsahového odporúčania

Vyhodnotenie odporúčacej metódy založenej na obsahu sme vykonali výpočtom presnosti a úplnosti s dôrazom na dĺžku zoznamu (*precision@k* a *recall@k*). Dosiahnuté presnosti na dátovej sade Movielens-1m objasňuje graf na obrázku B.1 a úplnosti graf na obrázku B.2. Jednotlivé farebné body reprezentujú váhy, ktoré boli použité na natrénovanie modelu. Napríklad modrý bod *opis=100%* znamená, že podobnosť medzi produktami bola počítaná len s využitím atribútu *opis*. Ako z týchto grafov vyplýva, na presnosť má najväčší vplyv zoznam hercov, kľúčové slová, názov a opis. Naopak rok vydania a žánre dosahovali najnižšiu presnosť, keď sa produkty odporúčali iba s použitím jedného z týchto atribútov. Na základe týchto poznatkov sme nastavili váhy jednotlivých atribútov a odporúčacia metóda dosahuje presnosť *P@3 = 20.79 %*, *P@5 = 19.94 %* a *P@10 = 17.91 %*. Úplnosti sa pohybujú od *2.02 %* do *4.88 %* v závislosti od *k*. Metóda dosiahla NDCG o veľkosti *0.436*.



Obrázok B.1: Presnosti odporúčania založeného na obsahu



Obrázok B.2: Úplnosti odporúčania založeného na obsahu

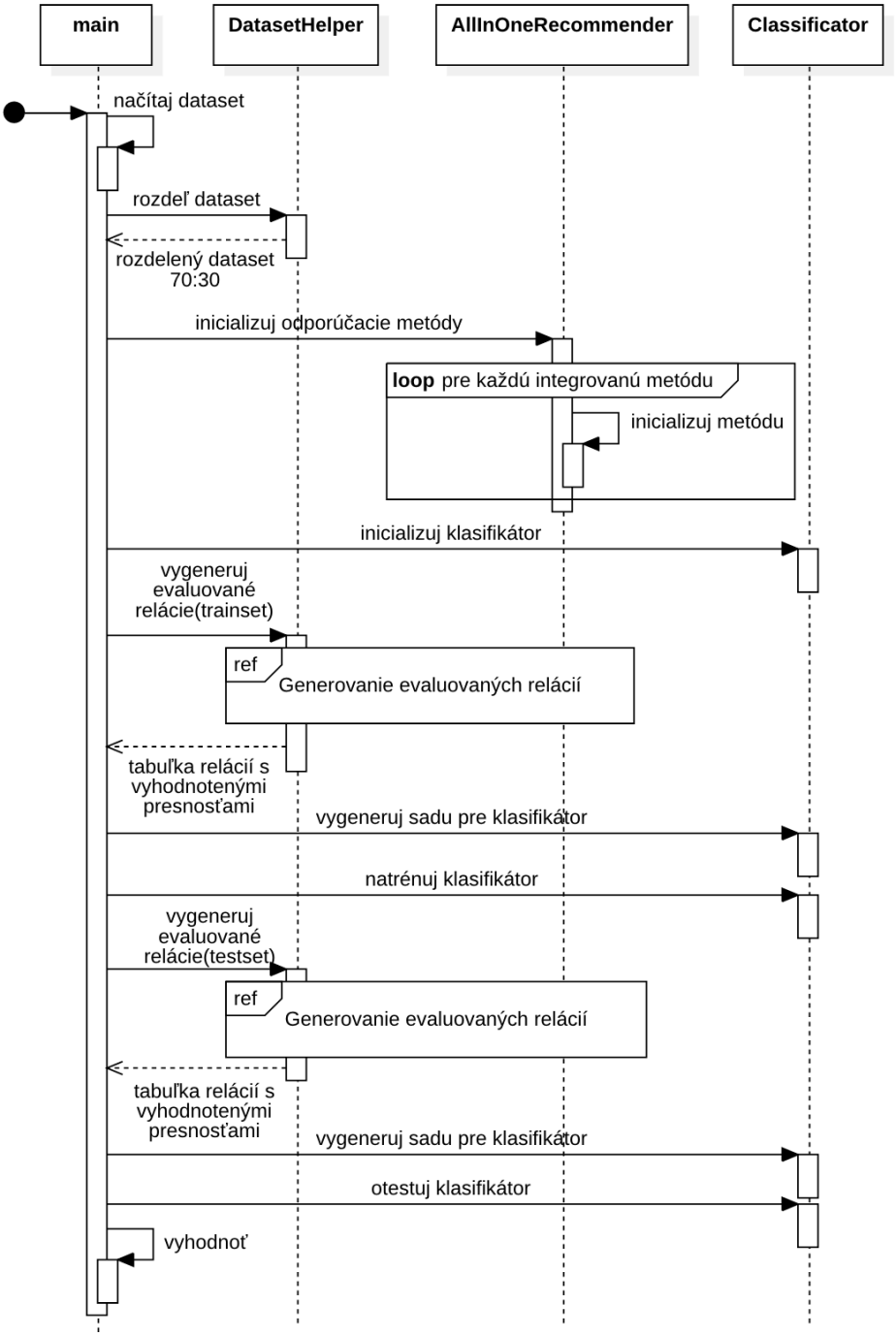
Príloha C: obsah elektronického média

Elektronické médium pribalené k tejto práci obsahuje súbory štruktúrované nasledovne:

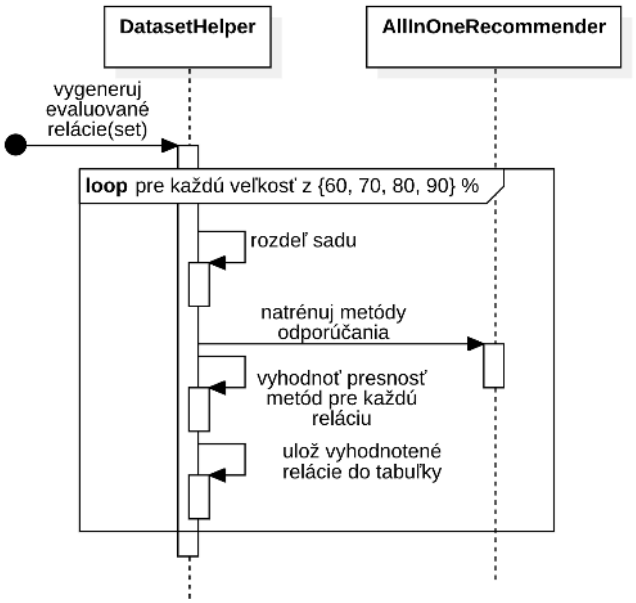
/Peter_Tibensky_DP
- /Zdrojove_kody – obsahuje skripty s implementovaným algoritmom
- /Dokument – obsahuje tento dokument
- /IITSRC – obsahuje článok z konferencie IIT.SRC 2019 a prezentačný poster
- /ML-1m – obsahuje informácie o filmoch stiahnuté zo stránky The Movie DB

Príloha D: Technická dokumentácia

V tejto prílohe zdokumentujeme zdrojový kód implementujúci navrhovanú metódu adaptívneho výberu odporúčacej metódy. Diagram na obrázku D.1 znázorňuje celkový proces odporúčania v režime evaluácie.



Obrázok D.1: Sekvenčný diagram modelujúci navrhnutú metódu



Obrázok D.2: Sekvenčný diagram modelujúci generovanie vyhodnotených relácií

Obrázok D.2 zobrazuje sekvenčný diagram generovania evaluovaných relácií (angl. session). Tie sú vytvorené z trénovacej, resp. testovacej sady, pričom jej inštancie predstavujú používateľov v konkrétnom časovom kontexte. Tieto relácie sú použité na vytvorenie kontextovo-preferenčného modelu používateľa. Na sekvenčnom diagrame z obrázku D.1 Main predstavuje hlavný skript *main.py*, ktorý je zodpovedný za načítanie dátovej sady a interakciu s ostatnými objektami za účelom natrénovania klasifikátora a jeho vyhodnotenia. Metódy ďalších tried a ich význam opisuje nasledujúca tabuľka.

Tabuľka D.1: Opis metód implementovaného algoritmu

Trieda	**Metóda**	**Popis**
DatasetHelper	–	Trieda obsahuje metódy na prácu s dátovou sadou
	split	rozdelí sadu v definovanom pomere
	splitTime	rozdelí sadu v definovanom pomere pri zachovaní časových následností pri každom používateľovi
	splitWithDifferentUsers	rozdelí sadu v definovanom pomere tak, že v každej zložke sa nachádzajú iní používatelia

Trieda	Metóda	Popis
	generateSetForClassifier	vygeneruje tabuľku relácií s vyhodnotenými presnosťami odporúčania
	buildSmartTestset	vytvorí testovaciu sadu obsahujúcu zúžené množstvo produktov na odporučenie
	buildSurpriseTestSet	mapuje sadu do formátu pre Surprise
	buildSurpriseTrainSet	mapuje sadu do formátu pre Surprise
	normalize	každému používateľovi v sade odčíta *n* hodnotení
	loadMovieFeatures	načíta metadáta o filmoch
Metrics	–	Trieda obsahuje metódy na výpočet presnosti odporúčania
	precision_recall	vypočíta presnosť a úplnosť
	f1	vypočíta f1 skóre
	dcg	vypočíta DCG skóre
Tools/UserSimilarity	–	Trieda slúži na zúženie produktov na odporúčanie pre *smart* dátovú sadu
	get_similar_users	získa zoznam podobných používateľov k používateľovi na vstupe
	get_items	získa *n* produktov od podobných používateľov zoradených podľa relevancie
	compute_similarity_matrix	vypočíta podobnosť používateľov
AllInOneRecommender	–	Hlavná fasádová trieda agregujúca metódy odporúčania. V tejto triede sa definujú integrované metódy odporúčania ako aj ich parametre

Trieda	Metóda	Popis
	trainAll	natrénuje všetky integrované odporúčacie metódy
	train	natrénuje konkrétnu odporúčaciu metódu
	testAll	vykoná top-N odporúčanie všetkými metódami
	test	vykoná top-N odporúčanie konkrétnou metódou
	getOnlyBestAlgPerUser	z výstupu metódy testAll získa pre každého používateľa iba inštanciu reprezentujúcu najpresnejšiu metódu
ContentBasedRecommender	–	Trieda implementujúca odporúčanie založené na obsahu.
	fit	natrénuje obsahový odporúčač
	recommendN	odporučí N produktov na základe produktov zo vstupu
	topN	odporučí N produktov každému používateľovi na vstupe
LightFMRecommender	–	Trieda implementujúca LightFM odporúčací systém
	fit	natrénuje odporúčaciu metódu
	recommendN	odporúča každému používateľovi nad množinou všetkých produktov
	recommendNSmart	odporúča každému používateľovi nad zúženou množinou produktov

Trieda	**Metóda**	**Popis**
	topn	rozhranie volajúce metódu *recommendNSmart*
Classificator	–	Trieda implementujúca klasifikačnú metódu. V tejto triede v metóde *train* je možné definovať parameter klasifikátora
	train	natrénuje klasifikátor
	predict	klasifikuje pre každú inštanciu zo vstupu do triedy odporúčania
	prepareSet	transformuje evaluované relácie do kontextovo-preferenčného modelu

Príloha E: Príručka k spusteniu skriptov

Táto príloha obsahuje informácie potrebné pre spojazdnenie skriptov z elektronického média. Tieto skripty implementujú algoritmus navrhnutý v tejto práci. Na správne spustenie skriptov je potrebná dátová sada MovieLens-1m dostupná na adrese *grouplens.org/datasets/movielens/1m* a metadáta o filmoch, ktoré sú umiestnené na elektronickom médiu.

Na elektronickom médiu sa nachádza niekoľko skriptov:

- main.py: skript vykonávajúci evaluáciu tak, ako je prezentované v práci,
- evaluation_contentbased.py: skript vykonávajúci vyhodnotenie obsahového odporúčania,
- evaluation_lightfm.py: skript vykonávajúci vyhodnotenie top-N odporúčania LightFM,
- evaluation_surprise.py: skript vykonávajúci vyhodnotenie top-N odporúčania metód z knižnice Surprise

Na spustenie týchto skriptov sa vyžaduje programovací jazyk *Python* verzie 3.6 a knižnice:

- pandas: pandas.pydata.org
- surprise: surpriselib.com
- matplotlib: matplotlib.org
- seaborn: seaborn.pydata.org
- sklearn: scikit-learn.org
- lightfm: lyst.github.io/lightfm/docs/home.html

Nasledujúca tabuľka zobrazuje vyžadované vstupné parametre pre jednotlivé skripty.

Tabuľka E.1: Vstupné parametre skriptov

Skript	**Parametre**
main.py	1. cesta k súboru movies.dat 2. cesta k súboru ratings.dat 3. cesta k súboru users.dat 4. cesta k priečinku s metadátami o filmoch
evaluation_contentbased.py	1. cesta k súboru movies.dat

Skript	Parametre
	2. cesta k súboru ratings.dat 3. cesta k priečinku s metadátami o filmoch
evaluation_lightfm.py	1. cesta k súboru ratings.csv
evaluation_surprise.py	1. cesta k súboru ratings.csv

Jednotlivé parametre ako aj integrované metódy odporúčania je možné meniť priamo v kóde v relevantných triedach a súboroch. Tabuľka E.2 vyjadruje umiestnenia, v ktorých je možné zmeniť konfiguráciu odporúčacej metódy.

Tabuľka E.2: Umiestnenie parametrov

Parameter	Umiestnenie
integrácia metód odporúčania	Trieda *AllInOneRecommender:* slovník *algs* a metódy *trainAll* a *testAll*
nastavenie klasifikátora	Trieda *Classificator* a jej metóda *train*

Príloha F: Príspevok na konferenciu IIT.SRC 2019

Context-aware Adaptive Personalized Recommendation: A Meta-Hybrid

Peter TIBENSKÝ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
peter.tibensky@icloud.com

Abstract. Recommender systems take place on a wide scale of e-commerce systems reducing the problem of information overload. The most common approach is to choose a recommendation method, which is used by the system to make predictions. However, users vary from each other, thus an one-fits-all approach seems to be a sub-optimal solution. In this paper, we propose a meta-hybrid recommender that uses a machine learning to predict a specific recommender algorithm for each specific session and user. This adaptive selection of the method depends on contextual and preferential information collected about the user.

1 Introduction

In the field of recommender systems, we recognize several techniques like collaborative filtering, content-based, knowledge-based and demographic [8]. Furthermore, there are hybrid recommender systems combining approaches of the mentioned techniques to reduce the impact of their drawbacks and on the other hand strengthen their advantages to increase performance of recommendation [1].

The typical approach is to select a recommender method which provides recommendations to every user in the same fashion. But nevertheless, users in the system behave differently and the way they use the system varies. Each of the methods achieves different performance for different users depending on theirs interactions with the system. For example, content-based methods does better for users rating similar items. Therefore, using various methods whose choice depends on user and contextual information may lead to improve recommendation performance.

In this paper, we propose a meta-hybrid recommender system which utilizes information about users and their context to select a method to generate recommendations. This information is extracted from user's feedback and items he/she rated before. Next, a classifier is used to determine the most precise method for them. We also demonstrate the possibility of use of contextual and preferential information as well as the classifier to adaptively select a method to provide the most precise recommendations.

2 Related work

The problem of picking a specific algorithm in the domain of recommendation was previously studied by several authors. Authors of [2] proposed to use Multi-armed bandit (MAB) algorithms to deal with the challenge of new users and items constantly appearing in the system. Recommender system has to constantly adapt to these changes to provide relevant recommendations to each user. Felicio et al [5] utilized MAB to select the right cluster for an user without having any feedback from him/her. First, they used matrix factorization to predict ratings of users which already have rated items and then cluster them using Euclidean distance function. Then, they computed average ratings for every item and cluster. MAB then select one of the clusters for an active user and items with the highest rating are recommended. Active user's feedback is then used as a reward to recalculate MAB's prediction model. MAB are also used by in many other recommender systems, for instance Amazon Stream service [10] utilizes Thompson sampling, and [7] can adapt to changes in preferences and multiple MAB algorithms can be used to maximize the average utility over each user's session.

Using a recommender method to recommend another recommender method is the next approach. In [3] authors utilized a collaborative filtering to recommend collaborative filtering method that will provide

* Master study programme in field: Software Engineering
Supervisor: Assoc. Professor Michal Kompan, Institute of Informatics, Information Systems and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2019, Bratislava, April 17, 2019, pp. 1–8.

the optimal precision of recommendations on an input dataset. Authors consider input dataset to be active user and CF methods to be items to recommend. As far as the rating matrix is concerned, they model preferences using subsampling landmarkers obtained from a random sample of instances, which determine the performance of algorithms on datasets.

Similar problem was researched by Ferrari [6]. Instead of recommendation, clustering algorithm selection for an input dataset was explored. This method uses meta-knowledge, which includes meta-data about a dataset such as Log_2 of the number of objects, Log_2 of the number of attributes, the proportion of binary, discrete and continuous attributes and the correlation between continuous attributes. All of these meta-attributes does not require any semantics about the content of the dataset. After extracting the meta-attributes, authors used meta-algorithm to learn the relationship between the meta-attributes and the ranking. KNN provides the best result using *t*-test metric.

All of these approaches deal with the problem of strategy selection. While explore-exploit solutions like multi-armed bandit gradually improve their recommendation performance, recommending a recommender method and using meta-knowledge use some kind of information extracted from the dataset. However, these approaches select one method used for all instances in a dataset. Bringing this idea to the level of instances may improve the recommendation performance. Contextual MAB methods are an example of selecting strategy based on user context [4, 9].

3 Proposed method

Our idea is based on the usage of the meta-hybrid recommender method which selects recommender methods for a user according to a user context model to achieve better recommendation performance. User context model includes user's feedback, information derived from feedback and context.

We consider the selection of recommender method to be a classification problem in which user needs to be classified into a class representing the recommender method. The proposed meta-hybrid requires user context model on the input to determine which recommender method provides the most precise results. Recommender algorithm consists of the following steps:

1. Construct user context model,
2. Predict user's class, i.e., recommender to be used,
3. Recommend n items to the user using the predicted method.

3.1 User context model

The classifier utilizes information about users reflecting their preferences and behavior. User context model represents a vector containing these features. Information from the user context model can be divided into the preferential and contextual information. A preferential information is gathered from user's explicit and implicit feedback and reflects the taste of the user. A contextual information contains meta-information extracted from the item descriptions, user's behavior and background known to the recommender system. In the domain of movies, we utilized the following features to be included in user context model: number of ratings, histogram of ratings, histogram of movie genres viewed by the user, variance of the genres and number of the most viewed genres. Number of the most viewed genres is number of genres user watched, which have more than 30% representation compared to the most viewed genre.

3.2 Prediction step

As the recommender candidates we consider several collaborative filtering methods: Baseline Only, Co-Clustering, Slope One and SVD. All of these methods are comparable to each other via *RMSE*[1], which we used to determine their performance. We selected These methods because they use different approaches to predict ratings. We assume their different properties will be reflected later in classification. We used Python library named Surprise[2], which implements these methods.

Subsequently, we train the classifier using a train set consisting of user context models labeled with classes. The labeling is assigned by calculating the metric for each recommender method over each user context model and selecting one with the lowest RMSE.

4 Evaluation

To evaluate proposed approach we follow standard offline evaluation methodology. We used the MovieLens 20M dataset containing over 20 million ratings[3] from approx. 138k users on 27k movies. Since each user has at least 20 ratings, we randomly reduced the amount of ratings for each user. Thanks to this "normalisation" we obtained so-called cold-

[1] RMSE - Root Mean Square Error
[2] surpriselib.com
[3] grouplens.org/datasets/movielens/20m/

start problem. Besides that we enriched the MovieLens dataset with movies meta-data from The Movie DB[4] (i.e., movie description, duration, cast, genres and keywords). As the predictor we used the Random Forest algorithm with parameters: max_depth: unlimited, min_samples_split: 2, min_stamples_leaf: 1, max_features: $\sqrt{numberof features}$, n_estimators: 1000.

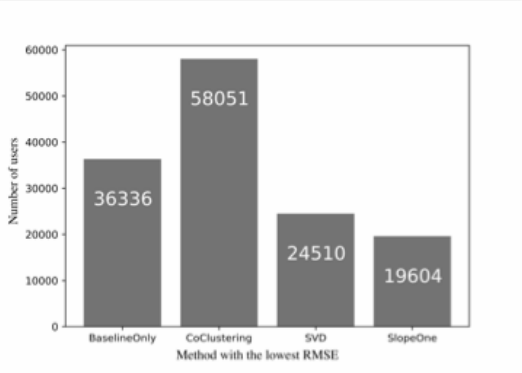


Figure 1. Histogram of methods with the lowest RMSE for all users

RQ1: What is the performance of particular recommenders on MovieLens dataset? Our first research question aimed to find out, how the distribution of the most accurate methods per user on ML-20m dataset looks like. As you can see in Figure 1, Co-Clustering leads over the other methods, however, other methods achieves lower RMSE for 58% of users. Difference between RMSE of the winning method and the second one is an average of *0.095* and the difference between the most accurate method and the others in an average of *0.175*. Comparing with performances published on MyMediaLite[5] and Surprise, the difference is significant and it is worth exploring adaptive method selection.

RQ2: Which features included in a user context model are most important for correct predictions? Based on a domain knowledge, firstly we considered the number of ratings and the distribution of ratings to be contained in classification. This is supported by numerous methods proposed for low activity domains in the literature. By analysis of the relation between the number of ratings and recommender methods' performances (RMSE), we identified the correlation that proves the existence of dependency between the number of ratings user made and method that recommends to the user with the lowest error. Table 1 shows Co-Clustering achieves the lowest RMSE compared to others when a user has on an average of 156 ratings. On the other hand, Baseline Only and Slope One perform their best when a user has 40 - 50 ratings.

Table 1. Number of ratings when methods achieve the lowest RMSE

Algorithm	Number of ratings			
	Average	Quantile		
		25 %	50 %	75 %
Baseline Only	40.66	11	25	52
Co-Clustering	156.05	26	72	188
Slope One	50.74	10	25	64
SVD	83.25	12	37	105

In the evaluation phase, the methodology was as follows:

1. Split dataset into folds A and B, each of them contains different users,
2. Split both folds into train and test tests A_{train}, A_{test}, B_{train}, B_{test},
3. Use fold A_{train} to train recommender methods,
4. Predict ratings using A_{test} and determine which method for which users achieves the lowest RMSE,
5. Train the classifier using A_{test},
6. Use B_{train} to create user context model and classify each user,
7. Train recommender methods using B_{train},
8. Predict ratings by using B_{test},
9. Determine, which method for which users achieves lowest RMSE and compare it to classifier's output

Training the classifier and its testing is performed on folds containing different users to exclude any dependencies between the train and test phase. To explore the impact of attribute number of user ratings, we trained it on user models which contain only number of ratings and histogram of ratings. Confusion matrix in Figure 2 shows Co-Clustering and Slope One are often interchanged, so as Slope One and SVD. Overall precision of the classifier is 44%.

In the next step we extended the user context model by the histogram of genres, variance of genres and the most viewed genres. The classifier achieves higher precision (Figure 3) with overall precision of 46.8%.

[4] themoviedb.org
[5] mymedialite.net

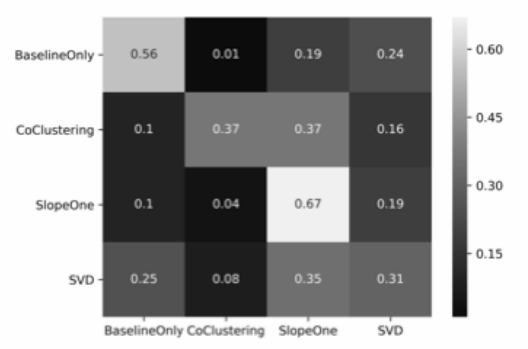


Figure 2. Precision of the classifier when trained only on number of ratings.

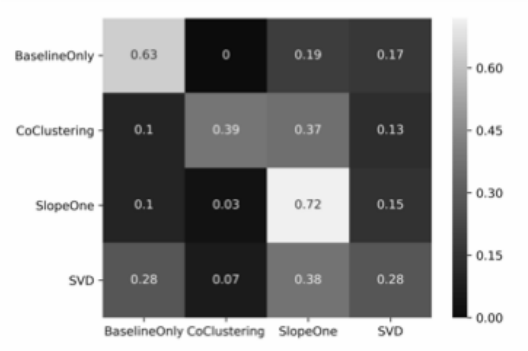


Figure 3. Precision of the classifier when trained on the whole user context model.

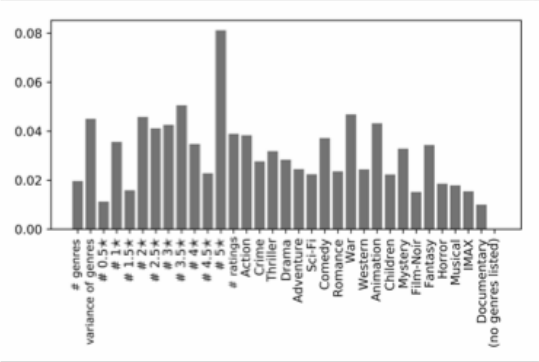


Figure 4. Importance of features used by classifier.

Comparing these two confusion matrices, we came to a conclusion that it is possible to determine the optimal recommender method for each user utilizing contextual and preferential information about them. In Figure 4 we can see features importance. Among the most important feature belongs number of ratings and histogram of ratings, however, histogram of genres and variance of genres also have an influence which leads to better precision of the classifier.

5 Conclusion and future work

In this paper, we proposed a meta-hybrid recommender which uses contextual and preferential information about a user. We showed the possibility to adaptively select recommender method according to user context model. According to our experiments, besides user's feedback, context and content information are also important when selecting a method.

In future work, we will aim to add more features to user context model such as time information due to the hypothesis that taste of users change over time. Moreover, we need to evaluate the recommendation itself, as until now we evaluated only the classifier and not the whole meta-hybrid.

References

[1] Burke, R.: Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 2002, vol. 12, no. 4, pp. 331–370.

[2] C. Aggarwal, C. In: *Advanced Topics in Recommender Systems*, 2016, pp. 411–448.

[3] Cunha, T., Soares, C., de Carvalho, A.C.: CF4CF: recommending collaborative filtering algorithms using collaborative filtering. In: *Proc. of ACM Conf. on Recommender Systems*, ACM, 2018, pp. 357–361.

[4] Edwards, J.A., Leslie, D.S.: Selecting multiple web adverts: A contextual multi-armed bandit with state uncertainty. *Journal of the Operational Research Society*, 2019, pp. 1–17.

[5] Felício, C.Z., Paixão, K.V., Barcelos, C.A., Preux, P.: A multi-armed bandit model selection for cold-start user recommendation. In: *Proc. of Conf. on User Modeling, Adaptation and Personalization*, ACM, 2017, pp. 32–40.

[6] Ferrari, D.G., de Castro, L.N.: Clustering algorithm recommendation: a meta-learning approach. In: *International Conference on Swarm, Evolutionary, and Memetic Computing*, Springer, 2012, pp. 143–150.

[7] Hariri, N., Mobasher, B., Burke, R.: Adapting to user preference changes in interactive recommendation. In: *Int. Joint Conf. on Artificial Intelligence*, 2015.

[8] Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: *Recommender systems handbook*. Springer, 2011, pp. 1–35.

[9] Tekin, C., Turğay, E.: Multi-objective contextual multi-armed bandit with a dominant objective. *IEEE Transactions on Signal Processing*, 2018, vol. 66, no. 14, pp. 3799–3813.

[10] Teo, C.H., Nassif, H., Hill, D., Srinivasan, S., Goodman, M., Mohan, V., Vishwanathan, S.: Adaptive, personalized diversity for visual discovery. In: *Proc. of Conf. on Recommender Systems*, ACM, 2016, pp. 35–38.