FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Využití principů dolování datových toků v doporučovacích systémech
Student:	Bc. Tomáš Chládek
Vedoucí:	Ing. Jaroslav Kuchař, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2019/20

Pokyny pro vypracování

Charakteristiky doporučovaných položek se v čase mění, proto je důležité tyto změny reflektovat v naučeném modelu. Při vhodném návrhu algoritmů pro vytěžování datových toků (angl. data stream mining) je možné využít dynamiky charakteristik v doporučovacích systémech.
- Proveďte rešerši existujících principů a technik, které se při vytěžování datových toků využívají.
- Sestavte vhodné kandidáty, kteří budou dle vybraných metrik vyhodnoceni na vybraném simulovaném datovém toku.
- Navrhněte, implementujte a otestujte vhodnou platformu pro práci s datovými toky.
- Porovnejte naměřené hodnoty mezi kandidáty a tradičními přístupy v doporučování.
- Prezentujte dosažené výsledky včetně shrnutí vhodnosti technik vytěžování datových toků k doporučování zvolených položek.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 6. února 2019

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Diplomová práce

Využití principů dolování datových toků v doporučovacích systémech

Bc. Tomáš Chládek

Katedra softwarového inženýrství
Vedoucí práce: Ing. Jaroslav Kuchař, Ph.D.

4. května 2019

Poděkování

Děkuji stromům za darování toho nejcenějšího, co měly. Děkuji planetě Zemi za život, který nám dala. Děkuji lidem, že všem těmto darům dávají smysl.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo"), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 4. května 2019 .....

Odkaz na tuto práci

Chládek, Tomáš. *Využití principů dolování datových toků v doporučovacích systémech.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.

Abstrakt

Hlavním tématem diplomové práce je využití technik dolování dat z datových toků k doporučování novinových článků. V teoretické části jsou rozebrány principy doporučovacích systémů a jejich testování. Dále jsou popsány principy, které se v algoritmech na dolování dat z datových toků využívají. V rešerši jsou zkoumána existující řešení v doméně novinových článků a platforma, která je k experimentování použita. V části dolování dat je popsán proces získání informací a analyzována data, která jsou použita k experimentům. Proudový algoritmus implementuje různé parametrizovatelné techniky vytěžování datových toků. V experimentální části jsou zkoumány vlivy jednotlivých technik a heuristik na měřené metriky. Experimenty jsou rozděleny do kategorii dle následujících heuristik: náhodný výběr, populárnost, iterátor a nedávno navštívený článek. Testování je provedeno na platformě StreamingRec. V závěru práce je shrnutí vhodnosti technik k doporučování položek v doméně novinových článků.

Klíčová slova doporučovací systémy, vytěžování dat, datové toky, StreamingRec, CLEF NewsREEL challenge, doporučování novinových článků

Abstract

Main topic of this master thesis is usage of data stream mining techniques in news recommendation systems. In theoretical part are described principles of recommendation systems, data mining and data streams. In previous work are revised existing algorithms in this domain and platform, that can be used for evaluation of recommendation system. In the data mining part is described the process of gathering information from the data stream and analyzed data, that are used for experiments. Streaming algorithm implements various parameterized techniques for data stream mining. Experiments are divided by following heuristics: random selection, popularity, iterator and recently visited article. Evaluation of experiments is performed on StreamingRec platform. The conclusion summarizes the benefits of using data stream mining techniques in the news recommendation systems.

Keywords recommendation systems, data mining, data streams, StreamingRec, CLEF NewsREEL challenge, news recommendation

Obsah

Úvod **1**

I Teoretická část **3**

1 Doporučovací systémy **5**
1.1 Principy 5
1.2 Kontext 6
1.3 Vlastnosti doporučování 6
1.4 Forma doporučení 8
1.5 Typy doporučovacích systémů 8
1.6 Testování 11
1.7 Metriky 12

2 Vytěžování dat **17**
2.1 Datový tok 18
2.2 Vytěžování datových toků v doporučovacích systémech 18
2.3 Zpracování datového toku 19
2.4 Vývoj datového toku 19
2.5 Techniky 22

II Rešerše **27**

3 CLEF NewsREEL challenge **29**
3.1 Algoritmy 29
3.2 Technologie 32
3.3 Výsledky 33

4 Datový tok **35**

4.1 Novinové články . . . 35
4.2 Plista . . . 36
4.3 Popis dat . . . 37
4.4 Proces generování transakcí . . . 38

5 Platforma **41**
5.1 Idomaar . . . 41
5.2 StreamingRec . . . 42

III Dolování dat **45**

6 Dataset **47**
6.1 Sběr dat . . . 47
6.2 Extrakce charakteristických rysů . . . 47
6.3 Analýza dat . . . 49

IV Implementace **57**

7 Platforma **59**
7.1 Změny v implementaci . . . 60

8 Streaming algoritmus **61**
8.1 StreamingBuilder . . . 61
8.2 StreamingManager . . . 61
8.3 IFilter . . . 62
8.4 IDataFrame . . . 63
8.5 IStreamingExecutor . . . 63
8.6 ISampler . . . 63
8.7 IHeuristic . . . 64
8.8 Metriky . . . 64

V Experimentální část **67**

9 Experimenty **69**
9.1 Nastavení . . . 69
9.2 Pouze heuristiky . . . 70
9.3 Náhodný výběr . . . 71
9.4 Populárnost . . . 73
9.5 Nedávno navštívená položka . . . 77
9.6 Iterátor . . . 78
9.7 Vyhodnocení experimentů . . . 81

Závěr **85**

Literatura **89**

A Seznam použitých zkratek **93**

B Obsah přiloženého flash disku **95**

Seznam obrázků

2.1 Hierarchie okének 23

4.1 Proces generovaných zpráv 39

5.1 Diagram třídy *Algorithm* 43
5.2 Diagram třídy *Metric* 44

6.1 Proces zpracování originálního datasetu 48
6.2 Počet vydaných zpráv za den 51
6.3 Počet transakcí za den 52
6.4 Návštěvnost nejnavštěvovanějších článků během hodiny 53
6.5 Počet článků zhlédnutých uživateli s nejvyšším počtem zhlédnutí článků během hodiny 54
6.6 Průměrný počet transakcí na hodinu 55

8.1 Provázanost tříd a interfaců jednoho z kandidátů 62

9.1 Vliv velikosti rezervoáru u heuristiky náhodného výběru 72
9.2 Vztah velikosti okénka a délkou překryvu v počtu transakcí u heuristiky populární položky 74
9.3 Vztah velikosti okénka a délkou překryvu v sekundách u heuristiky populární položky 75
9.4 Vztah velikosti cache paměti a dobou expirace záznamů u heuristiky populární položky 75
9.5 Vztah hloubky cache paměti a dobou expirace záznamů u heuristiky populární položky 75
9.6 Vztah velikosti okénka a délkou překryvu u heuristiky iterátor (velikost je dána v počtu transakcí) 79
9.7 Vztah hloubky cache paměti a dobou expirace záznamů u heuristiky iterátor 79

Seznam tabulek

6.1 Počet položek a transakcí pro jednotlivé vydavatelství 49
6.2 Přehled metadat transakcí 52

9.1 Naměřené metriky pouze heuristik 71
9.2 Vztah aspektů na měřené metriky u heuristiky náhodného výběru 73
9.3 Vztah aspektů na měřené metriky u heuristiky populární položky . 76
9.4 Vztah aspektů na měřené metriky u heuristiky nedávno navštívené položky 78
9.5 Vztah aspektů na měřené metriky u heuristiky iterátor 80
9.6 Souhrný přehled metrik vítězných kandidátů pro každou heuristiku za první týden 81
9.7 Souhrný přehled metrik vítězných kandidátů pro každou heuristiku za celý datový tok 81

Úvod

Množství obsahu na internetu roste každý den, stejně jako množství uživatelů, kteří mají k internetu přístup. V blízké budoucnosti přijde internet věcí, který tento trend ještě zintenzivní. Ve světle těchto skutečností zpracovávají doporučovací systémy velké množství dat a získávají z nich informace relevantní k doporučování. Vzhledem k množství dat není možné všechna data ukládat a opakovaně zpracovávat. Na základě vytěžených informací mohou systémy správně doporučovat položky, avšak musí vnímat jejich kontext a časovou závislost.

Doporučování novinových článků má svá specifika oproti jiným položkám, jako například knihy, filmy nebo zboží. Novinové články mají obvykle krátkou životnost, velký podíl anonymních uživatelů a nestrukturované informace. Vzhledem k množství článků a jejich krátké životnosti je tato doména vhodná pro dolování dat z datových toků.

Teoretická část si klade za úkol informovat o problematice doporučujících systémů, dolování dat a datových tocích. Kapitola týkající se doporučujících systémů popisuje stavební kameny a typy systémů. U doporučení jsou popsány sledované vlastnosti a vliv kontextu. V podkapitole testování jsou popsány metody a metriky, které se při vyhodnocování doporučujících systémů využívají. Kapitola dolování dat popisuje procesy, které jsou součástí získávání informací. V kontextu datových toků jsou zmíněny techniky, které se v různých variacích vyskytují u algoritmů pro dolování dat.

V rámci soutěže CLEF NewsREEL challenge vznikly vědecké články, které zkoumají různé přístupy k doporučování novinových článků. Z existujících zdrojů je provedena rešerše, která si klade za cíl shrnout jednotlivé přístupy a na jejich základě v implementační části navrhnout kandidáty. Tito kandidáti budou navrženi, aby bylo možné analyzovat vliv jednotlivých parametrů.

Výše zmíněná soutěž probíhala do roku 2017. Platforma, na které testy probíhaly, není od té doby udržována. Důležitým úkolem je zjistit, v jakém stavu se platforma nachází a zvolit alternativu, která bude pro testování

vhodná. Platforma musí mít možnost snadného implementování vlastních algoritmů. Výsledky kandidátů musí být porovnatelné s již existujícími algoritmy ze soutěže.

V části dolování dat budou data připraveny, aby mohly být využity platformou. Data budou dále analyzována, aby navržení kandidáti reflektovali použitá data. Úkolem této části je získat znalost o povaze dat. Data se mohou měnit vlivem času i okolních událostí. Zprávy mají krátký životní cyklus, který v této části bude popsán.

V implementační části budou popsány změny, které jsou v platformě provedeny. Dále je popsán proudový algoritmus, který bude základem všech testovaných kandidátů v experimentální části. Závěrem této kapitoly je implementace metrik, které budou k vyhodnocení kandidátů použity.

V experimentální části jsou na vybrané platformě a transformovaných datech analyzováni implementovaní kandidáti pomocí implementovaných metrik. Cílem této části je demonstrace vhodnosti technik dolování dat z datových toků v doporučovacích systémech. Dále je zde srovnání s tradičními přístupy. Závěrem práce je zhodnocení vhodnosti technik dolování dat z datových toků v doporučovacích systémech novinových článků.

Část I

Teoretická část

Kapitola 1

Doporučovací systémy

S rostoucím počtem možností výběru je schopnost uživatele se rozhodnout zhoršena. Uživateli trvá déle vybrat položku a zároveň je větší pravděpodobnost chybné volby. Tento jev popisuje Hicksův zákon [1]. Úlohou doporučujícího systému je zjednodušit proces výběru uživateli, aby při komplikovaném výběru z více možností, vybral správnou položku v relativně krátkém čase.

Doporučovací systém je softwarová komponenta, která se skládá z algoritmů a technik, které se využívají v dolování dat. Obvykle je obohacena uživatelským rozhraním, skrze které může uživatel ovlivnit doporučení. Doporučovací systémy pracují se třemi typy objektů: položka, uživatel a transakce [2][3]. Položkou je myšlena libovolná věc, při jejímž výběru je zapotřebí rozhodování a uživatel může potřebovat při výběru pomoc. Nabízené položky mohou být například:

- zboží;
- provedení akce;
- investiční plán.

Uživatel je klient, který s doporučujícím systémem interaguje. Doporučovací systém, na základě informací spojených s uživatelem, navrhne položky, které by pro něj mohly být vhodné.

Transakce je záznam o interakci mezi uživatelem a systémem. Jedná se o libovolnou událost, která je systémem sledovatelná a zvolená technika doporučujícího systému ji může využít ke zpřesnění doporučování. Příkladem transakce může být ohodnocení položky nebo zobrazení jejího popisu.

1.1 Principy

Doporučení položek lze předpovědět na základě již proběhlých transakcí. Systém si proto vytváří profily o uživatelích a položkách, které jsou identifikova-

telné unikátní hodnotou. Uživatelé následně hodnotí položky. Hodnocení může probíhat explicitně (například udělením počtu hvězdiček zhlédnutému filmu) a nebo implicitně (například zobrazením detailu položky). Čím více transakcí systém zpracovává o konkrétním uživateli nebo položce, tím kvalitnější je zaznamenaný profil.

Položky jsou ohodnoceny a na základě získaného skóre a formy doporučení jsou doporučeny uživateli. Složitost výpočtu skóre závisí na podstatě položky. Některé položky jsou doporučené pouze na základě hodnocení uživateli (například knihy) zatímco jiné jsou komplexnější na evaluaci (například vhodná nabídka práce). Uživatel na základě transakcí může položkám zvýšit nebo snížit skóre.

Zaznamenané informace v profilech uživatelů nejsou statické. Preference uživatelů se mohou časem vyvíjet nebo lišit v závislosti na situaci, ve které se uživatel nachází. Kontext dotazu může mít krátkodobý i dlouhodobý vliv na doporučení.

1.2 Kontext

Požadavek na doporučení s sebou nese informace (metadata), které mohou být systémem využity ke zpřesnění doporučení. Například uživatel hledající restauraci v době oběda má pravděpodobně jiné priority, když je v zaměstnání a když je na dovolené. Informace mohou být rozšířeny o dodatečné informace z profilu uživatele nebo z externích zdrojů. Příklad metadat je:

- čas;
- geolokace;
- identifikace zařízení.

Doporučování položek konkrétního typu je závislé na kontextu samotného doporučení. Stejný druh položky může mít pro stejného uživatele v různých kontextech jiné výsledky doporučení. Například doporučení knihy, kterou si chce uživatel koupit je pravděpodobně kniha, kterou si uživatel ještě nekoupil. Zatímco pokud tento uživatel bude chtít provést doporučení nějaké knihy známému, tak pravděpodobně doporučení bude obsahovat i knihy, které již sám přečetl.

1.3 Vlastnosti doporučování

Existují různé přístupy k doporučování, které mají své výhody a nevýhody. Jednotlivé přístupy se mohou prolínat, aby bylo možné dosáhnout ideálního výstupu pro daný případ. Mezi vlastnosti doporučování patří studený start (anglicky *cold start*), řídké matice (anglicky *matrix sparsity*) a subjektivita

hodnocení. Na konci kapitoly je zmíněno jak mohou být tyto vlastnosti použity k manipulaci doporučování.

Cold start nastane v okamžiku, kdy je přidán nový uživatel nebo položka do systému [2]. V tomto okamžiku nemá systém dostatek informací k tomu, aby mohl doporučení přizpůsobit. Existují různé techniky pro nové uživatele a položky, které tento problém řeší. V případě nově příchozího uživatele lze využít například následující techniky:

- uživatel sám definuje detaily svého profilu;
- použije se profil průměrného uživatele;
- využijí se generická doporučení;
- profil je definován uživateli, se kterými je uživatel propojen.

Pro nově příchozí položky se definuje vlastnost studenost (anglicky *coldness*), která určuje její novost v systému. Položka přestane být studená v okamžiku, kdy je v systému už dostatečně dlouho nebo dostala dostatek hodnocení [3].

Studené položky jsou pro uživatele často neznámé, a tak je jejich doporučení složitější. Doporučení studených položek může mít pozitivní dopad na aspekty novost a míra překvapení. Pro nově příchozí položky je možné využít následující techniky:

- manuální ohodnocení dedikovanými uživateli;
- odhadnutí hodnocení na základě podobnosti k existujícím položkám;
- odhadnutí hodnocení na základě analýzy atributů.

Matice hodnocení položek uživateli je řídká. Malá část uživatelů hodnotí více položek a zbylí uživatelé hodnotí jen málo položek. Distribuční funkce popisující počet uživatelů hodnotící určité množství položek má dlouhé chvosty. Uživatelé hodnotí většinou populární položky, které mají dostatek hodnocení, ale za to je jich relativně málo. Zbylých položek je množstevně více, ale mají výrazně nižší počet hodnocení na položku. Doporučování těchto položek zvyšuje aspekty jako novost a míru překvapení.

Samotné hodnocení uživateli je subjektivní a nemusí reflektovat skutečnost. Uživatel si například vybírá filmy tak, aby se mu líbily. Následně jen některé z nich skutečně ohodnotí. Filmy, které by se mu nelíbily si nevybere ke zhlédnutí, a tudíž je ani neohodnotí. Pokud zhlédne snímek mimo své preference, může snímek hodnotit negativně jen proto, že má uživatel jiný žebříček hodnot pro hodnocení filmů. Následně může vzniknout výrazný rozdíl v hodnocení položky uživateli v závislosti na jejich vkusu.

Populární položky dostávají lepší hodnocení, neboť jsou uživatelé ovlivněni kolektivním hodnocením. Hodnocení nereflektuje kvalitu položky, ale její obraz, který je popularností ovlivněn.

Posledním bodem jsou cílené útoky na hodnocení položky. Pomocí většího množství uživatelů je možné konkrétní položky hodnotit výrazně lépe nebo hůře. To má za následek, že jsou uživatelům doporučovány položky, které útočníci zvýhodnili, případně nejsou zobrazeny napadené položky [2].

Obranou proti takovým útokům může být ignorování hodnocení uživatelů s podezřelým chováním nebo ignorování nově příchozích uživatelů (cold start). Vzhledem k řídkosti matice může být relativně snadné ovlivnit hodnocení položek. Náhlé změny v hodnocení položek mohou být objeveny detektorem anomálií. V případě objevení anomálie je vždy důležité zhodnotit, zda-li se jedná o subjektivní hodnocení nebo cílený útok.

1.4 Forma doporučení

Forma výstupu závisí na využití doporučujícího systému. Výstup může být limitován počtem položek, strukturou výběru, množstvím zobrazených informací nebo možnostmi manipulace doporučovacích pravidel [3]. Pro některé využití nemusí být dostatečné vybrat pouze položky s nejvyšším skóre. Výstup může mít následující formy:

- **Nalezení dobrých položek** - doporučení obsahuje omezený počet položek na základě získaného skóre. Samotné skóre může, ale nemusí být zobrazeno. Tato varianta je vhodná pro větší spektrum nabízených položek, kde uživatel pravděpodobně hledá jen nějakou vhodnou položku.
- **Nalezení všech dobrých položek** - nalezení všech položek, které odpovídají alespoň nějakým požadavkům uživatele. Tato varianta je vhodná pro malé spektrum nabízených položek, kde uživatel potřebuje mít přehled o všech možnostech a případných alternativách. V případě porušení nějaké podmínky může systém zobrazit, jaké podmínky jsou porušeny, aby uživatel mohl lépe zhodnotit rizika jednotlivých kompromisů.
- **Pouze prohledávání** - uživatel nehledá konkrétní položku, ale pouze prohledává prostor položek. Doporučovací systém uživateli doporučuje položky, které jsou dostatečně rozprostřeny po prohledávaném prostoru a zároveň by mohly být pro uživatele zajímavé.

1.5 Typy doporučovacích systémů

Doporučovací systémy se liší v přístupu k doporučení. Některé typy se více soustředí na položky, jiné na uživatele a jiné zase na pravidla doporučení.

V praxi se tyto typy prolínají, aby bylo možné dosáhnout dostatečné kvality doporučení a vyhnout se tak nevýhodám, které jednotlivé přístupy mají [2].

1.5.1 Kolaborativní filtrování

Kolaborativní filtrování je implementace originální myšlenky doporučujících systémů, kdy uživatel dá na doporučení svých přátel. Přátelé mají obvykle větší množství podobných rysů. Pro vygenerování doporučení slouží okolí uživatele. Okolí tvoří uživatelé s podobným profilem. Z okolí jsou vybrány položky, které nejsou ohodnoceny uživatelem. Pro tyto položky je vypočteno skóre na základě podobnosti a hodnocení.

Doporučování je velmi závislé na korelaci mezi položkami a korelaci mezi uživateli. Většina uživatelů interaguje jen s malým množstvím položek. Z uvedeného vyplývá, že matice hodnocení jsou řídké. V případě příchodu nového uživatele nebo přidání nové položky je zde problém nízkého počtu hodnocení. Uživatel je podobný vícero různorodým skupinám. Položka nemá žádné hodnocení, tudíž ji nemá kdo doporučit.

Pro kolaborativní filtrování se používají dva základní typy metod: založené na paměti a založené na modelu. V metodách založených na paměti se používá algoritmů pro zjištění okolí uživatelů a položek. Typy založené na modelu využívají metod strojového učení a dolování dat k definování predikčního modelu. Tento model je následně naučen na testovacích datech pro konkrétní typ položek.

1.5.2 Filtrování založené na obsahu

Položky mají strukturované informace, které popisují jejich vlastnosti. Na základě podobnosti charakteristických rysů doporučovací systém vyhodnotí podobnost položek. Důležité pro fungování této metody je způsob extrakce a následné porovnání charakteristických rysů.

Nově příchozí uživatel nemá žádnou historii o položkách, a tak doporučovací systém neví, jaké položky doporučit. Po provedení pár transakcí nebo doplnění profilu je možné zjistit, jaké položky doporučit. Při vložení nové položky do systému není problém s jejím ohodnocením, neboť informace o položce mohou být z popisu položky extrahovány (například žánr filmu). Doporučení nově příchozí položky se může opřít o ohodnocení podobných položek, které již uživatelem byly ohodnoceny.

Nabízené položky trpí nízkou diverzitou, protože jsou založeny na podobnosti s již ohodnocenými položkami. Aby bylo možné tyto problémy vyřešit, je možné využít technik založených na znalosti domény.

1.5.3 Demografický typ

Uživatelé systému mohou spadat do různých demografických skupin. Pro každou skupinu může být navržen mírně jiný přístup v doporučování položek.

K identifikaci skupiny je zapotřebí zjistit informace o uživateli. Ty je možné získat z profilu uživatele, z jeho chování nebo kontextu dotazu.

Informace o uživateli mohou být neúplné a mohou se časem měnit. Algoritmus výběru vhodného modelu musí být dostatečně robustní, aby vybral správný model i pro neúplnou informaci o uživateli. Z tohoto důvodu se obvykle tento typ kombinuje s dalšími typy doporučovacích systémů.

Čím má model více dat, tím lépe může predikovat doporučení. Rozdělení uživatelů do demografických skupin může vést ke zvýšení přesnosti doporučování, neboť daná demografická skupina je více homogenní. Na druhou stranu se snižuje objem dat, který může být pro model použit. Z toho vyplývá, že rozdělení dat může mít pozitivní efekt, pokud nedojde k nedostatku dat pro určitou skupinu.

1.5.4 Filtrování založené na znalostech

Doporučovací systémy tohoto typu jsou přizpůsobeny konkrétní doméně. Důvodem může být nedostatečné množství hodnocení položek nebo komplexnost jejich výběru. Například výběr automobilu v základní výbavě má mít nižší skóre než automobil v plné výbavě. I přesto může uživatel tuto variantu preferovat, neboť je levnější.

Na základě znalosti domény je možné připravit pravidla k filtrování položek dle atributů. Z položek by mělo být možné extrahovat atributy. V tomto ohledu je přístup podobný filtrování založenému na obsahu.

Vzhledem ke komplikovanosti výběru a množství variant může být uživateli dáno rozhraní, skrze které může ovlivnit doporučení. Řízení doporučení může probíhat prostřednictvím omezení, kotev nebo pomocí kritiky:

- **Omezení** - uživatel může definovat omezení položek a jejich řazení. V případě, že nabízené položky neodpovídají potřebám uživatele, má uživatel sám možnost se rozhodnout, jaké podmínky zmírní.

- **Kotvy** - uživatel zadává výchozí položku, která by mohla splňovat jeho potřeby. Na základě tzv. kotvy jsou vybrány podobné položky, ze kterých si uživatel může vybrat. Opakováním tohoto procesu může uživatel nalézt vhodnou položku.

- **Kritiky** - jedná se o kombinaci dvou předchozích přístupů. Systém doporučí uživateli položku, kterou může následně uživatel kritizovat. Na základě doporučené položky může uživatel lépe specifikovat požadavky. To znamená, že uživatel přidává nebo mění omezení. Iterativně tak jsou uživateli nabízené různé položky, které vyhovují jeho kritériím výběru.

1.5.5 Hybridní typ

Jednotlivé typy systémů mají různé nevýhody. Proto se techniky kombinují a vznikají nové hybridní typy. Výběr modelu může být závislý na kontextu nebo množství nasbíraných informací. Zatímco pro běžné doporučování může být využit jeden přístup, pro nové položky a uživatele je vhodnější využít alternativní přístup. Na základě přesnosti a důvěry může být vybrán vhodný model k doporučení. Hybridním typům, které obsahují více doporučovacích modelů se říká anglicky *ensemble* model. V případě doporučení více položek mohou různé modely vytvořit mix položek.

1.5.6 Doporučovací systémy s kontextem

Kontext dotazu během doporučení hraje důležitou roli. Přidání kontextu do žádosti na doporučení může výrazně zvýšit přesnost doporučení. Jako kontext je možné považovat například čas a lokaci [2].

Závislost na čase může být kontinuální nebo dočasná. Kontinuální hodnota položky se mění jejím stářím. Například doporučovací hodnota filmu je rozdílná v den premiéry a po deseti letech. Parametr času je součástí dotazu pro doporučení. V případě kolaborativního filtrování podobnost respektuje čas hodnocení.

Dočasná hodnota doporučení se může měnit podle hodiny, dne, týdne, měsíce nebo roku. Příkladem je sezónní oblečení, které je vhodné doporučit jen pár měsíců v roce. V případě rozdělení doporučujících modelů dle časového rozsahu dojde ke zvýraznění problému řídkosti matice hodnocení.

Polohu je možné rozdělit do dvou kategorii: preference lokality a cestovní lokality. V prvním případě se jedná o lokalitu uživatele, se kterou se uživatel ztotožňuje a tedy sdílí zvyky uživatelů z dané lokality. Druhá varianta je vzdálenost položky od současné pozice uživatele. Vzdálenost restaurace má například velký význam při výběru odpoledního obědu. Pro určování vzdálenosti se používají ad-hoc heuristiky, které mohou případně penalizovat vzdálené položky.

Doporučovací systémy skládající se z více modelů (hybridní typy) používají heuristiku k výběru vhodného modelu. Kontext a metadata mohou být jedním z parametrů, které jsou zohledněny při výběru modelu.

1.6 Testování

Doporučovací systém je možné konfigurovat parametry, které mají vliv na kvalitu doporučování. Před nasazením do reálného provozu by měl být vliv těchto parametrů vyhodnocen. K testování kvality existují metody, které se liší v náročnosti na provedení a množstvím informací, které mohou získat [2]. Během testování by měly být měřeny vlastnosti, které jsou pro výsledné hodnocení

důležité. Vyhodnocení může proběhnout ve třech formách: online, user-study a offline.

V případě online verze je finální systém testován na cílových uživatelích. Ti jsou rozděleni do dvou skupin, kde jedna skupina používá stávající model a druhá skupina používá nový model. Uživatelé nevědí o testování, a tak je výpovědní hodnota podobná ostrému provozu. Po ukončení testovacího období mohou být provedena porovnání na naměřených aspektech. Princip je založeno na A/B testování, případně obecnějším multinomiálním testování.

Běžně je vybráno více kandidátů k testování. Aby mohlo testování probíhat paralelně využívá se zobecněný algoritmus zvaný multi-arm bandit. Algoritmus optimalizuje poměr doporučování výchozího modelu a nových kandidátů. Každý nový kandidát má stejnou pravděpodobnost výběru. Jestliže kandidát doporučí správné položky, je jeho pravděpodobnost budoucího vybrání zvýšena. V opačném případě je pravděpodobnost snížena. Tímto způsobem jsou postupně zvýhodňováni kandidáti, kteří vracejí dobré doporučení na úkor těch nepřesných.

User-studies jsou založené na reprezentativním vzorku uživatelů, kteří jsou testování na izolované verzi doporučovacího systému. Před a po dokončení testování jsou uživatelé dotázání na doplňující informace prostřednictvím dotazníku. Dotazníky umožňují získat aspekty doporučení, které nejsou jinak měřitelné (například míra důvěry v doporučení). Uživatelé vědí o testování, což vede k jejich nepřirozenému chování během výběru doporučení. Výpovědní hodnota dat je tím snížena.

Offline metoda je nejčastější varianta testování. Testování probíhá bez uživatelů. Interakce uživatelů se systémem je simulována zaznamenanými interakcemi. Různé doporučovací modely mohou být otestovány na stejných vstupních datech.

Tento způsob testování může naměřit jen omezené množství aspektů doporučení. Nejčastěji měřeným aspektem je přesnost doporučení. Dosažená hodnota však nemusí odpovídat realitě. Pokud uživatel dostane na výběr jiné položky, než které jsou zaznamenány v datech, je doporučení vyhodnoceno jako neúspěšné. Uživatel mohl mít v té chvíli zájem o danou položku, jen mu během testování nebyla navržena a on na ni nemohl kliknout. Uvedená skutečnost vede k false negatives a rozdílným výsledkům během offline a online testování.

1.7 Metriky

Při doporučování je možné měřit různé aspekty doporučení [2]. Před samotným měřením je důležité definovat, které aspekty jsou pro doporučovací systém klíčové. Výběr může ovlivnit typ položek a kontext doporučení. Volba aspektů závisí na využití systému a možnostech testovací metody. V případě doporu-

čování více položek je možné vybrat mix aspektů, které mají nabízené položky splňovat.

Přesnost predikce je důležitá, avšak není dostatečná pro dobrý doporučovací systém. Při vytváření vhodného mixu aspektů doporučení je důležité mít na mysli, že vlastnosti jsou vzájemně propojené, a tak změna jedné vlastnosti může ovlivnit všechny ostatní. Například zvětšením počtu nových položek se zvyšuje míra překvapení, ale zároveň se snižuje přesnost doporučení. Naopak při preferenci přesných doporučení dochází ke zmenšení celkového počtu doporučovaných položek. Vzhledem k omezenému objevování nových položek může být pro uživatele doporučení dlouhodobě nezajímavé, i když ze začátku bylo velmi přesné.

1.7.1 Přesnost

Přesnost predikce může mít například následující interpretace:

- procento doporučení, kdy nějaká položka byla vybrána;
- rozdíl v pořadí doporučených položek;
- rozdíl v pořadí prvních K položek;
- chyba při predikci hodnocení/skóre.

Interpretace přesnosti je zvolena dle kontextu doporučování. V případě predikce skóre nebo hodnocení je považováno za chybu rozdíl mezi predikcí a skutečnou hodnotou. Nejčastější metriky pro počítání chyby jsou RMSE (1.1) a MAE (1.2). První varianta penalizuje více větší chyby, zatímco druhá nedělá rozdíly ve velikosti chyby.

$$RMSE = \sqrt{\frac{1}{count}\Sigma_{i=1}^{count}\Big(true_i - predicted_i\Big)^2} \tag{1.1}$$

$$MAE = \frac{\Sigma_{i=1}^{count}|true_i - predicted_i|}{count} \tag{1.2}$$

V případě pořadí je možné počítat celkový počet změn ve vzorovém a predikovaném pořadí. Pokud je z doporučení vybrána jen 1 položka, tak může být měřena přesnost (anglicky *accuracy*) nebo MRR (1.3). Přesnost bere poměr mezi doporučením správné položky a doporučení všech. MRR bere v potaz pořadí doporučené položky. Čím výše je v doporučení položka, která byla doopravdy vybrána, tím vyšší je výsledné skóre. Formální reprezentace výpočtu je:

$$MMR = \frac{1}{count}\Sigma_{i=1}^{count}\frac{1}{rank_i} \tag{1.3}$$

Kde *count* je množina všech dotazů a $rank_i$ je pořadí doporučení v množině všech doporučených. MRR může dosáhnout nejlépe stejných výsledků jako přesnost.

Během online testování se obvykle využívá metrika clickthrough rate (1.4), která reprezentuje poměr kliknutí na doporučené položky ku všem zobrazením doporučení (anglicky *impression*). V ideálním případě uživatel vždy klikne na doporučení. Vysoké hodnoty metriky odpovídají kvalitnímu mixu doporučení. Uživatele zaujme nabídka a klikne na další položku [4].

$$CTR = \frac{(number\ of\ click\text{-}throughs)}{(number\ of\ impressions)} \tag{1.4}$$

1.7.2 Pokrytí

Tento aspekt definuje dvě roviny: pokrytí uživatelů a pokrytí položek. V případě uživatelů se jedná o poměr uživatelů, pro které je systém schopen doporučit vhodné položky. Systém může mít problém s doporučením položek nově příchozím uživatelům nebo uživatelům, kteří nemají žádné podobné uživatele. Systémy obvykle mají 100% pokrytí uživatelů, neboť pro každého uživatele jsou doporučeny populární položky, pokud neexistuje specifičtější doporučení.

Pokrytí uživatelů definuje poměr uživatelů, pro které je možné predikovat pevně dané množství položek [2]. Pro uživatele s malým množstvím hodnocení může být obtížné najít podobné uživatele nebo dostatek položek, aby mohly být nějaké doporučeny. Tato hodnota může být definována jako počet položek, které musí být ohodnoceny, než je možné uživateli nabídnout alespoň daný počet položek. Velikost limitu závisí na podstatě doporučovaných položek a kontextu doporučení.

V případě položek se jedná o poměr položek, který může být někomu doporučen (tzv. katalogové pokrytí). Tato hodnota reflektuje, kolik různých položek může být teoreticky zobrazeno v doporučení nějakému uživateli. Vysoké pokrytí ukazuje vysokou míru využitelnosti systému.

1.7.3 Důvěra a spolehlivost

Systém může podpořit své doporučení pravděpodobností, jak moc si je doporučením položky jistý. S rostoucím množství informací o uživateli a položkách roste důvěra ve správné doporučení. Nedostatek dat naopak vede k nízké důvěře. Hybridní modely mohou využít hodnoty důvěry v heuristice výběru aktivního modelu. Takovéto systémy preferují populární položky a mají negativní dopad na aspekty novost a míra překvapení.

Při testování s uživateli lze měřit míru spolehlivosti uživatele v doporučení, které systém nabídl. Aby se zvýšila spolehlivost v doporučení systému musí systém reflektovat transakce provedené uživatelem. To znamená, že systém

omezuje vhodným způsobem prohledávaný prostor. To je však v rozporu s novostí a překvapením.

1.7.4 Novost

Položky jsou pro uživatele nové do doby, dokud se s nimi neseznámí. Seznámení může proběhnout prostřednictvím doporučení, přečtení detailu položky nebo mimo systém. Novost položky nevyplývá ze skutečnosti, že systém nezaznamenal interakci s položkou. Obecně lze předpokládat, že populární položky jsou s velkou pravděpodobností pro uživatele známy [2].

Novost lze změřit offline testováním, pokud je znám čas interakcí. Nejprve je náhodně vybrán čas, kdy bude predikce doporučení probíhat. Následně jsou odebrány všechny hodnocení provedené po tomto okamžiku. Tyto položky jsou považovány za nové. Poté je odebráno pár hodnocení před tímto okamžikem. Tyto hodnocené položky reprezentují objevení položky mimo svět systému a nejedná se o nové položky. Následně je proveden požadavek na doporučení.

Za položky doporučené z doby před okamžikem doporučení jsou uděleny záporné body, neboť uživatel již zná tyto položky. Položky doporučené po tomto okamžiku znamenají pozitivní body, neboť položka byla pro uživatele nová a bude ohodnocena. Odebrání položek před okamžikem může mít vliv na doporučení položek. Proto musí být odebraná hodnocení pečlivě vybírána.

1.7.5 Míra překvapení

Popisuje subjektivní pocit uživatele, jak moc uživatel neočekával doporučenou položku, která byla i tak pro uživatele zajímavá. Položka je pro uživatele nová a není podobná položkám, se kterými uživatel doposud interagoval. Prvek překvapení podporuje prozkoumávání prostoru položek. Tuto hodnotu lze zjistit pomocí user-study.

1.7.6 Diverzita

Z doporučených položek lze spočítat průměrnou vzájemnou podobnost položek. Čím nižší je tato hodnota, tím větší je diverzita seznamu. Nabízené položky by měly být rozdílné, aby z nezájmu o jednu položku nevyplýval nezájem o ostatní doporučené položky. Výší diverzita zvyšuje katalogové pokrytí a má pozitivní vliv na doporučení nových a překvapujících položek na úkor přesnosti.

1.7.7 Doba odezvy

Poslední metrikou je délka odezvy na doporučení. Jedná se o časový úsek od přijetí požadavku až po odeslání odpovědi. Očekávaná doba záleží na kontextu dotazu. Pokud se uživatel dotazuje na doporučení dopravy do zvolené

destinace, očekává odpověď v řádu stovek milisekund. Pokud systém doporučuje ideální strategii pro rozvoj firmy je uživatel ochotný čekat i několik dní. V závislosti na zvolené doméně je stanoven limit, který by měl systém splňovat.

Kapitola 2

Vytěžování dat

Vytěžování dat je věda, která se zabývá získáním užitečných znalostí z dat. Základními problémy, kterými se věda zabývá, jsou: shlukování, klasifikace, vytěžování asociačních vzorců a analýza odlehlých hodnot [5]. Vzhledem k častému výskytu těchto problémů v různých podobách má tato věda široké spektrum nástrojů, jak tato specifika řešit. Vytěžování dat může být aplikováno na různé domény jako jsou například multidimenzionální data, text, obraz, časové řady nebo grafy. Vzhledem ke zvyšování množství generovaných dat dochází k rozšíření této vědy do sfér jako jsou datové toky, weby a sociální sítě.

Proces získání informací z dat obvykle začíná shromážděním dat z jednoho nebo více zdrojů. Přijatá data jsou zpracována a jsou z nich extrahovány charakteristické rysy. Data jsou následně vyčištěna. Očištěná data mohou být pomocí nástrojů dále zpracovávána a analyzována.

Extrakce charakteristických rysů (anglicky *feature extraction*) ze zdrojů je nutná, když příchozí data nemají jednotnou strukturu. Kromě uvedení dat do jednotné formy jsou data spojena. Rysy, které nejsou důležité, jsou zapomenuty. Jednotlivé rysy mohou být typu kvantitativního, kategorického, textového nebo grafového. Výsledný formát může být rozšířen o rysy z externích zdrojů.

Čištění dat je proces, který připraví data do stavu, aby mohla být dále analyzována algoritmy vytěžování dat. Data mohou mít různý formát a jednotky. Některé informace mohou chybět nebo mohou být redundantní. Některé hodnoty mohou být nepřesné. V případě dostupnosti stejné informace z více zdrojů, mohou být hodnoty porovnány. Heuristika může následně rozhodnout, jaká hodnota je vybrána. Aby byly údaje správně opraveny, je zapotřebí jim porozumět. Data musejí být použitelné v datové analýze a zároveň nesmí být příliš změněn jejich význam. Výsledkem čištění dat je unifikovaný formát, který je vhodný pro další analýzu. Upravená data jsou obvykle ukládána, aby při opakované analýze dat nemuselo znovu docházet k jejich úpravě.

Na závěr dojde ke zpracování dat. Za použití vhodných algoritmů jsou

získány cenné informace. Komplikací datové analýzy je široké spektrum aplikací, ve kterých jsou data použita. Různé aplikace vyžadují specifické množiny nástrojů a přístupů. K vytěžení informací může být zapotřebí zřetězení vícero nástrojů, které dolují z dat informace.

2.1 Datový tok

Datový tok je kanál, kterým putují informace od zdroje k cíli. Informace jsou zdrojem vyslány jen jednou. Informace jsou poté přijaty v pořadí, v jakém byly vyslány. Rychlost generování dat není možné ovlivnit. Datový tok je relativně nekonečný [6].

Se zvyšující se silou výpočetní techniky je možné generovat a zpracovávat větší množství dat s více detaily. Ze zpracovaných dat je možné získat další znalosti. Spektrum informací, které mohou být zpracovány a uloženy je obrovské. Zdrojem datového toku mohou být například uživatelé procházející internetové stránky (anglicky *clickstream*), sensory zaznamenávající teplotu nebo datový provoz v síti.

Zpracovávání datových toků je v praxi čím dál tím častější. Přímé zpracování datového toku zvyšuje propojení systémů a zkracuje dobu propagace změn. Systémy musí umět porozumět příchozím informacím a umět se správně adaptovat na změny. Přechodem od statických dat k datovým tokům dochází k přidání další dimenze do evaluace - času. Čas může být chápán ve formě zveřejnění zprávy nebo času zpracování. Důležitost zpráv se během času může měnit.

Algoritmy pracující nad datovými toky se nazývají online algoritmy. Jejich specifikem je, že na každý příchozí požadavek vrátí odpověď hned. Neberou v potaz budoucí požadavky. Offline algoritmy nejprve přijmou všechny požadavky, a až poté na ně vrátí odpovědi. K odpovědi tedy využívá znalosti všech požadavků. Online algoritmus může dosáhnout nejlépe takového výsledku, jakého dosáhne offline verze.

2.2 Vytěžování datových toků v doporučovacích systémech

Doporučovací systémy používají algoritmy na vytěžování dat. Tyto algoritmy jsou použity při získání informací o dostupných datech. Na základě získaných informací jsou optimalizována doporučení pro jednotlivé uživatele. Jedná se obvykle o algoritmy, které předpokládají přístup ke všem informacím a vícenásobný průchod daty.

S rostoucím tempem generování transakcí se stává nemožné mít všechny informace dostupné. Offline algoritmy limitují výkon doporučujících systémů a způsobují zpoždění. To není přijatelné pro praktické využití doporučovacích systémů.

K vytěžování informací se čím dál častěji využívají online algoritmy. Vzhledem k množství zpracovávaných informací pracují systémy s datovými toky. Online algoritmy pro vytěžování dat nad datovými toky vracejí dostatečně aproximované výsledky bez výrazné prodlevy odpovědi. Dostupnost všech dat je pro real-time odezvu nemožné splnit.

Uživatel může klást dotazy na doporučovací systém a systém musí odpovědět s minimální odezvou. Každá příchozí informace je zpracována pouze jednou. Cíl musí informace zpracovat nebo uložit, jinak budou ztraceny. Vzhledem k tempu a množství generovaných dat není možné uložit všechna příchozí data. Aby bylo možné rychle odpovědět na dotaz musí být data uložena v paměti, která má omezenou kapacitu. Data mohou být uložena v databázi. Zpoždění způsobené komunikací s databází musí být minimální.

Pokud je distribuce dat v toku stabilní, tak se jedná o podobný problém jako dolování dat z velkého statického data setu. Data není možné najednou nahrát do paměti. Streamováním je možné vytvořit dostatečně reprezentativní vzorek celého toku/data setu.

2.3 Zpracování datového toku

Délka zpracování zprávy by se neměla prodlužovat s délkou datového toku. Pokud trénování modelu trvá déle, není možné ho provádět po každé příchozí položce. Proto existují dvou průchodové algoritmy, které v první fázi (zvané online) sbírají data a třídí je do skupin. Tyto skupiny jsou jednou za čas zpracovávány paralelně a vytvářejí budoucí predikční model (offline část).

Vzhledem k omezeným paměťovým zdrojům platí podobná limitace i pro uložené informace. Paměťová složitost může s délkou toku mírně narůstat, avšak dosažení kritické hranice by mělo být v dostatečně časově vzdálené době, ideálně nikdy. Zpracování datových toků přichází s novými výzvami. Konvenční algoritmy většinou nesplňují požadavky kladené na zpracování datových toků. Mnoho konvenčních metod předpokládá, že je možné všechna data nahrát do paměti a zpracovat je nebo k nim přistoupit vícekrát.

Aby bylo možné zároveň splnit požadavky na paměť, rychlost zpracování příchozích dat a rychlost odpovědi, musí dojít ke snížení komplexnosti algoritmů [7]. Paměť může pojmout jen tolik informací, aby systém mohl odpovědět dostatečně aproximovanou odpovědí. Aproximovaná odpověď znamená, že odpověď může mít chybu nejvýše ϵ. Čím větší je tolerovaná chyba, tím méně paměti je zapotřebí a tím rychlejší může být odpověď.

2.4 Vývoj datového toku

Pokud informace v datovém toku nejsou stabilní, tak se během času mění. Důvodem změny může být prostředí nebo uživatelé. Změna může být postupná, náhlá, opakující se nebo přechodná. Detekci změny komplikuje šum,

který je v datech přítomný. Detekční mechanismy obvykle balancují mezi počtem falešných poplachů a nedetekovaných změn.

Modely jsou naučené a platné ve své časové lokalitě (anglicky *temporal locality*). V případě změny může dojít ke zhoršení kvality výsledků. Model by měl obsahovat adaptační metody, aby mohl změnu detekovat a v případě potřeby se přizpůsobil. Některé změny nevedou ke zhoršení kvality výstupu, a tak nemusí být adaptovány [8].

Prvním druhem změny je vliv vlastností vstupu na výsledek modelu. Některé vlastnosti mohou získat nebo ztratit význam během času. To může například znamenat, že model je naučený rozpoznávat cenu mobilního telefonu podle kapacity baterie. Postupem času však uživatelé začali vybírat telefony podle výrobce. Došlo tedy ke snížení důležitosti jedné vlastnosti a ke zvýšení důležitosti vlastnosti druhé. Tato změna se nazývá posun vlastnosti (anglicky *feature drift*).

Dalším druhem je změna samotného konceptu. Vstupní data zůstávají stejná, avšak jejich výsledek se změnil. Model takovou změnu v době učení nepředpokládal. Může se například jednat o sezónní zboží. Model se naučil preference uživatelů při nákupu oblečení uživatelů v zimním období. Zákazník však v létě očekává letní oblečení. Tuto variantu model při učení neočekával. Tomuto jevu se říká reálný posun konceptu (anglicky *real concept drift*). Virtuální posun konceptu nastane v situaci, kdy model správně predikuje výsledky na základě původně naučených dat, ale distribuce vzorů se změnila [7]. Tento jev popisuje například situaci, kdy do obchodu začnou nově chodit turisté, kteří však kupují stejné zboží, jako místní.

Čím výraznější je změna, tím lépe ji lze detekovat a zároveň je zapotřebí méně vzorků k jejímu odhalení. Data jsou mírně zašuměná a model by měl rozpoznat, kdy se jedná jen o anomálii, a kdy o začátek posunu. Například zákazník si může v zimních měsících koupit kraťasy, ale hlavní sezóna přijde až o pár měsíců déle. Model musí správně vyhodnotit, že se nejedná o začátek posunu.

Další jev, se kterým se může model potýkat, je změna distribucí výsledků. Distribuce výsledků může změnit svou formu, může zmizet nebo naopak nově vzniknout. Nové položky mohou být detekovány jako výrazně odlišní jedinci, kteří nejsou změnou konceptu. Model se může aktualizovat nebo nahrazovat. Pokud je více modelů, je možné vybírat nejvhodnější model pro dané zadání.

2.4.1 Detekce změny

Během zpracování datového toku může dojít ke změně. K odhalení změny existují detekční metody. Při vyhodnocení metody detekce změny je důležité měřit následující položky: počet odhalených změn, počet falešných poplachů, doba odhalení změny. Po odhalení změny musí dojít k adaptaci učícího algoritmu. K tomu slouží adaptační metody. Aby bylo možné algoritmus správně nastavit, musí si algoritmus vést statistiky o stavu datového toku [7].

Změna může být způsobena skrytými proměnnými, které nejsou systémem sledovány. Detekční metody sledují agregační ukazatele příchozích dat. Příkladem ukazatelů mohou být velikosti chyb, průměry hodnot, variance, vychýlené korelace nebo střední odchylky. Po dosažení určitého času nebo objemu dat se z ukazatelů vytváří statistiky, které se ukládají. Z existujících statistik mohou vzniknout agregované statistiky popisující delší časové období. Tímto vznikne schéma okének popisující různá časová období s různou granularitou statistik.

Na základě uložených statistik je možné vyhodnotit trendy a chování distribuce dat. Koncepty předpokládají, že nová data jsou relevantní a porovnávají je se staršími informacemi. Koncept je považován za perzistentní, pokud je konzistentní alespoň polovinu velikosti okénka. Detekce změny v datovém toku může být provedena algoritmem CUSUM nebo Page-Hinkley testem [7]. Parametry algoritmů jsou použity k balancování citlivosti mezi detekcí změny a planými poplachy.

V okénkách je sledována pravděpodobnost chyby a její standardní deviace. Pokud se hodnoty snižují, aktualizují se minimální hodnoty. V případě, kdy hodnoty překročí mez o danou hranici, dojde k varování. Pokud i další prvek překoná hranici, dojde k driftu. Hranice mezi varováním a driftem slouží k odladění velikosti okénka, přes které se pravděpodobnost chyby a střední odchylky vypočítává. Doba mezi varováním a driftem odpovídá rychlosti změn.

Změna není považována za šum. Změna znamená vývoj distribuce, která je podpořená daty v datovém toku a ta se po určitou dobu v toku objevuje. Šum je nahodilý a nestane se součástí toku. Algoritmy obvykle musí najít správný balanc mezi odhalením šumu a změny.

2.4.2 Adaptační metody a správa statistiky

Adaptační metody nad datovým tokem mohou být slepé nebo informované. V případě slepé adaptace dochází k adaptaci učících algoritmů v pravidelných intervalech. Model je adaptován bez ohledu na to, jestli v toku došlo ke změně či nikoliv. Příkladem slepé metody jsou okénka fixní velikosti nebo přiřazení váhy starším statistikám. Informované metody adaptují učící algoritmus pouze v případě, že detekční model odhalil změnu.

K vyhodnocení změny musí být statistiky ukládány. Paměť statistik může být úplná nebo částečná[7]. Úplná paměť přiřazuje jednotlivým statistikám váhu pomocí funkcí. Tato funkce je obvykle klesající, kdy je větší důraz kladen na novější statistiky. Obvykle se jedná o lineární nebo exponenciální funkci. Statistiky, které nemají žádnou váhu mohou být zapomenuty.

V případě částečné paměti se používá okénko k zapamatování statistik. Model si pamatuje jen posledních pár statistik. V případě příchozí nové statistiky, kdy je kapacita okénka naplněna, je nejstarší statistika zapomenuta. Velikost okénka ovlivňuje adaptabilitu a přesnost. Malá okénka zvyšují adaptabilitu, ale zhoršují přesnost. S větším okénkem je tomu naopak. Velikost

okénka může být fixní nebo adaptivní. Adaptivní okénka reagují na statistiky příchozích dat a podle toho ukončují okénko.

2.5 Techniky

Algoritmy nemohou pracovat s celým obsahem datového toku, kvůli jeho rychlosti a relativní nekonečnosti. Proto využívají zjednodušené modely, které dostatečně reprezentuje datový tok. Modely jsou navrženy tak, aby vhodně odpověděly na specifický druh otázek. Důsledkem je, že daný model může být nevhodný pro odpovědi jiného druhu.

Techniky často pracují s pojmy aproximace nebo randomizace. Aproximační algoritmus vrátí výsledek, jehož chyba je menší nebo rovna dané hranici. Randomizovaný algoritmus vrátí výsledek, jehož pravděpodobnost chyby je dána parametrem. Běžnými technikami pro vytvoření modelu jsou: vzorkování (anglicky *sampling*), náčrt (anglicky *sketch*), hashování, okénko, vlnka a histogram.

2.5.1 Hashování

Hashovací funkce je funkce, která převede jakoukoliv číselnou hodnotu na číselnou hodnotu z definovaného rozmezí. Vypočítat hodnotu není náročné na CPU ani na paměť. Funkce je deterministická a rozdělí položky do intervalu rovnoměrně. Použitím hashovací funkce se sníží velikost dimenze za cenu konfliktů. Ty mohou, ale nemusí být řešeny.

Vstupem funkce musí být číselná hodnota. Řetězce nebo objekty musí být nejprve převedeny na číselné hodnoty. V případě řetězců je možné převést každý znak na číselnou hodnotu. Pole čísel lze následně převést postupným hashováním na jednu hodnotu. V případě objektů se jednotlivé atributy převedou na číselné hodnoty, které se následně taktéž postupným hashováním převedou na jednu hodnotu.

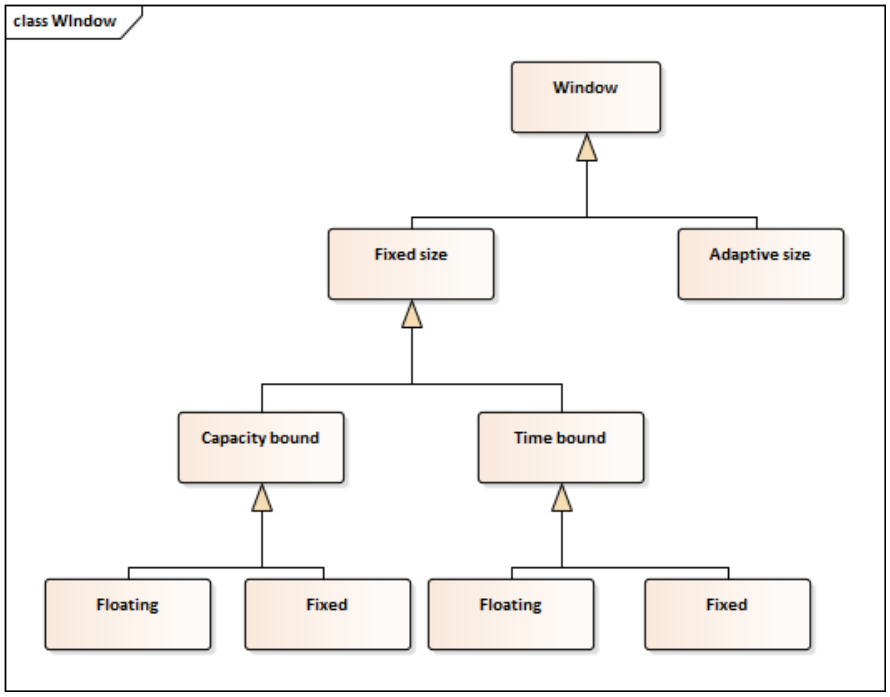
Hashování je vhodné pro výběr reprezentativního vzorku dat nebo rovnoměrného rozdělení příchozích dat do skupin. Algoritmy mohou následně paralelně zpracovávat jednotlivé skupiny a agregovat výsledky z každé skupiny. Hashování se využívá například v algoritmech LogLog, HyperLogLog nebo HyperLogLog++.

Na základě předpokladu, že hashovací funkce rozděluje výsledky do rozmezí rovnoměrně, je možné snížit počet položek, které jsou zpracovány nebo ukládány. Příchozí transakce mohou být hashovány a následně dle prefixu filtrovány. Tímto se sníží celkový počet zpracovaných položek, ale i tak může být zjištěn například aproximovaný počet unikátních prvků v toku.

2.5.2 Okénko

Datový tok je možné zpracovávat v intervalech tzv. okénkách [7]. Velikost okénka může být fixní nebo adaptivní. Fixní velikost se dělí na omezenou časem a nebo kapacitou. Obě tyto kategorie mohou být fixní nebo posuvné. Hierarchie okének je vizualizována v obrázku 2.1. Volba typu okénka záleží na využití jejich dat.

Obrázek 2.1: Hierarchie okének



Velikost okénka ovlivňuje přesnost a adaptabilitu modelu. Čím je okénko větší, tím se zvyšuje přesnost a snižuje adaptabilita na změny. Menší velikost okénka má nižší přesnost avšak rychlejší adaptabilitu. Velikost okénka může být adaptivní v závislosti na vlastnostech toku. Pokud je tok dostatečně stabilní, není zapotřebí vytvářet nové okénko, ale stačí zvýšit jeho kapacitu.

Po naplnění okénka je možné jeho data reprezentovat náčrtem. Datový tok je reprezentován skupinou okének, které mohou popisovat různá časová období. Obvykle jsou tato období větší, čím jsou události starší. Na základě porovnání okének je možné zjistit například vývoj dat, změny distribucí nebo anomálie. Pokud okénka popisují omezené časové období, fungují jako nástroje pro zapomínání.

2.5.3 Časové modely okének

Čas je důležitým aspektem v datovém toku [6]. Přicházející data ztrácejí svoji výpovědní hodnotu. Reprezentaci datového toku okénky popisují časové modely. Mezi základní typy patří: přirozený, logaritmický a progresivně logaritmický.

Přirozený model reflektuje rozdělení času na sekundy, minuty, hodiny, dny, měsíce a roky. V případě potřeby mohou být tyto jednotky rozděleny na zlomky. Tento model umožňuje volbu jednotek a velikostí dle požadavků domény. Tím pádem se jedná o ideální poměr mezi množstvím detailů a paměťovou náročností. Pro pochopení jsou tyto velikosti srozumitelné a představitelné.

Logaritmický model pracuje s exponenciálním růstem velikosti rozsahu okénka. Nejprve se zvolí velikost nejmenšího okénka a exponent pro rychlost růstu velikosti okénka. V případě zvolení počáteční velikosti jedna hodina a exponentu 2, mají následující okénka velikosti 2, 4, 8 a 16 hodin. V případě příchodu nového časového okénka dojde k posunování a slučování existujících okének.

Mohou vzniknout dočasná okénka, která čekají na naplnění kapacity, aby mohla být sloučena a použita jako okénko vyššího exponentu. Pro každou úroveň může existovat několik dočasných okének. Jejich počet závisí na kapacitě okénka vyššího exponentu. V případě posunu se z okénka vyšší úrovně stane nové dočasné okénko. Nevýhodou logaritmických okének je příliš velký skok v granularitě.

Progresivní logaritmický model si pamatuje položky posledních n položek s danou frekvencí. Model je definován řádem a kapacitou. Příchozí informace je uložena do daného okénka, jestliže platí tento vztah (2.1). Pokud má daný index již plnou kapacitu, je nejstarší informace odebrána. Okénka mají tzv. pyramidové schéma. Kapacita modelu je dána následujícím vztahem (2.2).

$$0 \equiv clock\ time \mod order^{window_i} \tag{2.1}$$

$$model\ capacity = order \cdot capacity \tag{2.2}$$

2.5.4 Vzorkování

Vzorkování vytváří reprezentativní vzorek datového toku. Provedení výpočtu nad reprezentativním vzorkem dat je dostatečně aproximovatelný výsledku, který byl proveden nad celým datovým tokem [7]. Položky jsou uloženy v datové struktuře fixní velikosti (velikost vzorku). Jednotlivé algoritmy se liší ve způsobu, jakým je reprezentativní vzorek vybrán.

Tok může být nekonečný, proto by vzorkovací algoritmus měl mít adaptační metody. Adaptačními parametry mohou být frekvence vzorkování nebo pravděpodobnost zapomínání vzorků. Vzorek musí být po adaptacích stále

reprezentativním vzorkem. Algoritmy založené na vzorkování jsou například reservoir sampling, min-wise sampling, concise sampling nebo sticky sampling.

2.5.5 Náčrt

Náčrt (anglicky *sketch*) je datová struktura, která reprezentuje část datového toku. V náčrtu mohou být uloženy libovolné agregované hodnoty toku. Kombinací menších náčrtů vznikne nový náčrt reprezentující větší část datového toku. Kombinace náčrtů není náročné na zdroje a čas. Na základě náčrtu je možné získat zpět informace, které tokem protekly [9].

Náčrt využívá deterministickou heuristiku místo pravděpodobnosti k výběru položek. Například v počtu unikátních položek je heuristikou prefix hashe položky. Algoritmy založené na náčrtu jsou například count sketch, count-min sketch nebo count-max sketch.

2.5.6 Vlnka

Vlnka popisuje vlastnosti proudu funkcí. U dané vlnky se ladí parametry nebo výčet funkcí, ze kterých je datový tok složen. Reprezentace se používá se pro dekompozici hierarchie dat a shrnutí položek [6]. Příkladem funkce je Haarova vlnka, která je jednoduchá na implementaci i výpočet. Velikost plochy funkce nad a pod osou je shodný, čímž je pozitivní i negativní energie v rovnováze. Každý signál má určitou dobu trvání a množství energie.

2.5.7 Histogram

Základní myšlenkou je rovnoměrné rozdělení dat do přihrádek. Tento předpoklad se však s vývojem datového toku může změnit. Proto přihrádky mohou mít fixní rozsah a nebo optimalizovaný rozsah pro minimální rozptyl v přihrádce [7]. Histogramy mohou být použity, pokud jsou položky ze statického rozsahu. Paměťová náročnost je dána počtem skupin.

Část II

Rešerše

Kapitola 3

CLEF NewsREEL challenge

Oddělení NewsREEL od roku 2014 do roku 2017 pořádalo každoročně soutěž v doporučování novinových článků. Každý ročník měl aktuální dataset článků pro daný rok. Testování probíhalo ve dvou fázích: online a offline. Úkolem účastníků bylo navrhnout algoritmus na doporučování novinových článků tak, aby splnil požadavky na doporučení do 100 ms a zároveň měl algoritmus co nejlepší výsledky metriky CTR. Krátce po ukončení soutěže byla pořádána konference, na které byly probírány výsledky a nové poznatky. V rámci soutěže byly publikovány vědecké články, které popisují implementaci algoritmů a dosažené výsledky. Tato kapitola obsahuje rešerši materiálů, které byly vydány v rámci ročníků soutěže CLEF NewsREEL challenge. V podkapitolách jsou zkoumány použité algoritmy, použité technologie a dosažené výsledky.

3.1 Algoritmy

V této části jsou popsány algoritmy, které byly použity k doporučování. Informace z materiálů byly rozděleny podle typů doporučování. U některých algoritmů bylo možné doplnit i získanou CTR. Obecný popis přístupů a jejich důsledků je popsán zde [10]. Tento materiál je stručný a neobsahuje dosažené výsledky. Proto jsou jednotlivé kapitoly rozšířeny o výsledky a detaily, které jsou v dalších materiálech popsány.

3.1.1 Nedávné transakce

První variantou jsou transakce popisující navštívené články. Uživateli jsou doporučeny nedávno čtené články komunitou. Jedná se o výpočetně i paměťově nenáročný přístup, který je dobře škálovatelný pro větší množství doporučení a nemá problém s doporučením nových položek. Celkově má algoritmus dobré výsledky, které však mají svůj limit daný především absencí přizpůsobení doporučení uživateli nebo zpracování informací o položce. Implementace algo-

ritmu může být cyklické pole fixní velikosti, které v případě příchodu nové položky, která se v poli nevyskytuje, přepíše nejstarší položku.

Druhou variantou jsou transakce popisující zveřejnění nebo aktualizování článku. Tato varianta nemá problém s doporučením nových článků. Aktualizace starších článků, které již nejsou populární, ale i tak jsou opraveny, způsobují přidání položky do doporučení, což snižuje celkový CTR. Implementace algoritmu může být totožná jako v předchozím případě. Algoritmus má tedy podobné vlastnosti jako doporučování posledních čtených článků. Na datasetu z roku 2017 dosáhl tento algoritmus 0,2 % CTR [11].

3.1.2 Populární články

Podobně jednoduchým algoritmem je doporučení nejčastěji navštěvovaných článků, které ještě uživatel neviděl. Vzhledem k velké návštěvnosti článků je pravděpodobné, že obsah je pro uživatele zajímavý. Systém si musí pamatovat dočasnou historii každého uživatele. Časová náročnost je nízká.

Na druhou stranu, stejně jako v předchozím případě, toto řešení nebere v potaz preference uživatele ani informace o článku. Doporučování nových položek je problém, neboť nové položky nejsou populární. Implementace algoritmu vyžaduje strukturu na zapamatování uživatelů (například slovník) a strukturu na četnosti položek. Během trénování si algoritmus vytváří statistiky o návštěvnosti položek. V případě požadavku o doporučení vybere ty položky, které mají nejvíce návštěv a zároveň ještě nebyly uživatelem navštíveny. V roce 2016 algoritmus inspirovaný tímto přístupem dosáhl CTR 0,3 % [12].

Popularita článku znamená, že na článek kliklo velké množství uživatelů. Kvalitu populárního článku lze změřit délkou času stráveného na stránce. Tuto hodnotu lze měřit jen pro delší sessions, neboť se jedná o rozdíl časových razítek mezi dvěma po sobě příchozími transakcemi v rámci jedné session. V případě posledního článku v session není možné tuto hodnotu zjistit. Parametry k optimalizaci mohou být hranice pro délku čtení, rozhodnutí o posledním článku v sekvenci a po kolika přečteních je článek považován za populární.

Populární cesta je další varianta algoritmu, kde si algoritmus pamatuje, kam uživatel přešel z dané položky nebo odkud na ni přišel. Pro každou položku je zapamatován počet přechodů. Na základě aktuální položky, dostupnosti session uživatele a historii přechodů je algoritmus schopný vybrat, jaké bude další doporučení. Pro každou položku je nutné si pamatovat přechody do všech položek, což může znamenat velkou paměťovou náročnost. Výpočetní složitost je nízká neboť všechny informace jsou již dostupné. V roce 2017 tento přístup dosáhl 0,1 % [11].

3.1.3 Kolaborativní filtrování

Standardním přístupem je doporučování na základě kolaborativního filtrování. To může být zaměřeno na uživatele nebo na položky. Uživatelé jsou v dané časové lokalitě shlukováni a jejich požadavky jsou ukládány pro daný shluk. Vzhledem k absenci profilů a anonymním uživatelům je možné využít metadata dotazu. V případě doporučení je uživatel zařazen do správného shluku, na základě kterého jsou poté vybrána doporučení. Paměťová náročnost se odvíjí od zvolené dimenzionality prostoru. Časová náročnost závisí na složitosti výpočtu podobnosti uživatelů. Správné přiřazení uživatele do shluku je navíc limitováno dostupností historie a profilu uživatele.

V případě kolaborativního filtrování zaměřeného na položky jsou články doporučovány podle uživatelů, kteří taktéž četli daný článek. Doporučení je založené na informacích o článku a komunitě uživatelů. Není zde problém identifikace uživatele ani s doporučením článků novým uživatelům. Nevýhodou je ignorování kontextu dotazu a doporučení nově příchozích článků. V roce 2017 tento přístup dosáhl 1,35 % [13].

3.1.4 Filtrování založené na znalostech

Doporučení založené na obsahové podobnosti sice nemá problém se zpracováním nových položek, ale za to se potýká s výpočetně náročnou analýzou článku. Články mají malé množství strukturovaných informací a mnoho charakteristických rysů je součástí nestrukturovaných dat (například titulek článku). Nevýhodou je ignorování preferencí uživatele a těžko vyhodnotitelná kvalita článku. Navic výzkumy naznačují, že kolaborativní filtrování má větší důsledek na doporučení než doporučení založené na obsahu [14].

3.1.5 Asociační pravidla

Asociační pravidla vytěžují informaci z transakcí a článků, které na základě parametrů, jako podpora, důvěra a lift, vytváří asociační pravidla přechodu. Pravidla nejsou limitována na informace o uživateli a session, ale mohou využít třeba metadata dotazu a nebo jeho kontext. Z tohoto hlediska mají asociační pravidla více možností, jak zlepšit doporučení. Algoritmy pro získání asociačních pravidel jsou například FPGrowth nebo apriori. Výpočet pravidel probíhá v dávkách. V roce 2017 tento přístup dosáhl CTR 0,2 % [15].

3.1.6 Obrázky

Součástí datového toku jsou záznamy o vytvoření a aktualizování článku. Tyto záznamy mohou obsahovat URL adresu, kde je dostupný obrázek, který byl s článkem spojen. Na základě obrázku je následně možné přizpůsobit doporučení uživatelům. Příkladem jednoduchého klasifikátoru obrázku [12] je rozhodnutí o jeho zajímavosti. Heuristika definovala zajímavý obrázek,

jako jednoduché pozadí s dominantním prvkem uprostřed. Pouze články se zajímavými obrázky byly doporučovány. V roce 2016 dosáhl tento přístup 0,4 % CTR, i když zvolený dataset neodpovídal soutěžním rozměrům. Obrázky jsou z portálů průběžně smazávány, což snižuje schopnost opakovatelnosti experimentů.

3.1.7 Soubor

Algoritmus obsahuje vícero různých algoritmů, které provádějí doporučení. Na základě zvolené heuristiky jsou následně vybírána doporučení případně mix doporučení. Z porovnání vícero algoritmů se zdá, že neexistuje konkrétní model, který by měl výrazně lepší výsledky než ostatní. Závěrem zmíněné práce je tvrzení, že kvalita doporučení závisí na kontextu doporučení a znalosti domény [16]. Může se tedy jednat i o vhodné využití více různých modelů. Například rozdělení modelů dle domén.

3.2 Technologie

Účastníci mohli zvolit pro implementaci doporučujícího systému programovací jazyk dle své volby. Protokol datového toku je ORP. Pro jazyky JAVA[1], PHP[2] a Node.js[3] vzniklo SDK, které implementovalo tento komunikační protokol. Účastníci, kteří si jedno z těchto SDK vybrali, se mohli soustředit na návrh samotného algoritmu doporučování.

Uživatelé, si mohli zvolit lokální nebo distribuované zpracování. Algoritmy, které lokálně zpracovávaly velké množství zpráv dosáhly paměťových nebo výpočetních limitů. Distribuované řešení nabízí škálovatelnost, kterou je možné využít v případě, že je algoritmus úspěšný a měl by zvládnout více požadavků zároveň. Ti soutěžící, kteří se rozhodli pro distribuované zpracování, si vzhledem k časovým požadavkům na doporučení a objemu dat vybrali jeden z následujících nástrojů: Apache Spark nebo Apache Flink [17].

Prvním nástrojem byl Apache Spark[4], který umožňuje distribuované paralelní zpracování datového toku. Uzly klasteru jsou rozděleny na řídící a exekuční. Řídící uzel dohlíží na zpracování úkolu exekučními uzly. Datový tok je rozdělen do okének. Velikost okénka (v kontextu Apache Spark nazýváno dávkou - anglicky *batch*) může být specifikována počtem položek nebo časem. Po naplnění okénka jsou data řídícím uzlem distribuována na exekuční uzly. Paralelizace je dosáhnuto principem MapReduce. Postupnými vylepšeními nástroje bylo dosaženo zpracování proudu dat (anglicky *data streaming*).

[1] https://github.com/plista/orp-sdk-java
[2] https://github.com/plista/orp-sdk-php
[3] https://github.com/jaroslav-kuchar/orp-sdk-node
[4] https://spark.apache.org/

Druhým nástrojem je Apache Flink[5], který umožňuje zpracování datového toku. Taktéž se jedná o nástroj pro distribuované paralelní zpracování datového toku. Jeho počátek je založený na okamžitém zpracování každé položky. Nástroj postupnými úpravami dosáhl funkcionality mikro-okének (anglicky *micro-batchnig*) a okének.

Technologie jsou si velmi podobné, avšak jejich počáteční myšlenka byla rozdílná. Správný výběr záleží na způsobu jejího využití. Převedením okének na proudění se sníží odezva odpovědi a zároveň se sníží celková propustnost. Použitím okének se zvýší odezva na odpověď, neboť zpráva není zpracována hned, ale na druhou stranu se zvýší celková propustnost systému. Při výběru technologie je důležité zvolit, k jakým částem výpočtu má být technologie použita a jaká kritéria jsou pro algoritmus důležitá. Výhodou těchto technologií bylo zpracování velkého množství dat s vysokou mírou odpovědí a horizontální škálovatelností.

3.3 Výsledky

Soutěž měla celkem 4 ročníky. Po skončení online testování byl vydán vědecký článek shrnující průběh soutěže a dosažené výsledky online testování [18][19] [17][20]. Základem každé soutěže byl baseline algoritmus, který se měli snažit soutěžící porazit. Účastníci náhrali své kandidáty do platformy a pomocí protokolu byla část datového toku přesměrována na doporučovací systémy. U kandidátů se měřilo množství zpracovaných zpráv, CTR a míra odpovědi (anglicky *response rate*).

Nejdůležitější metrikou byla úspěšnost doporučení neboli CTR. Během všech ročníků se podařilo dosáhnout nejlepší průměrné hodnoty 2,5 %. Přesnost baseline algoritmu byla kolem 0,59 %. Tato metrika byla výrazně ovlivněna příchozím datovým tokem. Významnými faktory byly: celková doba testování, denní doba a vydavatelství [21].

Kandidátům obvykle chvíli trvalo, než se naučili správně doporučovat položky. Hodnota CTR se začala zlepšovat až po pár hodinách či dnech. Výhodu měli kandidáti, kteří byli větší část soutěže online. Kandidát mohl být dočasně vyřazen z online testování kvůli chybám při zpracování požadavku.

Datový tok obsahoval požadavky na doporučení z více vydavatelství. Různorodé zaměření článků má vliv na výslednou přesnost jednotlivých algoritmů k doporučování. Algoritmus mohl být ve výhodě, pokud přidělený datový tok obsahoval větší poměr článků od vydavatelství, pro které byly položky vhodné.

Množství požadavků na zpracování se během dne liší. Některé přístupy jsou vhodné v ranních hodinách a jiné zase v odpoledních. Kandidát mohl být ve výhodě, pokud datový tok obsahoval větší část doporučení z doby, kdy byl daný přístup vhodnější.

[5] https://flink.apache.org/

Dalším měřeným údajem bylo množství zpracovaných požadavků. Tento údaj vypovídá nejen o tom, kolik algoritmus doopravdy zpracoval údajů, ale také o tom, že byl algoritmus úspěšný v doporučování a měl vysokou míru odpovědí. Přiřazení zpráv kandidátům probíhalo algoritmem *multi-armed bandit*. Algoritmy zpracovávaly od desítek tisíc až po jednotky milionů požadavků.

Míra odpovědí systému byla vyjádřena poměrem požadavků, na které bylo vráceno doporučení. Míra odpovědi měla velký vliv na počet zpracovaných zpráv. Dle naměřených dat se jedná o exponenciální závislost mezi množstvím zpracovaných dat a mírou odpovědi. I přes to existovala výrazná skupina kandidátů, která přes vysokou míru odpovědí nedostala takové množství zpráv. Toto je vysvětleno tím, že někteří kandidáti mohli být po určitou dobu odstaveny z online testování.

Výsledky z online a offline testování se liší. I přes pokus organizátorů minimalizovat rozdíl mezi těmito druhy testování se rozdíly vyskytují [22]. V rámci soutěže byl vydán článek, který porovnával rozdíly v online a offline testováních. Stejné algoritmy dosáhly rozdílných výsledků ať už v rámci jednoho nebo více ročníků. Závěrem práce je upozornění na opatrnost při interpretaci závěrů vlivu algoritmu na výsledné metriky.

Kapitola 4

Datový tok

Tato kapitola je zaměřena na popis datového toku. Nejprve bude popsána doména položek a její specifika. Následně bude krátce popsán autor datasetu a způsob, jakým byly zprávy v toku generovány. Závěrem kapitoly jsou popsány druhy zpráv, které se v toku nacházejí včetně jejich významu.

4.1 Novinové články

Pro experiment s algoritmy byl vybrán dataset transakcí popisující doporučování novinových článků. Narozdíl od doporučování zboží, hudby nebo filmů má doporučování novinových článků řadu specifik [23]. Naučený model je vázán na časovou lokalitu, pracuje s omezeným množstvím informací o uživatelích, s nestrukturovanými daty o položkách, velkou důležitostí nově příchozích zpráv, striktními požadavky na doporučení a adaptabilitou ke změnám.

Položky se velmi rychle mění a jejich životní cyklus je relativně krátký. Novinové články vznikají ve velkém množství a zároveň velmi rychle zastarávají. K existujícím článkům vznikají revize a následné články.

Rozlišit články popisující stejnou událost, různé aspekty události a vývoj dané události, je výpočetně náročné. Většina slov se může shodovat a přitom závěry článků mohou být rozdílné. Informace o článcích jsou většinou nestrukturované. Mezi strukturované informace patří například kategorie, klíčová slova, autor, datum vydání nebo číslo verze. Tyto informace však nejsou jednotné napříč portály a ne vždy musí být dostupné. Nedostatek strukturovaných informací ztěžuje doporučení na základě obsahu. Informace obsažené v titulku, textu a přiložených grafikách článku není možné rychle a jednoduše extrahovat.

Velký důraz je kladen na doporučování nových položek. Uživatelé sledují zprávy pravidelně a na populární události jsou vydávány navazující články. Starší články obvykle uživatel už zná, nebo se o nich dozví z novějších zpráv. V případě vývoje situace může mít uživatel zájem přečíst si starší články, které popisují detailněji určité události.

Míra překvapení je důležitou metrikou doporučení. Novinové články by měli rozšířit povědomí uživatele o událostech, které se staly. Uživatel čte pro něj nové a nečekané články. Jen zřídka čte články, které již dříve přečetl. Doporučovací systém může preferovat články, které popisují pro uživatele neznámé události.

Míra exkluzivity zprávy hraje větší roli než zájem uživatele. V případě novinky nebo populární zprávy může uživatel kliknout na položku, i když neodpovídá jeho zájmům.

Portály obvykle nevyžadují, aby byl uživatel přihlášen, když si čte novinové články. Chybějící explicitní profil uživatele, tak nahrazuje implicitní profil na základě kontextu a metadat. Stejný uživatel může přistupovat k jednomu portálu z vícero zařízení a současně z jednoho zařízení mohou na portál přistupovat různí uživatelé.

Jednoznačná identifikace uživatele mezi návštěvami portálu je obtížná, což ztěžuje přizpůsobení doporučení na základě uživatelových preferencí. Získané informace jsou obvykle aplikovatelné jen v časové lokalitě. Pokud uživatel znovu přijde na portál a není možné ho identifikovat například pomocí cookies, tak se jeví jako nový uživatel.

Obsahem novinového článku obvykle bývá konkrétní událost nebo pohled na událost. Uživatel při výběru zprávy může reagovat na známou entitu, o které se v článku píše. Entitou může být například osoba nebo společnost. Uživatel se může zajímat o danou entitu jen v konkrétních událostech, které vycházejí z preferencí uživatele.

Příkladem je situace, kdy si uživatel přečte článek o novém filmu ne proto, že je fanouškem filmu, ale kvůli tomu, že v něm hraje jeho oblíbený herec. Uživatel má neutrální vztah k filmům, ale v případě, že v něm hraje jeho oblíbený herec je téma zajímavé. Doporučovací systém by měl tento trend reflektovat. Vzhledem k podstatě položek uživatel očekává doporučení novinových článků v řádu milisekund. Je zde tedy kladen důraz na rychlost doporučení. Kvalita doporučování článků se obvykle měří metrikou CTR. Není zde možné využít metrik RMSE nebo MAE jako při predikci hodnocení.

4.2 Plista

Společnost Plista[6] se zaměřuje na real-time doporučování obsahu uživatelům na webových stránkách. Obsahem může být například zboží, novinový článek nebo reklama. Pomocí standardizovaného protokolu odesílá konzument informace o položkách, aktivitě uživatelů a požadavcích na doporučení. Zprostředkovatel na základě obdržených informací a požadavků doporučuje cílové položky. Zobrazení doporučení může být například ve formě banneru, widgetu nebo součástí toku informací.

[6] www.plista.com/about

ORP komunikuje standardizovaným protokolem mezi zprostředkovatelem a konzumentem. Zprostředkovatelem je v tomto případě společnost Plista a konzumenty jsou vydavatelské portály. Pomocí jednotného protokolu mají konzumenti přístup k širokému portfoliu řešení.

Hlavním benefitem využití této služby je přesunutí doporučovací logiky na společnost Plista. Ta se stará například o kvalitu doporučení, mapování uživatelů napříč zařízeními nebo vývoj preferencí uživatele. Výsledkem je zvýšení návštěvnosti portálů a prodloužení času, který uživatel na portálu stráví. V případě zobrazení reklam se jedná o způsob monetizace obsahu.

4.3 Popis dat

Data vznikla za vědecké spolupráce společnosti Plista GmbH a univerzity TU Berlin. Tyto dva subjekty spolupracovaly při každoroční CLEF NewsREEL challenge. V této práci je použit dataset, který vznikl k rámci soutěže CLEF NewsREEL challenge 2017. Soubor obsahuje záznam proudu, který byl zpracován prostřednictvím platformy ORP. Každý řádek souboru odpovídá jedné transakci. Datový tok obsahuje komunikaci mezi vícero konzumenty, a tak je důležité, aby model vrátil doporučení jen v rámci daného portálu.

Každá transakce se skládá ze 3 částí: typ, obsah a časové razítko [24]. Časové razítko odpovídá času vytvoření transakce. Transakce jsou řazeny vzestupně dle tohoto parametru. Typ transakce je identifikován pevně daným řetězcem. Soubor obsahuje tři druhy zpráv: *item_update*, *recommendation_request* a *event_notification*. Obsah je reprezentován JSON objektem a jeho obsah se liší dle typu transakce. Zprávy jsou odesílány v následujících kontextech:

- *item_update* - Zprávu odesílá konzument, když dojde ke zveřejnění nového článku nebo úpravě již existujícího článku.

- *recommendation_request* - Zprávu odesílá portál, když vyžaduje od zprostředkovatele doporučení dle zadaných kritérií. Jedním z kritérií je počet doporučení.

- *event_notification* - Transakce zastřešuje události, které mohou nastat během uživatelovy interakce s portálem. Jednou z událostí je kliknutí na položku, která byla uživateli doporučena. Tato zpráva obsahuje kontext notifikace jako geolokaci, typ prohlížeče nebo odhad pohlaví uživatele. Parametry obsažené v kontextu jsou nepovinné, ale mohou být využity pro optimalizaci doporučení.

Data jsou uložena ve strukturách, které se nazývají vektory. Vektor obsahuje kolekci dvojic klíč-hodnota, kde klíč je vždy číselná hodnota. Hodnoty v jednom vektoru mají stejný typ. Typy vektorů jsou: simple, list nebo cluster. Simple vektory obsahují pouze dvojice klíč a číselná hodnota. Řetězce patří

do této kategorie a jsou reprezentovány unikátním číslem. List vektor obsahuje pole numerických hodnot. Cluster obsahuje skupiny disjunktních rozsahů nebo množiny počtů.

Informace o článku obsahují identifikátor (id), titulek (title), úryvek textu (text), URL článku (url), identifikátor domény (domainid), URL obrázku (img), datum vytvoření článku (created_at), datum poslední aktualizace (updated_at), znaky (flag), verzi (version) a filter (Filter). Identifikátor může být využit k unikátní identifikaci položky. Titulek a úryvek textu popisuje, o čem článek je. URL článku odkazuje na stránku, kde je článek dostupný a URL obrázku odkazuje na adresu, kde je dostupný obrázek, které je k článku přiřazen. Identifikace domény slouží ke přiřazení článku vydavatelství. Datum vytvoření a datum poslední aktualizace jsou reprezentovány časovou značkou. Za každou změnu v položce je zvýšeno číslo verze a aktualizován čas poslední aktualizace. Znaky popisují atributy položky omezující doporučení položky. Jedná se o číselnou hodnotu, kde 0 znamená žádné omezení a libovolná jiná hodnota znamená nedoporučovat položku.

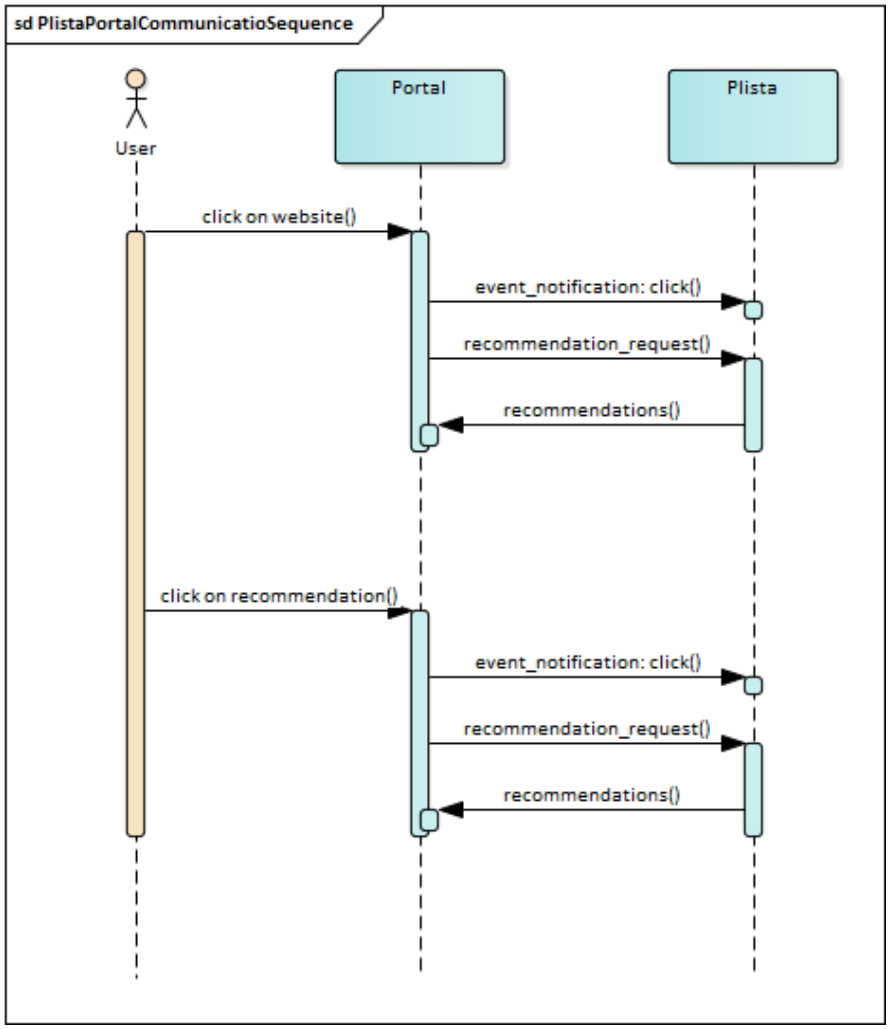
Uživatel je identifikován pouze na základě cookie. Ta je uvedena jak v požadavku na doporučení, tak ve zprávě o interakci. Není zde žádný záznam o detailech uživatele. Profil uživatele může být vytvořen z těchto dvou druhů zpráv. Zprávy nemusí mít vyplněný tento identifikátor a požadavky o doporučení mohou vyžadovat zákaz sledování. Takovýto uživatelé jsou bráni jako anonymní.

Články jsou psané v německém jazyce a byly zaměřeny na německy mluvící klienty. Dostupnost článků je celosvětová, ale nejvíce přečtených zpráv pochází z Německa, Rakouska a Švýcarska [10]. Novinové portály jsou zaměřeny na různé kategorie zpráv, jako například zpravodajství, sport nebo informační technologie.

4.4 Proces generování transakcí

V případě, že se uživatel dostane na stránku portálu, odešle portál zprávu typu *recommendation_request*, kde je definován očekávaný počet doporučení. Pokud uživatel klikne na jednu z následujících položek, je odeslána zpráva *event_notification* typu click. Daná zpráva obsahuje identifikaci článku, ze kterého byl klik proveden a zároveň identifikaci článku, na který je uživatel přenesen. V případě zobrazení stránky je opět odeslána zpráva typu *recommendation_request* a cyklus může začít od začátku. Celý proces je znázorněn v diagramu 4.1.

Obrázek 4.1: Proces generovaných zpráv



Kapitola 5

Platforma

V rámci oddělení NewsREEL vznikla platforma pro vyhodnocování doporučovacích systémů Idomaar, která zpracovává datové toky. Účelem platformy bylo vytvoření prostředí, ve kterém může akademický svět provádět online a offline testování s velkým množstvím dat. Doménou doporučování jsou novinové články. Nad touto platformou byla každoročně pořádána soutěž CLEF NewsREEL challenge. V této kapitole je popsán framework Idomaar, který byl použit k vyhodnocení soutěže. Dále je popsána platforma StreamingRec, která je zjednodušeným pokračováním tohoto frameworku Idomaar. StreamingRec je využit v experimentální části k vyhodnocení algoritmů nad datovým tokem.

5.1 Idomaar

Vybraný dataset byl součástí CLEF NewsREEL challenge, která probíhala od roku 2014. Tato soutěž měla 2 části: offline replay a online testing. V offline části si mohl účastník vyzkoušet naimplementovat kandidáta na testování a odladit jednotlivé parametry. V druhé části byl kandidát použit v online doporučování. Účastník mohl průběžně sledovat KPI jeho algoritmu a případně vyladit chyby, které se objevily.

Vzhledem k možnosti dvojího testování byla navržena platforma Idomaar[7], která je určena k testování doporučovacích systémů. Jednotlivé části platformy běžely na virtuálních strojích a komunikovaly spolu prostřednictvím portů. Tímto se simulace přesunula o něco blíže realitě a rozdíl mezi nasazením kandidáta do online a offline řešení byl minimální.

Projekt, ze kterého byla platforma financována, skončil v roce 2017. Vzhledem k reorganizaci oddělení na straně univerzity došlo k vynechání ročníku výzvy a společnost Plista v současnosti nenabízí online testování. V tuto chvíli není platforma dále udržována a návody na nastavení prostředí již nefungují, neboť balíčky, které jsou požadovány, již nejsou dostupné.

[7] https://github.com/crowdrec/idomaar

Výsledkem je nemožnost navázat na velké množství existujících publikací, které využívaly tuto platformu. Vzhledem k přiblížení se reálnému testování není možné s platformou provést ani offline testování s existujícími daty. Z tohoto hlediska není možné platformu využít.

5.2 StreamingRec

Tato platforma vznikla za účelem opakovatelného offline testování doporučovacích systémů [25]. Pro implementaci byl zvolen jazyk JAVA. V současnosti obsahuje platforma následující funkcionality:

- implementace algoritmů;
- implementace heuristik;
- transformace datasetu CLEF NewsREEL challenge;
- deduplikace datasetu;
- rozdělení datasetu pro trénování a testování;
- session;
- filtrování transakcí.

Platforma navazuje na platformu Idomaar s datasetem CLEF NewsREEL challenge. Projekt proto obsahuje nástroje, které ve dvou fázích transformují dataset do akceptovatelné formy pro experimenty. Na základě datasetu a konfigurace je možné opakovaně provádět testy s různými parametry, aniž by muselo docházet ke změně nebo kompilaci kódu.

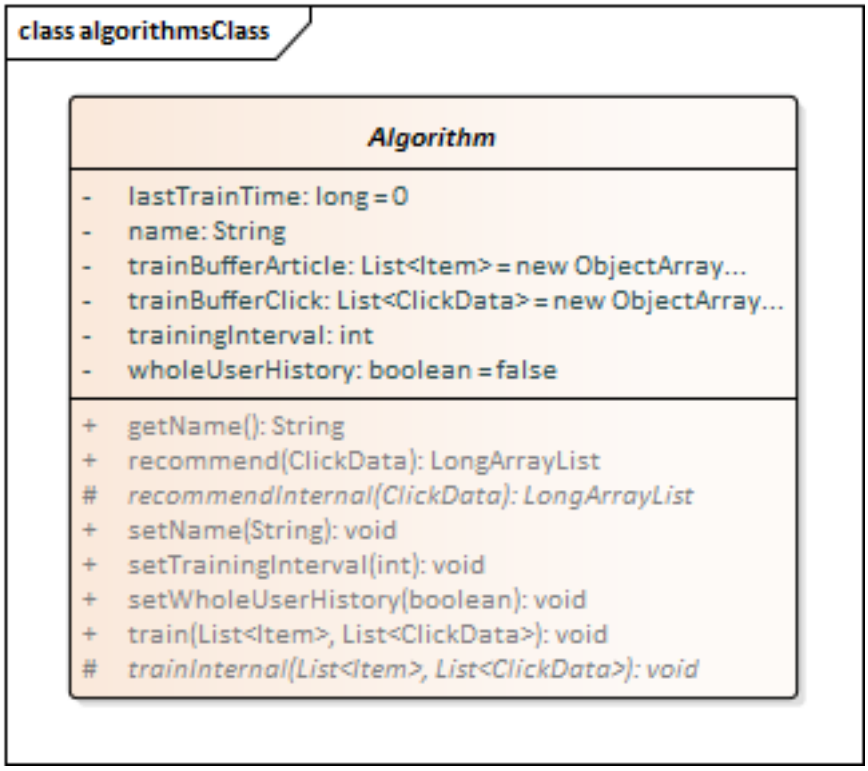
Povinnými parametry platformy jsou cesty k souborům obsahující: transakce, položky, konfigurace algoritmů a konfigurace metrik. Na základě zadaných argumentů jsou data přehrána. Algoritmy doporučují položky a metriky měří kvalitu doporučení. Platforma umožňuje jednoduchou implementaci vlastních algoritmů doporučení a metrik.

Vstupními daty může být libovolný záznam datového toku, který splňuje definovaný formát dat. Vstupní data jsou rozdělena do dvou souborů ve formátu CSV. První soubor popisuje transakce. Jedna transakce obsahuje následující informace: identifikaci položky, identifikaci uživatele a časové razítko. Záznamy v souboru jsou seřazené vzestupně podle časového razítka. Druhý soubor popisuje jednotlivé položky. Jedna položka musí obsahovat identifikátor, aby mohly být položky spárovány s transakcemi. Zbylé položky jsou dobrovolné. Informace o uživatelích je minimalizována na jejich unikátní identifikaci.

Konfigurace se skládá ze dvou souborů ve formátu JSON. První soubor definuje použité algoritmy a zvolené parametry. Atribut name slouží jako

unikátní identifikátor algoritmu. Atribut algorithm definuje jméno třídy, která obsahuje implementační logiku a rozšiřuje třídu *Algorithm*. Popis této třídy je možné vidět na diagramu 5.1. Konfigurace algoritmu následně obsahuje dvojice klíč-hodnota, které odpovídají jednotlivým parametrům konkrétní implementace algoritmu. Obdobně je strukturovaný i druhý soubor definující použité metriky. Místo atributu algorithm je zde atribut metric, který identifikuje jméno třídy obsahující logiku metriky. Tato třída musí rozšiřovat abstraktní třídu *Metric*. Detaily této třídy je možné vidět na obrázku 5.2.

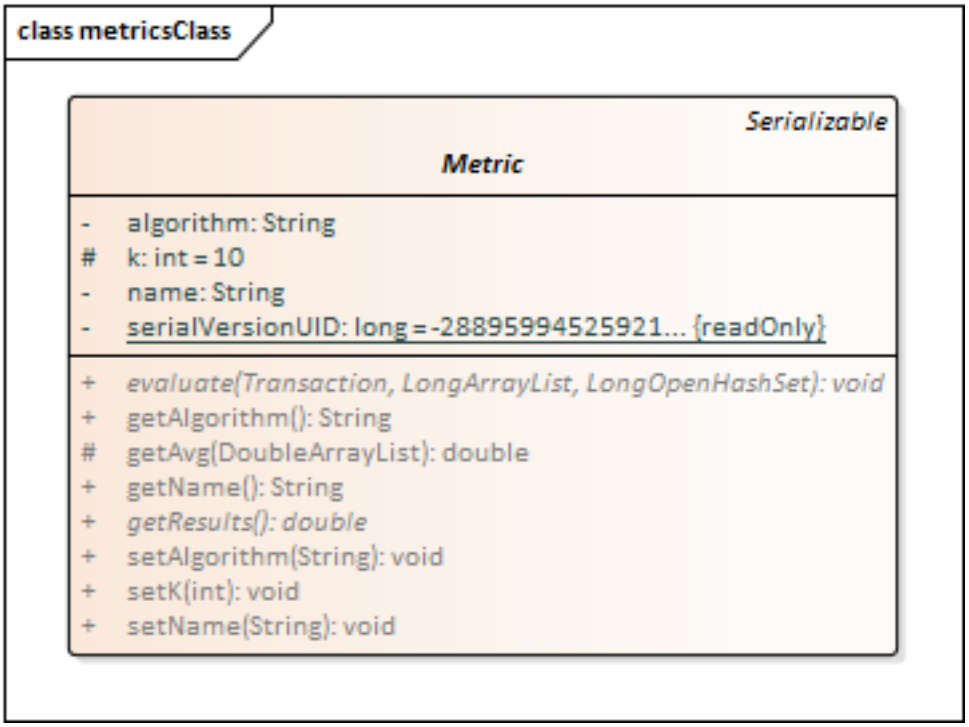
Obrázek 5.1: Diagram třídy *Algorithm*



Algoritmy musí implementovat abstraktní metody svého předka. První metodou je metoda *trainInternal*, která slouží k naučení algoritmu na transakcích. Tato metoda je zavolána na konci fáze učení. Poté v závislosti na parametru *trainingInterval* je přeučení voláno po každém požadavku na doporučení nebo po uplynutí daného časového rozmezí. Druhou metodou je *recommendInternal*, která slouží k doporučení položek. Algoritmus může využít informací, které získal během trénování. V parametrech metody je současná transakce, historie současné session a historie všech interakcí uživatele.

Metriky musí rozšířit abstraktní třídu *Metric* a implementovat její abstraktní metody. První metodou je *evaluate*, která vyhodnotí aspekt doporučení. Pro vyhodnocení má k dispozici současnou transakci, budoucí transakce

Obrázek 5.2: Diagram třídy *Metric*



uživatele v rámci session a doporučené položky. Metoda je volána pro každý požadavek o doporučení položek. Druhou metodou je *getResult*, která vrací číslo s plovoucí desetinnou čárkou. Tato metoda je volána po dokončení testování. Co je konkrétně vypočítáno a jaká je interpretace výsledku, závisí na implementaci metriky. Může se jednat například o precision, recall, RMSE, MAE, pokrytí nebo délku trénování.

Dobrovolnými parametry platformy jsou deduplikace proudu před zpracováním, nastavení maximální délky jedné session, minimálně počet transakcí v session, poměr dat trénování a počet vláken pro evaluaci.

Velikost session má vliv na počet položek, které jsou považovány za jednu session. Tato kolekce je následně použita k vyhodnocení v metrikách. Příliš krátká session způsobí, že není možné určit přesnost doporučení, neboť každá session má jen jeden článek. Příliš dlouhá session způsobí, že metrika při vyhodnocení předpokládá, že uživatel mohl kliknout na všechny položky, o kterých má v budoucnu v toku záznam. Session by měla sloužit k popisu aktuálního prohlížení uživatele.

Platforma nabízí funkci deduplikace datového toku. Transakce je označena za duplikát, jestliže existuje transakce, která má stejný identifikátor položky a stejný identifikátor uživatele jako duplikát a jejíž časová značka je menší nejvýše o jednu minutu. Pokud je nástroj aktivní nejsou duplikáty použity při testování. Odebrání duplikátů má vliv na odstranění krátkých session.

Část III

Dolování dat

Kapitola 6

Dataset

Autor datasetu, popis protokolu a popis platformy na zpracování byly popsány v rešerši. V této kapitole bude popsán způsob získání datasetu, který je možné použít v platformě. Dále bude provedena analýza dat, která ukáže základní vlastnosti datového toku. Tato kapitola je strukturována dle procesu dolování dat.

6.1 Sběr dat

Dataset je rozdělen do dvou částí. První část je obyčejný soubor s koncovkou log. Tento soubor obsahuje 100000 záznamů z rozmezí 21. 1. 2016 - 31. 1. 2016. Většina záznamů byla uložena během poslední hodiny intervalu. Tato část nebude v experimentech využita z důvodu málo dat.

Druhá část se skládá z 28 komprimovaných souborů s průměrnou velikostí 521 MB. Pro komprimaci byla využita metoda GZIP. Každý soubor odpovídá části datového toku za den uvedený ve jménu souboru. Soubory odpovídají toku za dny v rozmezí 1. 2. 2016 - 28. 2. 2016. Před zpracováním jednotlivých souborů musí dojít k dekompresi.

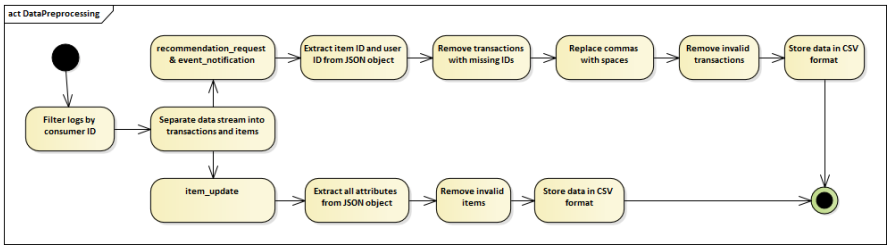
6.2 Extrakce charakteristických rysů

Data od Plisty nejsou ve formátu podporovaným platformou StreamingRec. Data musí být nejdříve transformována na formát CSV a následně musí být přizpůsobena formátu platformy. Celý proces úprav je znázorněn v diagramu 6.1. Současný dataset má následující nedostatky:

- obsahuje v jednom souboru informaci o položkách a transakcích;
- nesplňuje formát platformy;
- text článků obsahuje čárky v textu;

- záznamy nemusí mít vyplněné všechny požadované atributy.

Obrázek 6.1: Proces zpracování originálního datasetu



Požadavek na doporučení přichází vždy od jednoho vydavatele. Doporučené položky musí pocházet od stejného vydavatele. To znamená, že všechny položky nebo transakce, které pocházejí od jiného vydavatele, mohou být vyfiltrovány. Vzhledem k tomu, že vydavatelé mohou publikovat zprávy různých žánrů je pravděpodobné, že i jejich doporučování se liší. Například životní cyklus technických zpráv může být jiný než u bulvárních článků. Z tohoto hlediska dává smysl mít pro každého vydavatele jiný doporučovací model.

Problém doporučení článků od stejného vydavatele lze vyřešit rozdělením velkého datasetu na menší, kde každý dataset odpovídá jednomu vydavateli. Navíc je zapotřebí každý takto vzniklý dataset rozdělit do dvou souborů, na položky a transakce. Rozdělení do těchto dvou skupin závisí na typu zprávy, kde položky jsou popsány pouze zprávami typu *item_update*, zatímco transakce jsou popsány jako *recommendation_request* nebo *event_notification*.

U položek jsou vyextrahovány všechny atributy z obsahu JSON objektu. Vzhledem k výslednému formátování jsou všechny čárky nahrazeny mezerou. Tato změna nemění význam dat a zároveň umožní oddělování hodnot čárkou ve výsledném formátu. Každá položka musí mít atributy identifikátor, doména a časové razítko, aby mohla být využita platformou. Všechny ostatní atributy mohou být nevyplněné. Po extrakci atributů je položka uložena ve formátu CSV.

U transakcí jsou z obsahu JSON objektu vybrány pouze identifikátory položky a uživatele. Zprávy *recommendation_request* nebo *event_notification* typu *click* je možné považovat za stejné ve zjednodušeném modelu, neboť obě vypovídají o stejných položkách. V jednom případě uživatel na položce již je a v druhém případě je uživatel na stránku teprve přesměrován. V obou případech se jedná o indikátor zájmu uživatele o položku. Z tohoto rozhodnutí vyplývá, že výsledný datový tok obsahuje pro některé stránky dva totožné záznamy.

Uživatel má možnost zakázat systému jeho sledování. V tom případě je identifikátor uživatele nevyplněn. Tyto transakce jsou považovány za anonymní a jsou součástí cílového datového toku.

Tabulka 6.1: Počet položek a transakcí pro jednotlivé vydavatelství

Identifikátor vydavatelství	Počet transakcí	Počet položek
418	12047587	3487
596	2	0
694	2991518	31
1677	22325016	46168
2522	59	0
3336	8301	0
13554	9546351	0
15739	11	0
35774	123297593	8070

V analýze dat jsou použity následující atributy: odhadované pohlaví, odhadovaný věk, odhadovaný příjem, jazyk, operační systém, prohlížeč, druh zařízení, kategorie, geolokace, poštovní směrovací číslo. I když protokol obsahuje větší množství atributů [24], tak některé atributy nemají definovaný význam. Vzhledem k anonymizaci dat by bylo obtížné tyto informace jakkoliv interpretovat. Výsledek je uložen ve formátu CSV.

6.3 Analýza dat

V této části bude nejprve popsána skladba datasetu a následně bude vybrán vydavatel, který bude použit k analýze a experimentům. Nad konkrétním vydavatelem budou zkoumány atributy položek a transakcí. Zaznamenaný datový tok obsahuje záznamy z rozmezí 31. 1. 2016 - 28. 2. 2016. Z tohoto rozmezí bude zkoumán vliv dnů a hodin na počet transakcí a jejich skladbu. Součástí této kapitoly je analýza řídkosti matice.

6.3.1 Výběr vydavatele

Zprávy pocházejí z 9 vydavatelství, přičemž u 4 vydavatelství je počet transakcí menší než 10 tisíc a u 5 vydavatelství nejsou žádné zprávy o položkách. Pokud nejsou započítána výše zmíněná vydavatelství, tak se počet položek pohybuje v rozmezí 31 až 46168 a počet transakcí je v rozmezí od 3 do 123 milionů. Dataset dohromady obsahuje 170 milionů transakcí a 57 tisíc informací o položkách. Konkrétní přehled je k nahlédnutí v tabulce 6.1.

Absence informací o položkách ztěžuje doporučování, avšak i přesto je možné nějaké položky doporučit. Portály s nízkým počtem transakcí nejsou analyzovány pro nedostatek dat, neboť doporučování položek by bylo velmi specifické pro dostupné záznamy. Pro experimenty a trénování parametrů byl vybrán vydavatel s identifikátorem 418, který odpovídá portálu ksta.de, a vede na stránky kolínského deníku Kölner Stadt-Anzeiger.

6.3.2 Analýza položek

Identifikátory položek je možné získat ze záznamů o položkách nebo z identifikátoru v transakcích. Transakce celkem referencují 159 tisíc položek. Záznamy o položkách popisují jen 1080 položek (0,6 %). Tento zlomek položek však tvoří 46 % všech transakcí, čímž je podtrhnuta jejich důležitost.

Více jak polovina položek nebyla nikdy aktualizována. To znamená, že datový tok obsahuje pouze jednu zprávu typu *item_update* s tímto identifikátorem. Čtvrtina položek z celku byla aktualizována jen jednou. Pouze v 7 % případů došlo ke dvěma aktualizacím položky během měsíce února. Z tohoto vyplývá, že položky nejsou často aktualizovány, případně většina položek je aktualizována málokrát. Nejčastěji aktualizovaná položka tvoří 5 % všech zpráv o položkách. Zpráva s tímto identifikátorem byla odeslána 171krát.

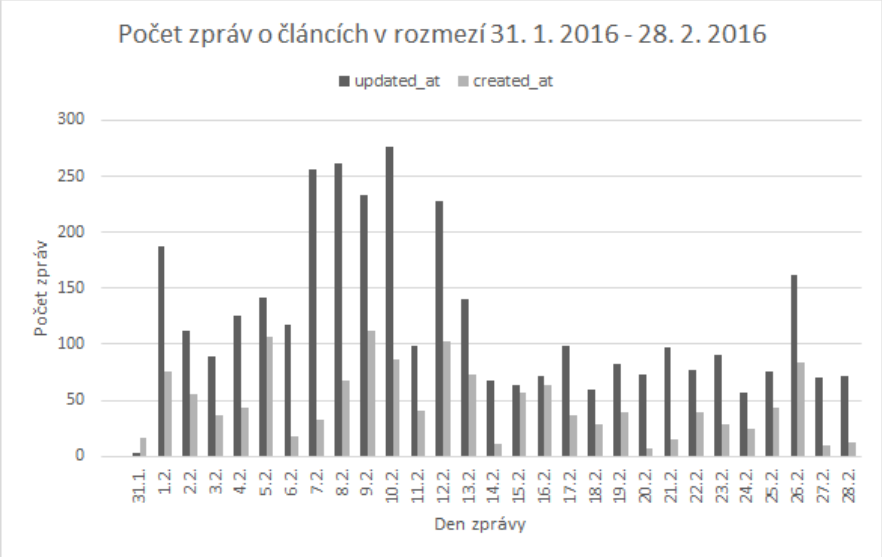
Položky mají údaj vytvoření článku (atribut *created_at*) z rozmezí 26. 8. 2001 - 28. 2. 2016. Široký rozsah této hodnoty může být vysvětlen například opravami vydaných článků, publikováním článků na kterých bylo pracováno delší dobu nebo zakázáním doporučení článků. Před únorem bylo vytvořeno 513 zpráv a z toho týden před únorem bylo vytvořeno 25 zpráv. Celkově tyto zprávy odpovídají 7 % všech transakcí. Během února bylo vydáno 557 zpráv, což tvořilo 39 % všech transakcí. Transakcemi jsou myšleny záznamy ve vstupním souboru s transakcemi.

Druhou časovou značkou je poslední aktualizace, která odpovídá času, kdy je zpráva zaznamenána. Tento údaj je z rozmezí 31. 1. 2016 06:10 - 28. 2. 2016 12:59, což je o 17 hodin dříve než začátek transakcí. Toto lze zdůvodnit počátečními informacemi pro doporučování. Následující graf 6.2 ukazuje, kolik článků bylo vytvořeno nebo aktualizováno během jednoho dne z daného období.

Z grafu je zřejmý nižší počet vytvořených zpráv během víkendů. Velký počet článků v rozmezí 7. až 10. února je dán především tradičním karnevalem v Kolíně, který se měl konat 8. února, ale nakonec byl zrušen kvůli špatnému počasí. Vzhledem k tomu, že se jedná o velkou událost, je zde větší počet aktualizovaných zpráv i o víkendu.

Záznam o položce má atribut flag, který definuje masku dostupnosti položky v doporučeních. Položky nejsou doporučovány, pokud je hodnota masky různá od nuly. Této hodnoty nabývá atribut pro 52 % článků, z čehož vyplývá, že skoro polovina položek, o kterých je informace dostupná, nemají být doporučovány. Pouze 4 položky změní svou masku během naměřené doby. Z analýzy dat vyplývá, že položky s nenulovou maskou tvoří 8 % všech transakcí. Položka může být pro uživatele zajímavá, i když už není doporučována a dle pravidel protokolu by již ani neměla být doporučována. Těchto 8 % je pro doporučovací algoritmus nedosažitelných.

Obrázek 6.2: Počet vydaných zpráv za den



6.3.3 Analýza transakcí

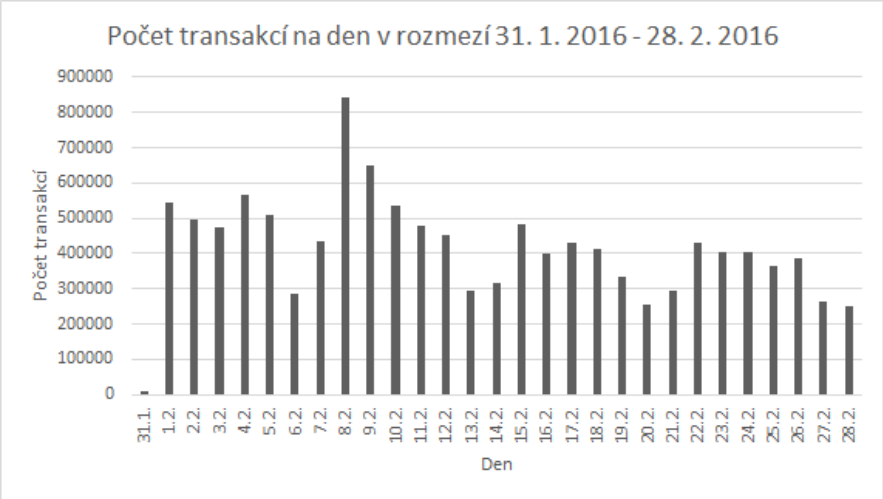
Transakce mají časové razítko z rozmezí 31. 1. 2016 23:00 - 28. 2. 2016 22:59. Celkově bylo během měsíce zaznamenáno 120 milionů transakcí. Rozdělení transakcí do dnů je možné vidět v grafu 6.3. Z rozsahu je zřejmý nižší počet transakcí během víkendů a mírně zvýšený počet transakcí v pondělí. Velký počet transakcí dne 8.2. vychází na den karnevalu. V době karnevalu vycházely články monitorující průběh události. Lidé měli motivaci číst si článek i opakovaně, neboť součástí článků jsou galerie obrázků. Karneval byl nakonec kvůli špatnému počasí zrušen. Tato skutečnost zvýšila zájem o články, neboť uživatelé chtěli zjistit více detailů.

Dataset obsahuje 81 % transakcí, které mají identifikaci položky i uživatele. Transakce, které nemají vyplněný identifikátor uživatele, jsou považovány za anonymní a tvoří něco málo přes 18 % všech transakcí. Tyto transakce je možné využít během trénování, ale není možné je využít k testování a výpočtu metrik. Transakce nemající uvedený identifikátor položky, nemohou být spojeny s žádným doporučením, a proto budou odebrány. Dané transakce tvoří 0,2 % všech transakcí.

V tabulce 6.2 jsou uvedeny základní statistiky o atributech v transakcích. Pro každý atribut je uveden výskyt v transakcích, počet unikátních hodnot a relativní četnost nejčetnější hodnoty. Uvedené statistiky lze dále využít při doporučování.

Atributy odhadu věku, pohlaví a příjmu nebyly ani jednou vyplněny. Z tohoto hlediska nepřináší žádnou informaci. Zbylé atributy byly vyplněny skoro u všech transakcí. U jazyku lze předpokládat, že nejdominantnějším jazykem

Obrázek 6.3: Počet transakcí za den



Tabulka 6.2: Přehled metadat transakcí

Atribut	**Podíl vyplněnosti**	**Počet unikátních hodnot**	**Podíl nejčastější hodnoty**
Limit	0,985	90	0,58
Kategorie	0,99992	308	0,20
Odhad příjmové skupiny	0	1	0
Odhad věkové kategorie	0	1	0
Odhad pohlaví	0	1	0
Geolokace	1	1906	0,60
PSČ	1	22814	0,13
Druh zařízení	1	5	0,73
Prohlížeč	1	435	0,18
OS	0.99998	198	0,42
Jazyk	1	98	0,93

je němčina, neboť se jedná o německý portál cílící především na německy mluvící občany [10]. Vzhledem k chybějícím překladovým tabulkám pro jednotlivé hodnoty nelze přesně interpretovat jakékoliv další hodnoty.

Výše zmíněné atributy mohou být použity pro správné doporučování anonymním uživatelům. I když uživatel nemá uvedený identifikátor uživatele, je možné na základě hodnot těchto atributů udělat otisk uživatele. Vybrané hodnoty jsou relativně stabilní a během jedné session by se neměly měnit. Na základě časové souslednosti lze danému otisku vytvořit dočasný profil a pro něj přizpůsobit doporučení. Vzhledem k 18 % anonymních transakcí je důležité

mít nástroj pro správné doporučení anonymním uživatelům.

Atribut limit definuje, kolik položek je zobrazeno při doporučení. V 58 % případů byla vyžádána jen jedna položka a v 38 % bylo vyžádáno 6 položek. Některé transakce nabývají hodnoty nula (1,5 %), což lze vysvětlit jako nezájem o doporučení. Tyto transakce přinášejí informaci o návštěvě uživatele položky, avšak nedávají prostor k pokračování session.

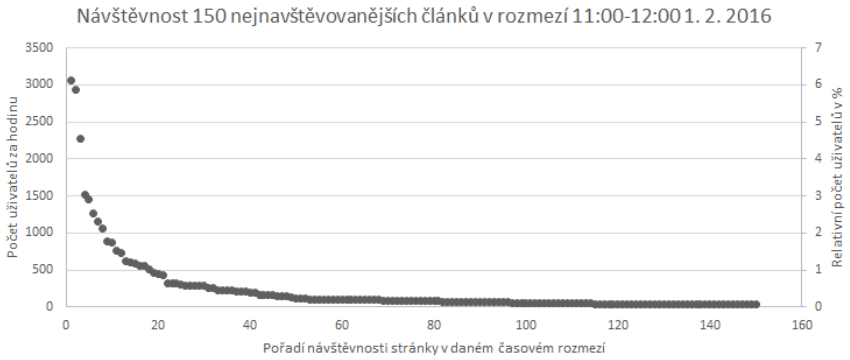
Počet doporučených položek významně ovlivňuje úspěšnost doporučení. Čím vyšší je počet doporučení, tím vyšší je pravděpodobnost výběru nějaké položky, ale zároveň se zvyšuje riziko, že uživatele nezaujme žádné doporučení nebo provede chybnou volbu. V případě dobrovolného doporučování, kdy uživatel nemusí vybrat žádnou položku je časté, že uživatel položku nevybere. Proto je důležité, aby položek bylo málo, aby nad výběrem položky nemusel dlouho přemýšlet.

Výběr zařízení, na kterým je doporučování zobrazeno může ovlivnit počet doporučených položek. Čím je zařízení menší, tím je obvykle méně doporučených položek. Z toho vyplývá zvýšený tlak na kvalitu doporučování u malých zařízení.

6.3.4 Řídkost dat

Doporučovací systémy obvykle mají řídké matice reprezentující interakci mezi položkami a uživateli. V tomto případě se jedná o vztah mezi uživateli a novinovými články. Za předpokladu, že uživatel čte jen aktuální články byl vybrán jen krátký interval datového toku.

Obrázek 6.4: Návštěvnost nejnavštěvovanějších článků během hodiny



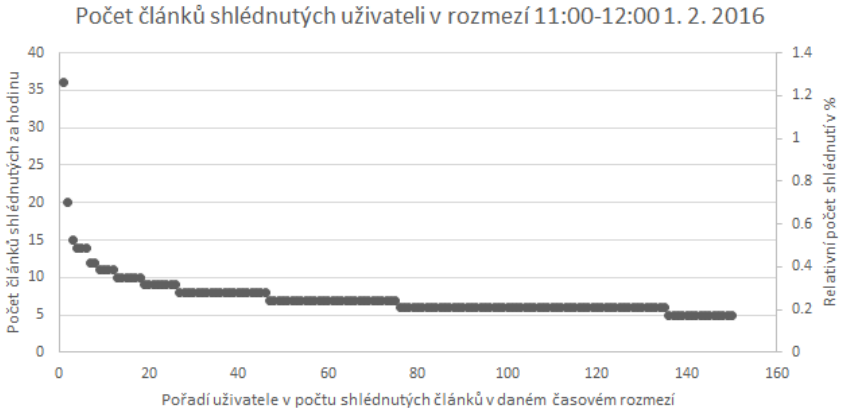
Z grafu 6.4 je zřejmé, že tento jev je patrný i v této doméně. To znamená, že je malé množství článků, které dosahují velkých počtů zhlédnutí, zatímco zbylé články zhlédne jen malé množství uživatelů. V intervalu 11:00-12:00 1. 2. 2016 měla nejnavštěvovanější stránka 6,4 % všech transakcí z dané hodiny a 5 článků tvořilo 25 % všech transakcí. Polovina všech interakcí náleží 24 článkům. Pro transakce za celý měsíc platí, že 74 nejnavštěvovanějších

článků tvoří celkem 25 % všech transakcí. Čím kratší je časový interval, tím výraznější je podíl nejnavštěvovanějších článků.

Medián návštěvnosti článků je 4, což znamená, že v celém toku je polovina transakcí, které mají nejvýše 4 zhlédnutí. 25 % nejnavštěvovanějších položek tvoří 96 % všech transakcí. Kdyby uživatelům byl po celý měsíc doporučován jen celkově nejnavštěvovanější článek, tak by úspěšnost byla 1,76 %.

Obdobný charakter má i graf 6.5 s délkou session v jednu konkrétní hodinu. Existuje pár uživatelů, kteří jsou výrazně aktivní při prohlížení článků, avšak výrazná většina uživatelů zhlédne výrazně méně článků. Více jak polovina uživatelů zhlédne jen jednu zprávu denně. Průměrně uživatel vidí 1-2 články denně. Maximální délka session na den je 95 článků.

Obrázek 6.5: Počet článků zhlédnutých uživateli s nejvyšším počtem zhlédnutí článků během hodiny

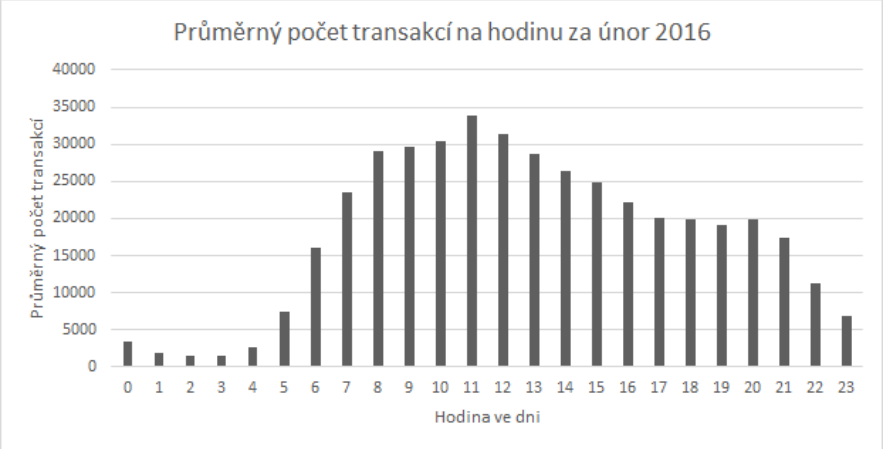


6.3.5 Vývoj dat během dne

Počet transakcí se během dne liší. V noci je obvykle provoz menší, neboť většina lidí v tuto dobu spí. Vzhledem k celosvětovému dostupnosti článků je možné, že část transakcí pochází z jiné časové zóny. Od 6 do 22 hodin je provoz výrazně vyšší, se špičkou kolem poledne, kdy si hodně lidí dopřeje klidný odpočinek a dostane se k přečtení zpráv. V grafu 6.6 je vidět detailnější přehled průměrného počtu transakcí na danou hodinu.

Ranní zprávy dosáhnou svého maxima nejdéle během oběda a postupně klesají. Pokud má zpráva dostatečnou atraktivitu, tak je navštěvována 2, případně i 3 dny. Z hlediska transakcí však valná většina transakcí proběhne během prvního dne a v následujících dnech je počet zanedbatelný. Tyto zprávy získají během 5 hodin od první transakce alespoň 50 % všech transakcí. Články, které mají nižší aktivitu mohou mít všechna zhlédnutí během jednoho dne, případně během pár hodin.

Obrázek 6.6: Průměrný počet transakcí na hodinu



Výjimkou je zpráva *245942593*, která byla průběžně čtená celý měsíc a stala se druhou nejčtenější zprávou za měsíc únor. Jedná se o článek popisující historii 100 let karnevalu v Kolíně. Vzhledem k významu tradice pro místní komunitu a obecnému ražení článku je pochopitelné, že je tento článek čtený výrazně déle.

Články zveřejněné v druhé polovině dne jsou omezeny nižším počtem uživatelů v den vydání. Na konci dne mají obvykle alespoň 50 % své popularity, kdy velkou pozornost má článek ještě celý následující den. Sice se nejedná o tak vysokou četnost transakcí, ale za to se transakce rozprostřou do celého dne. Samozřejmě atraktivitě článku odpovídá i počet zhlédnutí následující den.

Články jsou zveřejňovány a aktualizovány pouze v první polovině dne. Polovina článků je zveřejněna do 6 hodin ráno. Další čtvrtina článků je zveřejněna do 10 hodin a zbytek je zveřejněn do poledne. Z toho vyplývá, že odpolední špička je podpořena třemi vlnami zpráv, které přicházejí relativně krátce po sobě. V odpoledních hodinách již nové články nevycházejí, ale dostávají se do povědomí články, které byly zveřejněny kolem poledne. Ranní články v tu dobu již ztrácejí svou popularitu.

Část IV

Implementace

Kapitola 7

Platforma

V této kapitole jsou popsány nedostatky platformy a její změny, které jsou otestovány pomocí frameworku JUnit a zadokumentovány prostřednictvím javadoc komentářů. Platforma v současnosti neumí pracovat se všemi informacemi, které jsou v datasetu poskytnuty. Před samotným experimentováním je důležité tyto aspekty zvážit a rozhodnout, jak s nimi bude naloženo. Původní návrh platformy neumí zpracovat následující vlastnosti:

- limitování počtu doporučených položek v závislosti na požadavku;
- informace o aktualizování položky;
- metadata transakce;
- anonymní uživatele.

Informace o počtu doporučení není součástí rozhraní platformy. Tato hodnota je ale důležitá pro porovnání s výsledky z již proběhlých soutěží. Z tohoto důvodu je zvolena hodnota limitu 1. Ve 42 % případech se jedná o přísnější podmínky. Tímto rozhodnutím je zaručena porovnatelnost výsledků. Výsledky během experimentování budou penalizovány, nikoliv zvýhodněny. Medián atributu limit je jeden článek na doporučení. Není vhodné zvolit průměr, který je 3 články, neboť by výsledky bylo obtížné porovnávat.

Položky jsou průběžně aktualizovány což může ovlivnit atraktivitu položky. Z analýzy dat vyplývá, že datový tok obsahuje popis pouze 0,6 % položek, které jsou referencovány v datovém toku a tvoří 46 % všech transakcí. Zbylých položek je výrazně více a tvoří více jak polovinu všech transakcí. Položky musí být doručovány i bez znalosti jejich obsahu. U některých položek existují transakce dříve, než došlo k přijetí informace o vydání položky. Z výše vypsaných důvodů je daná limitace akceptována s tím, že má malý vliv na doporučování.

Současná platforma nepodporuje žádná metadata transakce. Uvedená metadata jsou anonymizována, takže jejich význam není znám. Hodnoty by

bylo možné využít k identifikaci anonymních uživatelů nebo ke shlukování požadavků. Identifikace anonymních uživatelů není obsahem této práce. Shlukování požadavků by bylo možné využít ke zlepšení doporučování. V této práci je tato limitace akceptována a navržení kandidáti nebudou brát na metadata v dotazu ohled. Anonymní uživatele je možné použít k trénování modelu, ale nejsou použiti k určení přesnosti.

7.1 Změny v implementaci

Původní verze platformy umožňuje transformaci dat ze soutěže CLEF NewsREEL challenge na data, která jsou použitelná platformou. Tato data mohou být filtrovaná dle vydavatele. Všechny transakce, které nemají vyplněnou kategorii, identifikátor uživatele nebo identifikátor položky jsou odstraněny. Toto je v kódu změněno. Během experimentů může kandidát využít anonymní uživatele k učení a vyplnění kategorie není vyžadováno.

Platforma neví nic o anonymních uživatelích. Každý uživatel je rozpoznán na základě unikátního identifikátoru, který mají všichni anonymní uživatelé stejný. To má za následek, že session anonymních uživatelů obsahuje všechny stránky navštívené anonymními uživateli. Tento fakt povede k nesprávným výsledkům při doporučení. Jak metriky, tak algoritmy, mohou být ovlivněny faktem, že se jedná o anonymního uživatele.

Platforma je proto modifikována. Anonymní sessions mají velikost 1. Filtrování sessions podle velikosti ignoruje anonymní sessions. Algoritmy a metriky již dostávají informaci o transakci, ve které je uveden uživatel. Na základě objektu *Transaction*, který přichází do metod jako argument, se mohou algoritmy rozhodnout, jak se chtějí s anonymními uživateli vypořádat.

Původní implementace platformy neobsahuje žádné testy. Vzhledem k tomu, že je platforma rozšířena o novou funkcionalitu, byla logika z třídy *StreamingRec* extrahována do dvou nových tříd *DataManager* a *WorkPackageFactory*. První třída má na starost správné rozdělení vstupních dat na testovací a trénovací. Druhá třída vytváří objekt, jehož atributy jsou použity jako parametry pro doporučování, trénování a vyhodnocení metrik. Po refaktorování byly tyto třídy otestovány. Po rozšíření funkcionality vznikly nové testy, které verifikovaly funkčnost přidané funkcionality.

Kapitola 8

Streaming algoritmus

V rámci platformy je naimplementován algoritmus *Streaming*, který zapouzdřuje logiku různých technik dolování dat z datového toku. Vzhledem k rozsáhlé parametrizaci byl vytvořen builder *StreamingBuilder*, který všechny parametry přeloží na správné objekty. Výsledkem stavitele je třída *StreamingManager*, která obsahuje konkrétní kombinace technik. Tyto techniky jsou v rámci algoritmu *Streaming* volány při učení a doporučování.

Příklad propojení konkrétních instancí technik je možné vidět v diagramu 8.1. Jednotlivé komponenty jsou propojené prostřednictvím interfaců. To zaručuje testovatelnost, rozšířitelnost a menší provázanost kódu. V této podkapitole budou popsány jednotlivé komponenty.

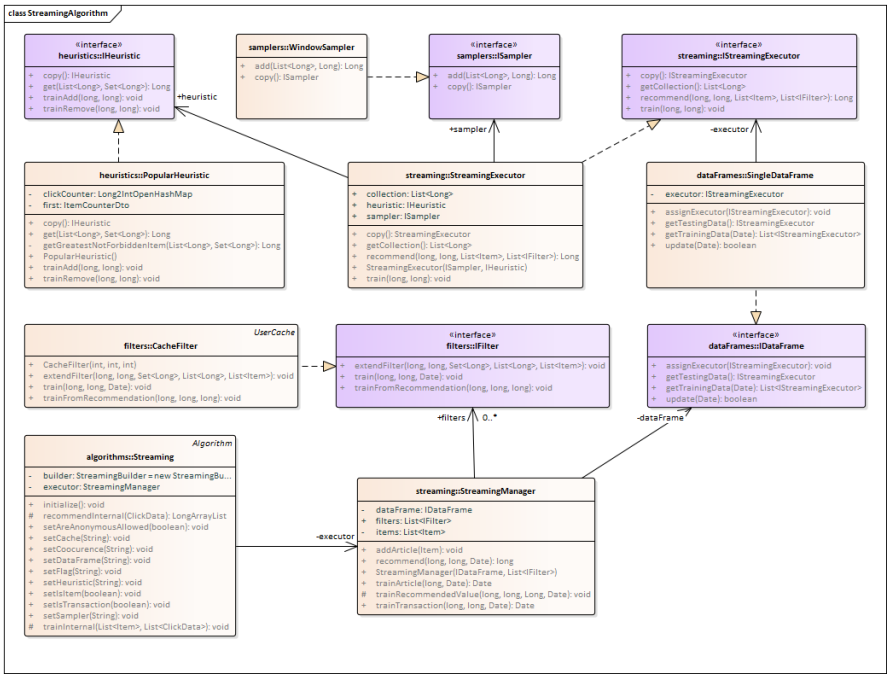
8.1 StreamingBuilder

Tato třída je částí návrhové vzoru builder. Jedná se o část samotného vytváření objektů a skládání výsledné instance *StreamingManager*. Řídícím prvek návrhového vzoru je samotná třída *Streaming*, která získává parametry z konfiguračního souboru a předává je staviteli. Aby bylo možné kombinovat různé parametry, jsou techniky a parametry kódované v řetězci. Úkolem stavitele je řetězec dekódovat a získat parametry, které jsou použity v konstruktoru. Stavitel si udržuje informaci o používání anonymních uživatelů k trénování, zpracování položek a zpracování transakcí.

8.2 StreamingManager

Tato třída zapouzdřuje logiku exekuce a funguje jako návrhový vzor template. Sdružuje sdílené techniky mezi modely *IDataFrame* a kolekci instancí typu *IFilter*. V rámci doporučování a učení jsou volané jednotlivé metody technik, aby došlo k jejich učení a zároveň získání správného doporučení.

Obrázek 8.1: Provázanost tříd a interfaců jednoho z kandidátů



8.3 IFilter

Tento interface zaštiťuje všechny filtrace, které mohou zúžit výběr transakcí, ze kterých je vybíráno. Soubor transakcí může být omezen výběrem položek, ze kterých bude heuristika vybírat a zároveň může definovat položky, které nesmějí být vybrány. Filtry mají separátní cyklus učení a resetování od *IDataFrame*. K učení může docházet v době učení nebo na základě doporučené položky. Tento interface implementují třídy: *CacheFilter*, *Coocurent-Filter* a *FlagFilter*.

Třída *CacheFilter* určuje jaké položky nemohou být vybrány na základě historie uživatele. Cache je definována velikostí, hloubkou a časem expirace záznamů. Za použití hashování lze výrazně zmenšit prostor identifikátorů uživatelů a získat krátkodobou historii session uživatele. Vzhledem ke krátkým sessions uživatelů, je možné tyto záznamy relativně často anulovat a i přesto získat dobrý mechanismus na zvýšení přesnosti.

Druhý filtr si udržuje záznam o propojení stránek. Při učení jsou zapamatovány přechody mezi jednotlivými články. V případě příchodu uživatele na článek je možné na základě znalosti davu rozhodnout, kam uživatelé přecházeli z dané stránky. Vzhledem k rychlému růstu paměťové náročnosti je struktura validní jen po omezenou dobu.

Poslední implementovaný filter vychází z omezení protokolu. Ten předpokládá, že položky, které mají vyplněný nenulový atribut flag, již nejsou do-

poručovány. Na základě analýzy dat je zřejmé, že články, které by neměly být doporučovány, jsou navštěvovány. Pomocí tohoto filtru je možné vyzkoušet, jaký vliv má daný filtr na měřené metriky. U položek, které nemají informaci o atributu flag, je předpokládáno, že mohou být doporučeny.

8.4 IDataFrame

Tento interface zastřešuje operace, které mohou být provedeny nad kolekcemi modelů. Model je reprezentován interfacem *IStreamingExecutor*. Kolekce modelů může být použita k paralelnímu nebo oddělenému procesu učení a doporučování. Správou vícero modelů je dosažena adaptivita systému. Třídy realizující tento interface jsou: *SingleDataFrame*, *OverlappingDataFrame* a *SeparateDataFrame*.

Třída *SingleDataFrame* obsahuje pouze jeden model. Na doporučování a učení se tím pádem využívá stejná kolekce. Tato implementace nemá žádné adaptivní metody.

Třída *OverlappingDataFrame* používá dva částečně se překrývající modely. Délka překryvu je parametrizovatelná. Velikost modelu je dána okénkem, a to buď v počtu transakcí nebo času. Velikost okének je dána sekvencí velikostí.

Třída *SeparateDataFrame* obsahuje dva modely. Po dobu testování je jeden model používán na doporučování a druhý model na trénování. Po uplynutí okénka je trénovací okénko použito pro doporučování a vzniká nové okénko na trénování. Tento princip lze využít u dvoufázových proudových algoritmů. Doporučování a trénování probíhá nad separovanými modely, a tak mohou být akce paralelizovány. Parametrem algoritmu je sekvence velikostí okének.

8.5 IStreamingExecutor

Tento interface má pouze jednu implementaci. Jedná se o reprezentaci jednoho modelu a dat z datového toku. Třída má reference na interfaces: *IHeuristic* a *ISampler*. *IDataFrame* rozhoduje jaké modely jsou trénované a jaké jsou použity na doporučování. Model samotný se stará o správu dat a udržování heuristiky připravené k doporučování. Důležitou metodou je klonování exekutora, které umožňuje použití nového modelu se stejnými vlastnostmi.

8.6 ISampler

Tento interface reprezentuje logiku správy datasetu. Dataset je pouhá kolekce položek. V případě příchodu nové položky je na implementaci, jak naloží s nově příchozí položkou a jestli nějak změní původní kolekci. Implementace umožňují klonování instancí. Tento interface je implementován: *AbstractReservoirSampler*, *FloatingWindowSampler* a *WindowSampler*.

První zmíněný je implementace vzorkovacího rezervoáru. Základem je kolekce fixní velikosti, která reprezentuje vzorek datového toku. Dokud tato kolekce není plná, je obsazována příchozími položkami z datového toku. Když je kolekce naplněna, každá další příchozí položka nahradí položku v rezervoáru na základě stanovené pravděpodobnosti.

Pravděpodobnost náhrady může být fixní (odpovídá třídě *FixedReservoirSampling*) nebo se může snižovat s počtem příchozích položek (odpovídá třídě *DynamicReservoirSampling*). Volba nahrazovací metody ovlivňuje adaptabilitu modelu. Pokud se pravděpodobnost náhrady postupně snižuje, klesá schopnost adaptability novým změnám, ale zachovává se reprezentativnost delšího úseku datového toku.

Druhá implementace je plovoucí okénko. Základem je okénko fixní velikosti. Dokud kapacita není naplněna jsou položky přidávány principem FIFO. V okamžiku naplnění kapacity je nejstarší transakce odebrána a nový prvek přidán na začátek. Velikost okénka definuje, po kolika prvcích dochází k zapomínání transakcí.

Poslední implementace interfacu je rostoucí okénko. Každá příchozí položka se přidá na konec okénka a nedochází k žádnému odebírání. Tato technika neimplementuje adaptaci toku. Pokud algoritmus *Streaming* neobsahuje jinou adaptivní technikou, tak dojde k postupnému zapamatování celého datového toku, dokud nedojde k ukončení toku nebo nedostatku zdrojů.

8.7 IHeuristic

Heuristika určuje způsob výběru položky ze vzorku datového toku. Heuristika je průběžně učena z příchozích dat. Vzhledem k různým implementacím *ISampler* musí heuristiky reagovat jak na přidání, tak na odebrání položky ze vzorku dat. Tento interface implementují: *IteratorHeuristic*, *PopularHeuristic*, *RandomHeuristic* a *RecentHeuristic*.

Iterační heuristika prochází kolekcí a vybere první vhodnou položku. Způsob průchodu je Top-down. Heuristika založená na populárnosti si vytváří strukturu návštěvnosti jednotlivých položek. V případě požadavku o doporučení vybírá položku s nejvyšší návštěvností, kterou je možné vybrat a není zakázána. Náhodná heuristika opakuje náhodný výběr položky, dokud nedojde k výběru vhodné položky. Poslední heuristikou je implementace záznamu posledních navštívených položek všemi uživateli. Heuristika vybírá nejnovější článek, který byl nedávno navštíven a zároveň je možné ho doporučit.

8.8 Metriky

Během experimentů budou vyhodnoceny následující metriky: offline CTR, průměrná doba odpovědi a katalogové pokrytí. První zmíněnou metrikou je of-

fline CTR, kterou lze chápat také jako predikční přesnost (anglicky *prediction accuracy*). Vztah popisuje vzoreček (8.1).

$$accuracy = \frac{number\ of\ true\ predicted}{number\ of\ predictions} \tag{8.1}$$

Doporučovací algoritmus se snaží doporučit všechny položky, které uživatel v rámci jedné session navštívil. Vzhledem k doporučení pouze jedné položky se jedná o predikci položky, na kterou uživatel v rámci session klikl. Pokud je metrika rovna nule, nepodařilo se předpovědět ani jednu transakci. V případě, že je metrika rovna jedné, podařilo se predikovat všechny transakce. Metrika je implementována třídou *Accuracy*.

V soutěži byl požadavek, aby doporučení bylo odesláno během 100 ms. V online prostředí může ke kolekcím přistupovat více vláken zároveň a může dojít k prodloužení délky odezvy, kvůli sdíleným prostředkům. Během offline testování jsou záznamy zpracovány sekvenčně. Nedochází zde ke zpomalení způsobené více vlákny. Proto budou využity již implementované metriky *Runtime*, které umožňují monitorování času trénování a doporučování položek.

Poslední metrikou je katalogové pokrytí spektra zpráv. Dříve již bylo zmíněno, že většina uživatelů navštíví jen na pár vybraných zpráv. Důsledkem toho je celá matice zhlédnutí zpráv uživateli řídká. Záznam datového toku popisuje 159 tisíc položek a 75 (0,04 %) nejčtenějších zpráv tvoří 25 % všech transakcí. Polovinu všech transakcí tvoří 408 (0,25 %) nejnavštěvovanějších článků. Nízké katalogové pokrytí může znamenat dobré výsledky v přesnosti.

Část V

Experimentální část

Kapitola 9

Experimenty

V této kapitole jsou popsány experimenty s proudovým algoritmem využívající techniky dolování dat z datového toku. Nejprve jsou provedeny testy na základních heuristikách bez dodatečných technik. Na jejich základě jsou provedeny testy na algoritmu *Streaming*.

Vzhledem k obrovskému stavovému prostoru není možné vyzkoušet všechny stavy algoritmu. Z toho důvodu je důležité stavový prostor procházet postupně. Experimenty jsou proto rozděleny do skupin dle zvolené heuristiky. Zvolenou heuristikou průchodu stavového prostoru je hladové procházení, jejímž hlavním měřítkem kvality je přesnost doporučení. Při výběru jsou však brány v potaz časové a paměťové nároky. Zvolený stav nemusí mít nejvyšší přesnost doporučení, ale může mít malou velikost paměti a časovou náročnost pro dostatečně vysokou přesnost. Dalším zvýšením paměťových nebo časových nároků by bylo dosaženo jen nepatrného zvýšení přesnosti.

Všichni zkoumaní kandidáti měli průměrnou délku odpovědi pod hranici 0,01 ms na požadavek. Variance hodnot byla nízká a proto nejsou u experimentů tyto hodnoty uvedeny. Paměťovou náročnost ovlivňují použité techniky jejichž velikost lze řídit parametry. U kandidátů je možné určit kolik paměti potřebují.

9.1 Nastavení

Experimenty byly provedeny na počítači s dvoujádrovým procesorem Intel(R) Core(TM) i7-7500 a pamětí 16 GB. Celé testování probíhalo na operačním systému Windows 10 Pro verze 17134. JVM měla omezení paměti na 14 GB.

Evaluace algoritmů byla provedena na datech z prvního týdne datového toku. Nejlepší kandidát z každé kategorie byl následně otestován na celém měsíci. Některé techniky využívají randomizace, což vede k nestabilním výsledkům. Z toho důvodu jsou pozorování opakovány 5x, aby bylo možné říci, jestli jsou hodnoty stabilní. Výsledkem je průměrná hodnota měření.

Platformu je možné nastavit pomocí parametrů. Aby bylo možné měření opakovat, jsou tyto parametry vypsány, neboť mohou mít vliv na získané výsledky. Parametry třídy *StreamingRec* mají následující hodnoty:

- velikost session je 20 minut;
  - *SESSION_TIME_THRESHOLD*;
- poměr trénovací části je jeden den;
  - *SPLIT_THRESHOLD* má hodnotu 0.0531 pro celý dataset a 0.2005 pro experimentální část datasetu (týden);
- deduplikace je aktivní;
  - *DEDUPLICATE*;
- session musí mít velikost větší než 1;
  - *SESSION_LENGTH_FILTER*.

Deduplikaci je vhodné použít pro zvolenou doménu, neboť při transformaci dat na CSV soubory došlo k přeměně každé zprávy *recommendation_request* a *event_notification* na stejný záznam transakce. Popis transformace dat je popsán v kapitole 6.2. Tyto zprávy se vyskytují ve dvojicích, jak již bylo popsáno v kapitole 4.4. Aplikací deduplikace dojde k odstranění pozdější zprávy a nedojde ke ztrátě informace.

Proudové algoritmy začínají v počáteční konfiguraci, která má určité parametry sdílené napříč heuristikami. Některé z těchto parametrů jsou v podkapitolách prozkoumány a jejich vlivy na metriky jsou popsány. Parametry mají následující hodnoty:

- heuristika dle kapitoly;
- anonymní uživatelé jsou použiti na trénování;
- pouze transakce jsou použity na trénování;
- žádné filtry nejsou aktivovány.

9.2 Pouze heuristiky

Tyto algoritmy jsou jednoduché a stanoví základní kvality heuristik. V případě použití těchto heuristik v proudových algoritmech, bude sledováno, jak se mění vlastnosti a jaký vliv mají na přesnost, katalogové pokrytí, rychlost a paměťovou náročnost. Mezi základní heuristiky se řadí náhodný výběr, poslední

Tabulka 9.1: Naměřené metriky pouze heuristik

Heuristika	**Přesnost**	**Katalogové pokrytí**
Náhodné generování čísla	0 %	0,05 %
Náhodný výběr z existujících transakcí	0 %	96 %
Poslední publikovaný článek	0,3 %	0 %
Poslední navštívený článek	2,5 %	100 %
Nejnavštěvovanější článek	0,8 %	0 %

navštívený článek a nejpopulárnější článek. Konfigurace algoritmů je v souboru *algorithm-config-heuristics.json.*Výsledky jednotlivých algoritmů jsou shrnuty v tabulce 9.1.

Náhodný výběr může ignorovat příchozí položky a na dotaz o doporučení pouze vygenerovat náhodné číslo. I když je celková náročnost algoritmu nízká, přesnost je skoro nulová, neboť pravděpodobnost uhodnutí správného čísla je velmi nízká. Z toho důvodu je zde ještě druhá implementace, která náhodně vybírá položku z již proběhlých transakcí. Za cenu více paměti bylo dosaženo mírného zlepšení v přesnosti. Paměť je omezena počtem zpracovaných položek $\mathcal{O}(number\ of\ items)$.

Heuristika posledního navštíveného článku ukazuje více na chování uživatelů. Často navštěvovaný článek bude častěji doporučovaný a současně nově příchozí články mohou být dále doporučeny, pokud existuje někdo, kdo si takový článek již přečetl. Paměťová i časová náročnost je nízká $\mathcal{O}(1)$. Přesnost je výrazně lepší než u předchozích algoritmů, a katalogové pokrytí je 100% neboť doporučuje všechny položky, které byly navštíveny. V porovnání s výsledky ze soutěže se dokonce jedná o velmi dobrý výsledek.

Poslední heuristikou je doporučení celkově nejnavštěvovanějšího článku. Tento algoritmus nefunguje dobře pro nově příchozí položky. Pokud je článek populární, pravděpodobně už není pro mnoho uživatelů nový a zajímavý. Algoritmus si navíc musí držet informaci o všech položkách a četnostech $\mathcal{O}(number\ of\ items)$. Při učení modelu může dojít k aktualizaci nejpopulárnějšího článku. Z toho důvodu je odpověď na požadavek rychlá. Nejpopulárnější článek je doporučován, dokud není pokořen jiným článkem. To může způsobit doporučení článku i několik dní po jeho vydání, kdy už článek není pro čtenáře zajímavý. Z toho důvodu je výsledná přesnost nízká.

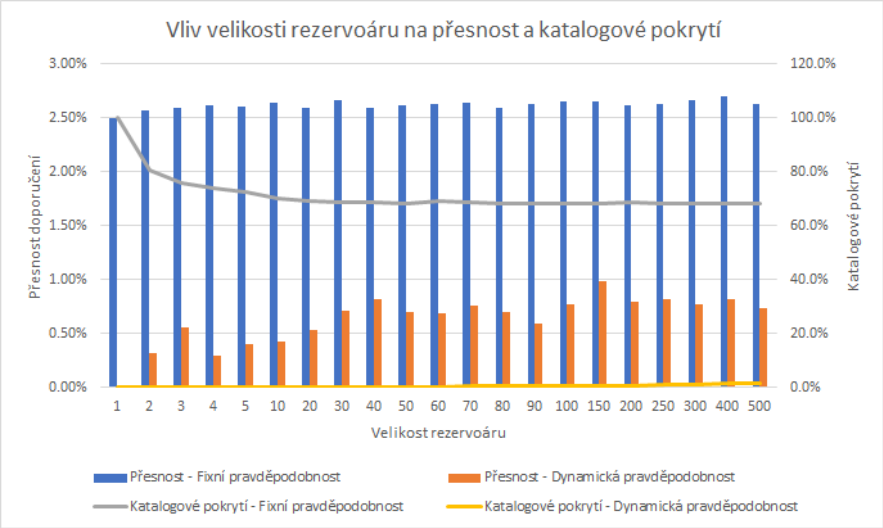
9.3 Náhodný výběr

Experimenty popsané v této podkapitole jsou založeny na heuristice náhodného výběru. Algoritmus vybere náhodný prvek z položek, které mohou být vybrány. Výchozí konfigurace algoritmu je následující:

- DataFrame = single;
- Sampler = reservoir;
  - Offset = 0;
  - Size = 1;
  - Mode = fixed.

Tato konfigurace dosáhla přesnosti 2,5 % s pokrytím 100 %, jedná se o stejný přístup jako v případě heuristiky doporučení posledního čteného článku. Při optimalizaci parametrů třídy *ISampler* byly zkoušeny různé velikosti offsetu, velikosti rezervoáru a mody adaptace. Graf 9.1 popisuje vzájemný vztah těchto veličin. Zvětšením velikosti lze mírně zvýšit přesnost za cenu nižšího katalogového pokrytí. To je dáno náhodností výběru a přepisu položek v rezervoáru. Málo časté položky mohou být zapomenuty a nikdy nedoporučeny. Dynamický mód má stejně jako zvýšený offset za následek nízkou adaptivitu algoritmu. Čím nižší je pravděpodobnost nahrazení, tím horší je dosažená přesnost a katalogové pokrytí.

Obrázek 9.1: Vliv velikosti rezervoáru u heuristiky náhodného výběru



Nahrazení rezervoáru plovoucím okénkem fixní velikosti nedošlo ke zvýšení přesnosti. Došlo však ke zvýšení katalogového pokrytí. Vzhledem ke stejným paměťovým nárokům a mírně zvýšené časové náročnosti se jedná o dobrou alternativu.

Velikost cache byla zkoumána ve 3 dimenzích: čas expirace, velikosti paměti a hloubka paměti. Čas expirace a hloubka paměti neměli pozitivní vliv na metriky. Velikost hloubky paměti vedla ke zlepšení jen pár setinek procenta. Malá

Tabulka 9.2: Vztah aspektů na měřené metriky u heuristiky náhodného výběru

Aspekty	**Vliv**	**Hodnoty**
Zvýšení offsetu	-	0-495
Zvýšení size	+	1-500
Změna mode	-	dynamický
Změna implementace ISampler	+	plovoucí okénko
Změna implementace IDataFrame	0	oddělený, překrývající
Cachování uživatelů	0	hloubka:1-2, expirace:1-5, velikost:4-4096
Filtrování na základě flagu	0	ano
Filtrování na základě koexistence	-	ano
Použití anonymních uživatelů	+	ne

velikost pole vedla k velkému množství kolizí, a tak i zhoršení přesnosti. Velikosti cache byly v mocninách 2. Od velikosti 512 již nedocházelo ke kolizím a přesnost byla zachována. Aplikací cachování došlo ke zvýšení katalogového pokrytí průměrně o procento. To však za cenu cache pro uživatele není dostatečně dobrý výsledek.

Posledním zkoumaným aspektem bylo vynechání anonymních uživatelů při trénování. Došlo tím ke zvýšení přesnosti na 2,9 % za cenu výrazného snížení průměrného katalogového pokrytí na 50 %. To je stále obstojná hodnota. Zbylé zkoumané vlastnosti jsou uvedeny v tabulce 9.2. Celkově lze říci, že tato heuristika má svůj strop přesnosti vcelku nízko, neboť většina technik měla jen malou pozitivní odezvu v získané přesnosti. Vzhledem ke zvoleným technikám je paměťová náročnost limitována velikostí fronty $\mathcal{O}(sampler\ size)$. Koncové nastavení kandidáta je následující:

- DataFrame = single;
- Sampler = floating;
  - size = 5;
- areAnonymousAllowed = false.

9.4 Populárnost

Experimenty v této podkapitole jsou založeny na heuristice populární položky. Základem je struktura, reprezentující počet návštěv jednotlivých článků.

V případě přidání je počítadlo zvýšeno a v případě odebrání je sníženo. Pro doporučení se vybere položka, která je mezi povolenými a zároveň má nejvyšší návštěvnost. Počáteční nastavení algoritmu je následující:

- DataFrame = overlap;
  - mode = count;
  - frames = 100;
  - size = 50;
- Sampler = list.

Obrázek 9.2: Vztah velikosti okénka a délkou překryvu v počtu transakcí u heuristiky populární položky

Velikost překryvu	Velikost okénka													
	10	30	50	70	90	110	120	130	150	175	200	300	250	300
1	3.06%	4.31%	4.77%	5.05%	5.20%	5.30%	5.28%	5.39%	5.39%	5.41%	5.45%	5.43%	5.51%	5.43%
10	3.12%	4.78%	5.15%	5.30%	5.43%	5.47%	5.45%	5.49%	5.51%	5.58%	5.53%	5.48%	5.51%	5.48%
20		4.91%	5.38%	5.48%	5.56%	5.57%	5.59%	5.57%	5.57%	5.64%	5.59%	5.50%	5.58%	5.50%
30		4.33%	5.45%	5.61%	5.63%	5.64%	5.61%	5.63%	5.64%	5.62%	5.60%	5.57%	5.57%	5.57%
40			5.24%	5.63%	5.65%	5.65%	5.64%	5.61%	5.64%	5.62%	5.59%	5.55%	5.55%	5.55%
50			4.82%	5.56%	5.64%	5.69%	5.64%	5.64%	5.66%	5.67%	5.63%	5.60%	5.61%	5.60%
60				5.40%	5.65%	5.66%	5.70%	5.65%	5.64%	5.66%	5.60%	5.54%	5.58%	5.54%
70				5.09%	5.58%	5.66%	5.68%	5.67%	5.64%	5.62%	5.62%	5.59%	5.57%	5.59%
80					5.44%	5.64%	5.67%	5.64%	5.65%	5.64%	5.61%	5.56%	5.61%	5.56%
90					5.21%	5.59%	5.65%	5.65%	5.64%	5.63%	5.60%	5.50%	5.55%	5.50%
100						5.47%	5.62%	5.63%	5.65%	5.64%	5.60%	5.54%	5.59%	5.54%
AVG	3.09%	4.58%	5.13%	5.39%	5.50%	5.58%	5.60%	5.60%	5.60%	5.61%	5.58%	5.53%	5.57%	5.53%
MAX	3.12%	4.91%	5.45%	5.63%	5.65%	5.69%	5.70%	5.67%	5.66%	5.67%	5.63%	5.60%	5.61%	5.60%

První je prozkoumán prostor velikosti okénka a překryvu (parametry *frames* a *size*). Vztah mezi těmito dimenzemi není přímočarý. Obecně lze říci, že příliš malá okénka nemají dostatek dat na správné doporučení položky. Příliš velká okénka nejsou tak adaptivní a dosahují nižších přesností. Optimální se zdá být velikost okénka 120 položek s překryvem 60 položek, což je polovina okénka. Vztah těchto dvou dimenzí je možné vidět v teplotní mapě 9.2. Katalogové pokrytí se snižuje se zvyšujícím se překryvem a velikostí okénka. Hodnoty, které dosáhly mírně lepší přesnosti dosáhly i mírně horšího katalogového pokrytí.

Vzhledem k implementaci *OverlappingDataFrame* jsou vždy dostupná pouze dvě okénka. V případě, že je překryv příliš velký, nestíhá se druhé okénko znovu naučit na nových datech, a proto se zhoršuje přesnost s vyšším překryvem. Toto potvrzují i dobré výsledky u jiných velikostí okénka s velikostí překryvu kolem poloviny velikosti okénka. Lze předpokládat, že by bylo možné dosáhnout lepších výsledků při jiné implementaci. Nevýhodou většího překryvu je velké množství současně trénovaných okének, což může zvýšit paměťovou i časovou náročnost.

V případě změny velikosti okének z počtu transakcí na čas lze dosáhnout obdobných výsledků. Více detailů je v teplotní mapě 9.3. I když přesnost je o desetinu nižší, jak v případě modu počtu transakcí, tak katalogové pokrytí

Obrázek 9.3: Vztah velikosti okénka a délkou překryvu v sekundách u heuristiky populární položky

Velikost překryvu v sekundách	Velikost okénka v sekundách										
	30	36	42	48	54	60	66	72	78	84	90
7	5.28%	5.33%	5.40%	5.42%	5.39%	5.44%	5.43%	5.46%	5.46%	5.48%	5.44%
12	5.42%	5.47%	5.46%	5.49%	5.53%	5.52%	5.49%	5.56%	5.51%	5.50%	5.50%
15	5.49%	5.51%	5.52%	5.52%	5.54%	5.57%	5.49%	5.57%	5.53%	5.53%	5.53%
18	5.53%	5.52%	5.57%	5.51%	5.58%	5.57%	5.52%	5.56%	5.57%	5.54%	5.53%
21	5.54%	5.53%	5.58%	5.55%	5.59%	5.58%	5.51%	5.57%	5.55%	5.55%	5.55%
24	5.55%	5.54%	5.58%	5.59%	5.55%	5.59%	5.53%	5.57%	5.55%	5.56%	5.55%
27	5.55%	5.54%	5.60%	5.60%	5.53%	5.57%	5.53%	5.56%	5.58%	5.57%	5.54%
30	5.55%	5.54%	5.59%	5.60%	5.57%	5.58%	5.53%	5.56%	5.56%	5.54%	5.56%
36		5.59%	5.56%	5.60%	5.58%	5.57%	5.55%	5.59%	5.57%	5.53%	5.55%
42			5.57%	5.59%	5.58%	5.55%	5.55%	5.57%	5.57%	5.55%	5.57%
48				5.56%	5.56%	5.56%	5.60%	5.57%	5.57%	5.56%	5.57%
54					5.58%	5.57%	5.58%	5.57%	5.58%	5.56%	5.57%
60						5.54%	5.57%	5.55%	5.55%	5.57%	5.53%
66							5.57%	5.55%	5.56%	5.54%	5.53%
72								5.52%	5.55%	5.52%	5.51%
AVG	5.49%	5.51%	5.54%	5.55%	5.55%	5.55%	5.53%	5.55%	5.55%	5.54%	5.54%
MAX	5.55%	5.59%	5.60%	5.60%	5.59%	5.59%	5.60%	5.59%	5.58%	5.57%	5.57%

Obrázek 9.4: Vztah velikosti cache paměti a dobou expirace záznamů u heuristiky populární položky

Doba expirace v minutách	Velikost paměti										
	4	8	16	32	64	128	256	512	1024	2048	4096
1	5.74%	5.71%	5.72%	5.72%	5.73%	5.74%	5.75%	5.76%	5.76%	5.76%	5.76%
2	5.73%	5.71%	5.72%	5.72%	5.73%	5.74%	5.76%	5.78%	5.79%	5.79%	5.79%
3	5.73%	5.71%	5.71%	5.72%	5.73%	5.75%	5.77%	5.80%	5.81%	5.82%	5.82%
4	5.74%	5.71%	5.71%	5.72%	5.72%	5.75%	5.79%	5.81%	5.82%	5.83%	5.84%
5	5.73%	5.71%	5.71%	5.72%	5.73%	5.74%	5.78%	5.81%	5.83%	5.84%	5.84%

je vyšší. Katalogové pokrytí navíc nedosahuje horších hodnot pro kombinace, které mají lepší přesnost.

Obrázek 9.5: Vztah hloubky cache paměti a dobou expirace záznamů u heuristiky populární položky

Doba expirace v minutách	Velikost historie									
	1	2	3	4	5	6	7	8	9	10
1	5.76%	6.04%	6.05%	6.08%	6.09%	6.09%	6.09%	6.09%	6.09%	6.09%
3	5.83%	6.40%	6.46%	6.54%	6.55%	6.57%	6.57%	6.58%	6.58%	6.58%
5	5.85%	6.53%	6.64%	6.75%	6.79%	6.82%	6.83%	6.84%	6.83%	6.84%
7	5.86%	6.60%	6.72%	6.85%	6.91%	6.95%	6.97%	6.98%	6.98%	6.98%
9	5.86%	6.61%	6.76%	6.90%	6.96%	7.01%	7.03%	7.04%	7.04%	7.05%
11	5.86%	6.65%	6.80%	6.97%	7.04%	7.09%	7.12%	7.13%	7.14%	7.15%
13	5.87%	6.67%	6.83%	7.00%	7.07%	7.13%	7.16%	7.18%	7.19%	7.20%
15	5.87%	6.68%	6.84%	7.00%	7.08%	7.14%	7.17%	7.19%	7.20%	7.21%
17	5.87%	6.70%	6.86%	7.02%	7.11%	7.16%	7.20%	7.21%	7.22%	7.24%
19	5.87%	6.70%	6.87%	7.04%	7.12%	7.18%	7.22%	7.24%	7.25%	7.26%
21	5.87%	6.71%	6.88%	7.04%	7.13%	7.20%	7.23%	7.25%	7.26%	7.27%

Cachování uživatelů má pozitivní vliv ve všech dimenzích. V případě malé paměti nedochází ke zhoršení přesnosti, ale také nedochází k zásadnímu zlepšení. V teplotní mapě 9.4 je vidět, jaký je vztah mezi velikostí paměti a dobou expirace při hloubce jedna. Z vývoje hodnot je zřejmé, že přesnost má tendenci se zvyšovat, čím větší je paměť a délka expirace.

Z experimentů vychází, že paměť velikosti 4096 má vyšší přesnost při větší

hloubce historie. Vztah mezi hloubkou historie a dobou expirace je možné vidět v teplotní mapě 9.5. Kromě zvýšení přesnosti dochází i ke zvyšování katalogového pokrytí. V případě, že si historie pamatuje nejvýše 6 položek po dobu alespoň 13 minut, dochází k nejoptimálnějším hodnotám. Zvětšením hloubky historie nebo doby expirace nad tyto hranice nedochází k zásadnímu zlepšení ani jedné metriky.

Tabulka 9.3: Vztah aspektů na měřené metriky u heuristiky populární položky

Aspekty	**Vliv**	**Hodnoty**
Změna frames	+	10-300
Změna time	0	1-100
Změna mode	0	mode:time, time:7-72s, frames:30-90
Změna implementace IDataFrame	-	oddělené
Změna implementace ISampler	-	plovoucí okénko a rezervoár
Změna IDataFrame a ISampler	0	oddělený a plovoucí okénko/rezervoár
Cachování uživatelů	+	hloubka:1-10, expirace:1-21, velikost:4-8192
Filtrování na základě flagu	+	ano
Filtrování na základě koexistence	-	ano
Použití anonymních uživatelů	+	ne

Filtrování dle flagu a ignorování anonymních uživatelů při trénování zvýšilo přesnost a snížilo katalogové pokrytí. Detailní přehled vlivů všech aspektů a volených hodnot je možné naleznout v tabulce 9.3. Výsledkem je dosažená přesnost 7,5 % a katalogové pokrytí 2,6 %. Výsledné nastavení kandidáta je následující:

- DataFrame = overlap;
  - mode = count;
  - frames = 120;
  - size = 60;
- Sampler = list;
- areAnonymousAllowed = false;
- userCache;
  - exponent = 13;

  - expirationTime = 780;
  - size = 6;

- flag.

9.5 Nedávno navštívená položka

Algoritmy v této podkapitole používají heuristiky *RecentHeuristic*. Tato heuristika si udržuje záznam o zpracovaných transakcích. V případě nové transakce, je přidána transakce na začátek. V případě odebrání transakce je transakce odebrána z konce. Pro doporučení je vybrána první položka od začátku, která je mezi povolenými položkami. Výchozí parametry algoritmu jsou:

- DataFrame = list;
- Sampler = overlap;
  - size = 60;
  - frames = 120;
  - mode = count.

Počáteční nastavení dosáhlo mírně lepších hodnot než doporučení posledního navštíveného článku. Přesnost doporučení je 2,6 % a katalogové pokrytí je 100%. Jakákoliv změna parametrů, reprezentace datového toku nebo okénkování neměla pozitivní vliv na přesnost.

Cachování uživatelů mělo negativní vliv na doporučování při malé velikosti paměti. Jakmile byla paměť dostatečně velká a hluboká, došlo k nepatrnému zlepšení o půl desetiny procenta. Filtrování dle flagu a ignorování anonymních uživatelů mělo pozitivní vliv na přesnost, i když došlo ke snížení katalogového pokrytí. Koncová konfigurace dosáhla přesnosti 3 % a pokrytí 59 %. Vliv jednotlivých aspektů je možné vidět v tabulce 9.4. Nastavení parametrů je následující:

- DataFrame = list;
- Sampler = overlap;
  - size = 60;
  - frames = 120;
  - mode = count.
- areAnonymousAllowed = false;
- userCache;

  - exponent = 13;
  - expirationTime = 420;
  - size = 6;
- flag.

Tabulka 9.4: Vztah aspektů na měřené metriky u heuristiky nedávno navštívené položky

Aspekty	**Vliv**	**Hodnoty**
Změna frames	0	10-300
Změna size	0	10-300
Změna mode	0	mode:time, time:7-72 s, frames:30-90
Změna implementace IDataFrame	-	Oddělené
Změna implementace ISampler	-	plovoucí okénko a rezervoár
Změna IDataFrame a ISampler	-	oddělený/single a rezervoár/plovoucí okénko
Cachování uživatelů	+	hloubka:1-10, expirace:1-21, velikost:4-8192
Filtrování na základě flagu	+	ano
Filtrování na základě koexistence	-	ano
Použití anonymních uživatelů	+	ne

9.6 Iterátor

Experimenty v této podkapitole používají heuristiku *IteratorHeuristic*. Tato heuristika doporučí první možnou položku z dodané kolekce. Výchozí nastavení algoritmu je:

- DataFrame = list;
- Sampler = overlap;
  - size = 60;
  - frames = 120;
  - mode = count.

Experimentování s velikostí okének a překryvů vedlo k mírnému zlepšení přesnosti. Nejlepších výsledků je dosaženo při malých okéncích a překryvem poloviny velikosti. Tento přístup dosahuje lepších výsledků než poslední čtený článek, i když myšlenka je velmi podobná. Vzhledem k malým okénkům je katalogové pokrytí 33%. Detailnější vztah jednotlivých aspektů je možno vidět na obrázku 9.6.

Obrázek 9.6: Vztah velikosti okénka a délkou překryvu u heuristiky iterátor (velikost je dána v počtu transakcí)

Velikost okénka	Velikost překryvu 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	MAX
2	1.29%	2.50%	2.50%																		2.50%
3	1.75%	2.55%	2.59%	2.59%																	2.59%
4	1.96%	2.59%	2.64%	2.59%	2.59%																2.64%
5	2.10%	2.60%	2.64%	2.66%	2.59%	2.59%															2.66%
6	2.19%	2.60%	2.64%	2.62%	2.66%	2.60%	2.60%														2.66%
7	2.24%	2.59%	2.63%	2.64%	2.61%	2.61%	2.60%	2.60%													2.64%
8	2.30%	2.60%	2.64%	2.66%	2.66%	2.61%	2.63%	2.61%	2.61%												2.66%
9	2.35%	2.60%	2.60%	2.64%	2.66%	2.63%	2.61%	2.66%	2.62%	2.62%											2.66%
10	2.39%	2.61%	2.62%	2.62%	2.65%	2.65%	2.64%	2.62%	2.66%	2.63%	2.63%										2.66%
11	2.34%	2.63%	2.62%	2.59%	2.65%	2.62%	2.63%	2.60%	2.57%	2.58%	2.57%	2.57%									2.65%
12	2.39%	2.58%	2.63%	2.63%	2.60%	2.66%	2.61%	2.63%	2.62%	2.62%	2.62%	2.59%	2.59%								2.66%
13	2.39%	2.58%	2.59%	2.63%	2.61%	2.60%	2.67%	2.59%	2.61%	2.61%	2.57%	2.60%	2.58%	2.58%							2.67%
14	2.43%	2.57%	2.60%	2.61%	2.63%	2.60%	2.60%	2.66%	2.63%	2.62%	2.62%	2.65%	2.62%	2.62%	2.62%						2.66%
15	2.40%	2.60%	2.58%	2.58%	2.61%	2.63%	2.60%	2.61%	2.57%	2.59%	2.58%	2.60%	2.57%	2.60%	2.58%	2.58%					2.63%
16	2.45%	2.56%	2.61%	2.58%	2.58%	2.59%	2.64%	2.58%	2.62%	2.64%	2.62%	2.63%	2.60%	2.63%	2.62%	2.59%	2.59%				2.64%
17	2.43%	2.59%	2.57%	2.61%	2.57%	2.57%	2.58%	2.61%	2.56%	2.62%	2.64%	2.60%	2.59%	2.61%	2.58%	2.58%	2.56%	2.56%			2.64%
18	2.45%	2.56%	2.59%	2.57%	2.62%	2.57%	2.57%	2.60%	2.58%	2.59%	2.60%	2.62%	2.62%	2.59%	2.60%	2.60%	2.60%	2.58%	2.58%		2.62%
19	2.48%	2.58%	2.58%	2.56%	2.58%	2.62%	2.57%	2.58%	2.59%	2.59%	2.60%	2.61%	2.62%	2.62%	2.61%	2.63%	2.63%	2.63%	2.61%	2.61%	2.63%
MAX	2.48%	2.63%	2.64%	2.66%	2.66%	2.66%	2.67%	2.66%	2.66%	2.64%	2.64%	2.65%	2.62%	2.63%	2.62%	2.63%	2.63%	2.63%	2.61%	2.61%	2.67%

Všechny variace založené na *FloatingWindowSampler* dosahovaly konzistentních výsledků jako poslední čtený článek. Algoritmy založené na rezervoáru občas dosáhly lepších výsledků. V průměru však se jedná o stejné výsledky jako poslední čtený článek. Separované modely dosáhly dobrých výsledků, a tak je možné využít tuto heuristiku k paralelnímu učení a doporučování. Ca-

Obrázek 9.7: Vztah hloubky cache paměti a dobou expirace záznamů u heuristiky iterátor

Doba expirace v minutách	Velikost historie 1	2	3	4	5	6
1	2.68%	2.71%	2.71%	2.71%	2.71%	2.71%
3	2.69%	2.72%	2.73%	2.73%	2.73%	2.73%
5	2.69%	2.73%	2.73%	2.74%	2.74%	2.74%
7	2.69%	2.73%	2.74%	2.74%	2.75%	2.75%
9	2.69%	2.73%	2.74%	2.75%	2.75%	2.75%
11	2.69%	2.73%	2.74%	2.75%	2.75%	2.75%
13	2.69%	2.73%	2.74%	2.75%	2.75%	2.75%
15	2.69%	2.74%	2.74%	2.75%	2.76%	2.76%

chování zvyšuje přesnost při velikosti alespoň 1024 záznamů. Zvýšení je velmi malé. Zvětšením hloubky paměti nad 2 položky nebo zvýšením doby expirace nad 15 minut nedojde ke zvýšení přesnosti. Detailnější vztah mezi hloubkou

Tabulka 9.5: Vztah aspektů na měřené metriky u heuristiky iterátor

Aspekty	**Vliv**	**Hodnoty**
Změna frames	+	2-300
Změna size	+	0-300
Změna mode	-	mode:time, time:6-90, frames:30-90
Změna implementace IDataFrame	0	oddělené a single
Změna implementace ISampler	0	plovoucí okénko a rezervoár
Změna IDataFrame a ISampler	0	oddělený/single a rezervoár/plovoucí okénko
Cachování uživatelů	+	hloubka:1-10, expirace:1-21, velikost:4-8192
Filtrování na základě flagu	+	ano
Filtrování na základě koexistence	-	ano
Použití anonymních uživatelů	+	ne

paměti a délkou expirace je možné vidět na obrázku 9.7. Pomocí filtrování dle flagu a ignorování anonymních uživatelů na testování je dosaženo přesnosti 3 % a katalogového pokrytí 25 %. Vliv jednotlivých aspektů na metriky je shrnut v tabulce 9.5. Závěrečné nastavení algoritmu je:

- DataFrame = list;
- Sampler = overlap;
  - size = 60;
  - frames = 120;
  - mode = count.
- areAnonymousAllowed = false;
- userCache;
  - exponent = 13;
  - expirationTime = 420;
  - size = 6;
- flag.

Tabulka 9.6: Souhrný přehled metrik vítězných kandidátů pro každou heuristiku za první týden

Heuristika	**Přesnost**	**Katalogové pokrytí**
Náhodný výběr	2,9 %	50 %
Populárnost	7,5 %	2,6 %
Nedávno navštívená položka	3 %	59 %
Iterátor	3 %	25 %

Tabulka 9.7: Souhrný přehled metrik vítězných kandidátů pro každou heuristiku za celý datový tok

Heuristika	**Přesnost**	**Katalogové pokrytí**
Náhodný výběr	2,8 %	55 %
Populárnost	6,4 %	5 %
Nedávno navštívená položka	2,9 %	63 %
Iterátor	2,9 %	32 %

9.7 Vyhodnocení experimentů

Základní algoritmy dávají dobrý start pro testování, avšak bez jakýchkoliv rozšíření jsou pro datový tok nepoužitelné. Principy založené na novosti, náhodnosti a populárnosti byly rozšířeny technikami, které se využívají v dolování dat z datových toků. Výsledkem jsou 4 heuristiky, které byly různými technikami rozšířeny za účelem zvýšení přesnosti doporučování. Detailní výsledky jsou v tabulce 9.6. Po nalezení nejlepších konfigurací byla provedena validace konfigurací na celém datovém toku. Přesnost algoritmů se snížila, ale katalogové pokrytí se zvýšilo. Detailní výsledky je možné vidět v tabulce 9.7.

Aplikováním principů dolování dat z datových toků bylo dosaženo zlepšení přesnosti u všech heuristik. Techniky mají nízké paměťové nároky (omezeno $\mathcal{O}(data\ frame\ size + 2^{cache\ exponent} \cdot cache\ size)$) a přidávají minimální časové zpoždění. Z toho vyplývá vhodnost technik na zpracování datových toků v doméně novinových článků.

Použití anonymních uživatelů k trénování nevedlo ke zlepšení přesnosti. Cachování uživatelů bylo vhodnou technikou, i když míra vlivu se lišila u jednotlivých heuristik. Filtrování na základě flagu se ukázalo být dobrým přístupem, neboť nikdy nevedlo ke zhoršení přesnosti.

Naopak negativní dopad měla koexistence článků. Propojení článků nevedlo ke zlepšení u žádné heuristiky. Zatímco u některých heuristik bylo zhoršení jen v řádech desetin procenta, tak v jiných případech byla přesnost nulová. Toto lze vysvětlit nevhodně reprezentovanou koexistencí článku. V upraveném datasetu chybí informace o přechodu uživatele z jedné stránky na druhou. Na základě toho není možné bez pamatování si historie uživatelů možné zjistit, jaké stránky uživatel navštívil.

Velikost okének byla fixní a relativně malá. Zpracování takovýchto okének vedlo k výrazně lepším výsledkům. Tento trend bylo možno sledovat u všech heuristik. To lze vysvětlit důrazem na adaptabilitu datového toku. Jakékoliv snížení adaptability vedlo ke zhoršení přesnosti a obvykle zlepšení katalogového pokrytí. Malá okénka dosahovala lepších přesností než okénka velikosti 1. To bylo dáno doporučením alternativy, pokud uživatel už na dané stránce byl.

Pokud byla velikost okénka definována v počtu transakcí, bylo dosaženo vyšší přesnosti než v případě okének definovaných časovým rozsahem. Zde může být negativní vliv dán nočními hodinami, kde je transakcí méně a okénka nemusí být zcela naplněna a proto mohou být nepřesná.

Průměrný čas na zpracování jedné transakce byl u všech heuristik nejvýše 0,01 ms. Vzhledem k omezení na 100 ms a relativně malým variacím mezi experimenty nejsou tyto hodnoty u jednotlivých heuristik uvedeny. Lze říci, že navržené a zkoumané heuristiky jsou velmi rychlé a zvládají zpracovat velké množství transakcí během krátké doby.

Dle výsledků soutěže bylo během online testování dosaženo CTR 2,5 %. Výsledky během online testování byly horší než výsledky během offline testování. Na základě těchto závěrů lze předpokládat, že výše navržení kandidáti by dosáhli horších výsledků v online testování. Ze zdrojů o offline testování lze vyvodit, že navržení kandidáti i tak dosahují dobrých výsledků [15][11]. Vzhledem ke ztíženým podmínkám ohledně počtu doporučení je možné, že by kandidáti mohli dosáhnout výrazně lepších výsledků.

Během experimentů byly vyhodnoceny metriky: přesnost, katalogové pokrytí a průměrná doba odpovědi. Zbylé metriky je možné částečně vyhodnotit:

- Pokrytí uživatelů dosáhlo 100 % neboť všechny heuristiky nějaké doporučení vždy navrhly.
- Důvěru systému nebylo možné měřit vzhledem k absenci dodatečných informací o uživateli. Důvěra v navržené položky by byla pro všechny položky stejná, případně s velmi hrubou granularitou.
- Míru spolehlivosti nelze snadno vyčíslit. Uživateli nebyla nabídnuta současná stránka nebo v případě cachování jedna z posledních stránek. I tak však pro delší sessions nebo kolizi při cachování mohlo dojít ke zobrazení již zhlédnutého článku. Z tohoto důvodu je spolehlivost spíše nízká.
- Novost se u jednotlivých heuristik liší. Zatímco některé byly založené na doporučování nejnovějších položek, tak například heuristika populárnosti s dostatečně velkým překryvem doporučovala spíše časté články, které mohly být pro uživatele již známé.
- Diverzitu článků nelze měřit vzhledem k tomu, že vždy byla doporučována pouze jedna položka. Určitým ukazatelem diverzity je katalogové pokrytí, které ukazuje jakých různých hodnot mohlo doporučení nabývat.

Pokud bylo použito cachování, tak nedocházelo k opakovanému doporučování přečteného článku. Tím pádem články určitou diverzitu měly. Samotná diverzita článků však nebyla měřena a je obtížné ji vyhodnotit vzhledem k chybějícím strukturovaným datům a komplikované textové analýze.

- Míru překvapení doporučení není možné změřit.

Závěr

Cílem práce bylo zhodnotit využití principů dolování datových toků v doporučovacích systémech. Na základě teoretické části o doporučujících systémech, datových tocích a dolování dat bylo možné v rešerši existujících řešení rozdělit algoritmy do skupin. V algoritmech byly identifikovány techniky z dolování dat v datových tocích, které byly následně použity v implementační a experimentální části.

Rešerše existujících řešení ukázala vhodnost populárnosti položky na doporučování. Autoři byli schopni dosáhnout dobrých výsledků, i když nebrali v potaz informaci o uživateli nebo položce. Díky této simplifikaci jsou kandidáti založení na populárnosti vhodní pro techniku dolování dat z datových toků. Paměťová a výpočetní náročnost algoritmů je nízká, stejně jako doba odezvy. Zpracování položek či uživatelů zvyšuje přesnost algoritmů za cenu zpomalení učení, zpomalení doporučení a větší paměťové zátěže.

Platformu Idomaar, která byla používána v rámci soutěže CLEF NewsREEL challenge, již není možné dle dostupných návodů a skriptů používat. Autoři neudržují projekt aktuální a skripty pro vytvoření virtuálních strojů nefungují. Z tohoto důvodu byla vybrána platforma StreamingRec, jejíž autoři navazují na platformu Idomaar. Platforma je vhodná pro offline testování a podporuje jednoduché rozšíření platformy o vlastní algoritmy a metriky. V rámci rešerše byly popsány existující funkcionality. V implementační části byly vypsány změny, které byly provedeny.

Kapitola dolování dat definovala postup, jak mají být data zpracována, aby z nich mohly být vytěženy informace a zároveň byla použitelná platformou. Vstupní dataset musel být transformován do dvou souborů ve formátu CSV, kde jeden soubor popisoval položky a druhý transakce. Všechny záznamy referencovaly vydavatelství s identifikátorem 418. Analýza dat popsala vlastnosti dat, jejich vývoj během dne a měsíce.

V rámci implementace byly popsány změny platformy, použité metriky a implementovaný parametrizovatelný proudový algoritmus. Platforma byla rozšířena o zpracování anonymních uživatelů. Jako hlavní metrika byla zvolena

přesnost doporučení. Dalšími metrikami jsou katalogové pokrytí a průměrná doba odezvy na požadavek o doporučení. Implementace proudového algoritmu nabízí širokou míru parametrizace jednotlivých technik. Techniky jsou rozděleny do následujících skupin: správa datových rámců, správa vzorků, filtry a heuristiky.

V experimentální části byly nejprve zkoumány samotné heuristiky: náhodný výběr, poslední navštívený článek a nejnavštěvovanější článek. Testování probíhalo na čtvrtině datového toku. Heuristika doporučení posledního navštíveného článku získala přesnost 2,5 % s katalogovým pokrytím 100 %. Zbylé heuristiky měly velmi nízkou přesnost. Všechny metody měly dobu odpovědi pod 0,01 ms.

Vzhledem k rozsáhlé parametrizaci proudového algoritmu byly experimenty rozděleny do sekcí dle heuristik. Pro každou heuristiku byly zkoumány jednotlivé parametry s různými nastaveními. Postupným průchodem stavovým prostorem byli vybráni nejlepší kandidáti pro každou heuristiku. Tito kandidáti byly následně validováni na celém datovém toku.

Heuristika založená na populárnosti dosáhla suverénně nejlepších výsledků s přesností 6,4 % a katalogovým pokrytím 5 %. Zbylé heuristiky dosáhly přesnosti okolo 2,9 % s katalogovým pokrytím v rozmezí 32-63 %. Vzhledem ke zvolení přesnosti jako dominantní metriky je zvolena heuristika založená na populárnosti za nejlepšího kandidáta.

Přesnost jednotlivých heuristik je v porovnání s dostupnými výsledky ze soutěží dobrá. V soutěži bylo dosaženo CTR nejvýše 2,5 %. Oproti soutěžní platformě byla vyhodnocena jen polovina transakcí, neboť zbytek transakcí měl v session jen jeden článek nebo měli anonymního uživatele. Vzhledem k simplifikaci datasetu nebylo možné tyto uživatele vyhodnotit. V soutěži byl počet položek v doporučení dán požadavkem, zatímco v experimentech byla doporučena vždy jen jedna položka. Experiment tedy není identický, ale z existující dokumentace lze vyvodit rozsah odchylky. V nejhorším případě, by byla přesnost kandidátů na soutěžní platformě poloviční, což je stále velmi dobrý výsledek. Vzhledem k předpokladu, že by heuristiky zafungovali i pro položky ignorované v experimentální části lze předpokládat, že by hodnoty byly sníženy o méně než polovinu, ale jistě by ke snížení došlo například kvůli nemožnosti cachování anonymů. Je tedy možné, že by kandidáti dosáhli velmi dobrých výsledků i v online testováních.

Zvolené techniky dolování dat z datových toků se ukázaly být klíčové pro dosažení vyšší přesnosti. Velikost okénka umožnovala adaptivitu modelu. Pomocí cachování bylo dosaženo zvýšení přesnosti za cenu malé paměti. Během optimalizace parametrů bylo důležité zvolit vhodnou hranici, kdy nedochází k velkému množství konfliktů. Vzhledem k návrhu je algoritmus dobře škálovatelný, neboť paměťovou náročnost lze ovlivnit prostřednictvím parametrů. Výsledné kandidáty je možné využít i pro distribuované doporučování. Vhodný způsob náčrtu dokázal zvýšit přesnost u doporučujících heuristik. Vzorkování se ukázalo jako velmi rychlý a snadný způsob, jak doporučovat

položky s minimálními požadavky na systém.

Závěrem lze říci, že implementováním technik dolování datového toku k doporučování novinových článků byla zvýšena přesnost doporučování. Techniky jsou vhodné ke zvýšení adaptability modelu a snížení paměťové a výpočetní náročnosti. Pomocí okének a náčrtů je možné detekovat změny v toku a dynamicky na ně reagovat. Doména novinových článků je vhodnou doménou pro testování jednotlivých technik, neboť krátká životnost článků klade důraz na rychlou adaptabilitu a zapomínání starých transakcí.

Literatura

[1] Rosati, L.: How to design interfaces for choice: Hick-Hyman law and classification for information architecture. In *How to design interfaces for choice: Hick-Hyman law and classification for information architecture*, ročník Classification and visualization. Interfaces to knowledge, Haag, Nizozemsko, 10 2013, str. 1.

[2] Aggarwal, C. C.: *Recommender Systems: The Textbook*. Springer, první vydání, 2016, ISBN 978-3-319-29657-9.

[3] Ricci, F.; Rokach, L.; Shapira, B.: *Recommender Systems Handbook*. Springer, 2011, ISBN 978-0-387-85819-7.

[4] Kille, B.; Lommatzsch, A.; Turrin, R.: Stream-Based Recommendations: Online and Offline Evaluation as a Service. In *CLEF*, 9 2015, str. 7.

[5] Aggarwal, C. C.: *Data Mining: The Textbook*. Springer, 2015, ISBN 978-3-319-14141-1.

[6] Aggarwal, C. C.: *Data Streams: Models And Algorithms*. Springer, 2007, ISBN 978-0-387-47534-9.

[7] Gama, J.: *Knowledge Discovery from Data Streams*. Chapman and Hall/CRC, první vydání, 2010, ISBN 9781439826119.

[8] Bifet, A.; Gama, J.; Gavalda, R.; aj.: Advanced Topics on Data Stream Mining. In *CLEF*, Bristol, Spojené království, 9 2012, s. 14–30.

[9] Chekuri, C.: CS 598: Algorithms for Big Data: Lecture 4. online, 9 2014, [cit. 2019-04-25]. Dostupné z: `https://courses.engr.illinois.edu/cs598csc/fa2014/Lectures/lecture_4.pdf`

[10] Hopfgartner, F.; Kille, B.; Lommatzsch, A.; aj.: Benchmarking News Recommendations in a Living Lab. In *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*, Springer International Publishing, 2014, ISBN 978-3-319-11382-1, s. 250–267.

[11] Liang, Y.; Loni, B.; Larson, M.: CLEF NewsREEL 2017: Contextual Bandit News Recommendation. In *CLEF*, 2017, str. 8.

[12] Corsini, F.; Larson, M.: CLEF NewsREEL 2016: Image based Recommendation. In *CLEF*, 2016, s. 3,7.

[13] Beck, P. D.; Blaser, M.; Michalke, A.; aj.: A System for Online News Recommendations in Real-Time with Apache Mahout. In *CLEF*, 2017, s. 10,11.

[14] Lommatzsch, A.; Plumbaum, T.; Albayrak, S.: A Linked Dataverse Knows Better: Boosting Recommendation Quality Using Semantic Knowledge. *SEMAPRO 2011: The Fifth International Conference on Advances in Semantic Processing*, 01 2011: s. 97–103.

[15] Golian, C.: *Aplikace pro doporučování novinových zpráv v reálném čase.* Diplomová práce, České vysoké učení technické v Praze, Fakulta informačních technologií, Praha, Česká republika, 2017.

[16] Lommatzsch, A.: Real-Time News Recommendation Using Context-Aware Ensembles. In *Advances in Information Retrieval*, Springer International Publishing, 2014, ISBN 978-3-319-06028-6, s. 51–62.

[17] Kille, B.; Lommatzsch, A.; Hopfgartner, F.; aj.: CLEF NewsREEL 2016: Comparing Multi-dimensional Offline and Online Evaluation of News Recommender Systems. In *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum*, Évora, Portugal, 9 2016, s. 593–605. Dostupné z: `http://ceur-ws.org/Vol-1609/16090593.pdf`

[18] Kille, B.; Brodt, T.; Heintz, T.; aj.: NewsREEL 2014: Summary of the News Recommendation Evaluation Lab. In *Working Notes for CLEF 2014 Conference*, Sheffield, UK, 9 2014, s. 790–801.

[19] Kille, B.; Lommatzsch, A.; Turrin, R.; aj.: Overview of CLEF NewsREEL 2015: News Recommendation Evaluation Lab. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum*, Toulouse, France, 9 2015, s. 4–11.

[20] Kille, B.; Lommatzsch, A.; Hopfgartner, F.; aj.: CLEF 2017 NewsREEL Overview: Offline and Online Evaluation of Stream-based News Recommender Systems. In *CLEF 2017: Conference and Labs of the Evaluation Forum*, Dublin, Ireland, 9 2017, s. 650–654.

[21] Lommatzsch, A.: Recommender Algorithms for News Streams [online]. online, 7 2016, [cit. 2019-04-25]. Dostupné z: http://irml.dailab.de/wp-content/uploads/2016/07/20160711_lommatzsch_recommenderChallenge.pdf

[22] Gebremeskel, G. G.; de Vries, A. P.: Recommender Systems Evaluations : Offline, Online, Time and A/A Test. In *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum*, Évora, Portugal, 9 2016, s. 642–656.

[23] Tavakolifard, M.; Gulla, J.; Almeroth, K.; aj.: Workshop and challenge on news recommender systems. In *RecSys 2013 - Proceedings of the 7th ACM Conference on Recommender Systems*, 10 2013, s. 481–482, doi: 10.1145/2507157.2508004.

[24] Plista: *ORP Protocol [online]*. Verze 0.4.1, [cit. 2019-04-25]. Dostupné z: https://www.plista.com/wp-content/uploads/2017/07/plista_ORP_final.pdf

[25] Jugovac, M.; Jannach, D.; Karimi, M.: Streamingrec: a framework for benchmarking stream-based news recommenders. In *RecSys '18 Proceedings of the 12th ACM Conference on Recommender Systems*, 09 2018, s. 269–273, doi:10.1145/3240323.3240384.

Příloha **A**

Seznam použitých zkratek

RMSE Root mean square error

MAE Mean absolute error

MRR Mean reciprocal rank

CUSUM Cumulative sum

CPU Central processing unit

NewsREEL News Recommendation Evaluation Lab

CLEF Conference and Labs of the Evaluation Forum

URL Uniform resource locator

JSON JavaScript Object Notation

GmbH Gesellschaft mit beschränkter Haftung

TU Technische Universität

OS Operační systém

ORP Open Recommendation Platform

SDK Software development kit

KPI Key performance indicator

CSV Comma-separated values

PSČ Poštovní směrovací číslo

FIFO First in, first out

JVM Java virtual machine

Příloha **B**

Obsah přiloženého flash disku

- `readme.txt` ..... stručný popis obsahu flash disku
- `jar` ..... adresář se spustitelnou formou implementace
- `src`
  - `impl` ..... zdrojové kódy implementace
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `text` ..... text práce
  - `thesis.pdf` ..... text práce ve formátu PDF