SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

Vyhľadávanie informácií

ZLAVADNA.SK RECOMMENDER

Autor: Bc. Jozef Zaťko

Vypracované: zimný semester 2016/2017

1 Špecifikácia zadania

Cieľom zadania bolo vytvoriť **odporúčací systém pre portál zlavadna.sk**, ktorý aktuálnemu používateľovi prezerajúcemu si zľavu (deal) odporučí 10 (alebo aj iný počet) jemu relevantných zliav. Vstupom je id používateľa a id aktuálnej zľavy (deal).

Pre účely vytvorenia odporúčacieho systému boli k dispozícii dáta z roku 2014, ktoré boli rozdelené na 2 časti:

- Tréningová množina (január – júl 2014)
- Testovacia množina (august – september 2014)

Celkovo sme si vytýčili vytvoriť nasledovné typy odporúčaní:

1. **Odporúčanie založené na obsahu (content-based recommender)**
    - podobné zľavy k aktuálne prezeranej zľave na základe podobnosti textu
    - podobné zľavy ku všetkým zľavám, ktoré si daný používateľ kúpil (tiež na základe textu)
    - kombinácia predchádzajúcich dvoch prípadov
2. **Odporúčanie na základe podobnosti používateľov** (collaborative recommender)
    - nájdenie podobných používateľov k aktuálnemu používateľovi podľa kúpených zliav (z tréningovej množiny) a následné nájdenie najkupovanejších zliav, ktoré si aktuálny používateľ nekúpil (z testovacej množiny)
3. **Odporúčanie najpredávanejších zliav v lokalite aktuálne prezeranej zľavy**

Tieto prístupy k odporúčaniu sme následne vyhodnotili na testovacej množine a porovnali.

2 Použité technológie v projekte

- Java 8 (Maven, log4j)
- PostgreSQL 9.4
- Postgis 2.3.0
- Hibernate 5.2.4
- JPA 2.1
- Elasticsearch 2.4.1

3 Reprezentácia dát a ich import

K dispozícii sme dostali 6 súborov vo formáte CSV:

- train_dealitems.csv
- train_deal_details.csv
- train_activity.csv
- test_dealitems.csv
- test_deal_details.csv
- test_activity.csv

Ide o tréningové a testovacie dáta 3 modelov (deal, dealitem, activity), vrátane ich prepojení. Na základe týchto dát sme vytvorili dátový model (rozšírení o samostatnú triedu pre používateľa). Vznili tak 4 triedy:

- Deal

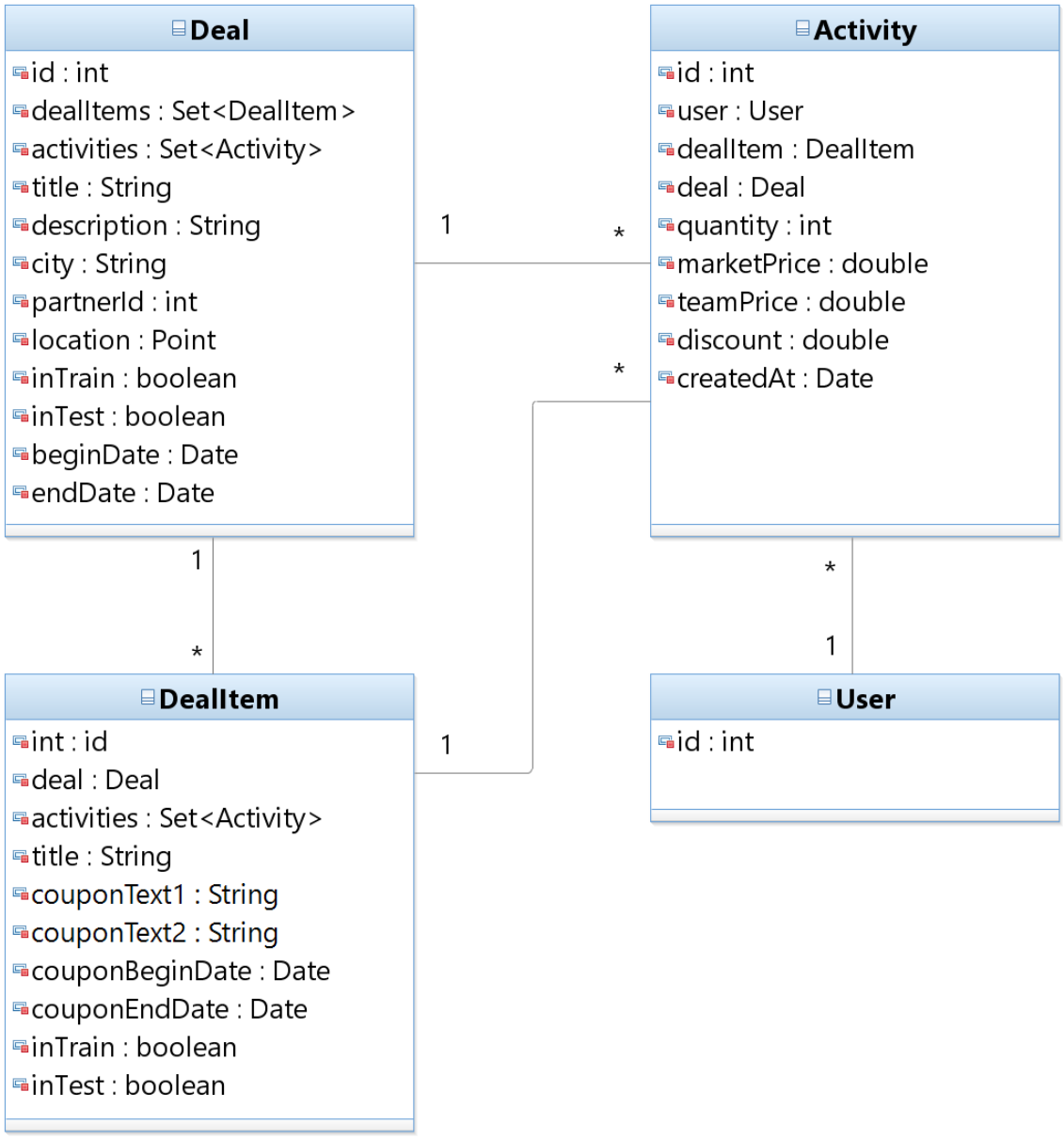
- DealItem
- Activity
- User

3.1 Postgres relačná databáza

Celý dátový model sme pomocou JPA a Hibernate (cez XML súbory) namapovali na Postgres databázu. Urobili sme tak z nasledujúcich dôvodov:

- štruktúra dát kopíruje normalizovaný entitno-relačný databázový model
- databáza zabezpečuje perzistenciu dát
- databáza poskytuje silný jazyk SQL pre vytváranie dopytov
- Postgres rozšírenie Postgis umožňuje goe-dopyty

Výsledný dátový model vyzerá nasledovne:



Obrázok 1 – Dátový model odporúčacieho systému

Dátový model sme kvôli potrebe odporúčania rozšírili o ďalšie atribúty, kvôli zrýchleniu a zjednoduchšeniu dopytov. Sú to tieto atribúty:

Deal:

- **location**: transformácia atribútov gpslat a gpslong z csv súborov do typu Point, ktorý sa vďaka geo-rožíreniu Postgis namapuje v databáze na typ geometry a umožní geo-dopyty
- **inTrain**: boolean hodnota určujúca, či je daný deal v tréningovej množine dát
- **inTest**: boolean hodnota určujúca, či je daný deal v testovacej množine dát
- **beginDate**: boolean hodnota určujúca, od kedy je daný deal platný (podľa platnosti prvého DealId)

```
UPDATE deals dls
SET begin_date = (
    SELECT i.coupon_begin_date FROM dealitems i
    JOIN deals d
    ON i.deal_id = d.id
    WHERE d.id = dls.id
    ORDER BY i.coupon_begin_date
    LIMIT 1
)
```

- **endDate:** boolean hodnota určujúca, do kedy je daný deal platný (podľa platnosti posledného DealId)

```
UPDATE deals dls
SET end_date = (
    SELECT i.coupon_end_date FROM dealitems i
    JOIN deals d
    ON i.deal_id = d.id
    WHERE d.id = dls.id
    ORDER BY i.coupon_end_date DESC
    LIMIT 1
)
```

DealItem:

- **inTrain**: boolean hodnota určujúca, či je daný deal v tréningovej množine dát
- **inTest**: boolean hodnota určujúca, či je daný deal v testovacej množine dát

Ako databázový systém sme vybrali **PostgreSQL** a dáta sme importovali cez vytvorené aplikáciu. Po vyskúšaní rôznych možností sme tréningové aj testovacie dáta vložili **do jednej spoločnej databázy**, pričom na rozšírenie dealov a dealitemov sme použili atribúty **inTrain a inTest**.

Počas importu dát sme **odstránili duplicitné dáta** ktoré sa v CSV súboroch nachádzali, dealy ktoré mali **prázdne atribúty** (title alebo description), alebo obsahovali text „testovacia zlava“. Taktiež sme odstránili dáta, ktorý mali cudzí kľúč v neexistujúcom zázname.

Všetky atribúty, ktoré používame pri generovaní odporúčaní sme z výkonnostných dôvodov obohatili o **indexy**, ktoré sme nastavili na úrovni aplikácie.

3.2 Elasticsearch

Keďže sme sa rozhodli vytvoriť **odporúčanie aj na základe podobnosti obsahu** dealov, tak sme potrebovali tieto dáta uložiť do nástroja ElasticSearch. Keďže sme robili **odporúčanie dealov na základe atribútov title a description,** tak nám stačilo indexovanie z tabuľky deals.

Pred indexovaním dát bolo potrebné **nastaviť Elasticsearch pre slovenčinu**. S týmto nám pomohol Github repozitár Slovenskej národnej galérie https://github.com/SlovakNationalGallery/elasticsearch-slovencina. Tu sme našli všetky potrebné súbory vrátane **synoným** a **slovenských stop slov**. Na základe priloženého návody sme vytvorili **analyzer** (slovak_synonym_analyzer), ktorý z atribútov title a description vytvorí základné tvary slov vrátane synoným s výnimkou stop slov.

```
{
	"settings": {
		"analysis": {
			"filter": {
				"lemmagen_filter_sk": {
					"type": "lemmagen",
					"lexicon": "sk"
				},
				"sk_SK" : {
					"type" : "hunspell",
					"locale" : "sk_SK",
					"dedup" : true,
					"recursion_level" : 0
				},
				"synonym_filter": {
					"type": "synonym",
					"synonyms_path": "synonyms/sk_SK.txt",
					"ignore_case": true
				},
				"stopwords_SK": {
					"type": "stop",
					"stopwords_path": "stop-words/stop-words-slovak.txt",
					"ignore_case": true
				}
			},
			"analyzer": {
				"slovak_standard_analyzer": {
					"type": "custom",
					"tokenizer": "standard",
					"filter": [
						"stopwords_SK",
						"lemmagen_filter_sk",
						"lowercase",
						"asciifolding",
						"stopwords_SK"
					]
				},
				"slovak_synonym_analyzer": {
					"type": "custom",
					"tokenizer": "standard",
					"filter": [
						"stopwords_SK",
						"lemmagen_filter_sk",
						"lowercase",
						"synonym_filter",
						"asciifolding",
						"stopwords_SK"
					]
				}
			}
		}
```

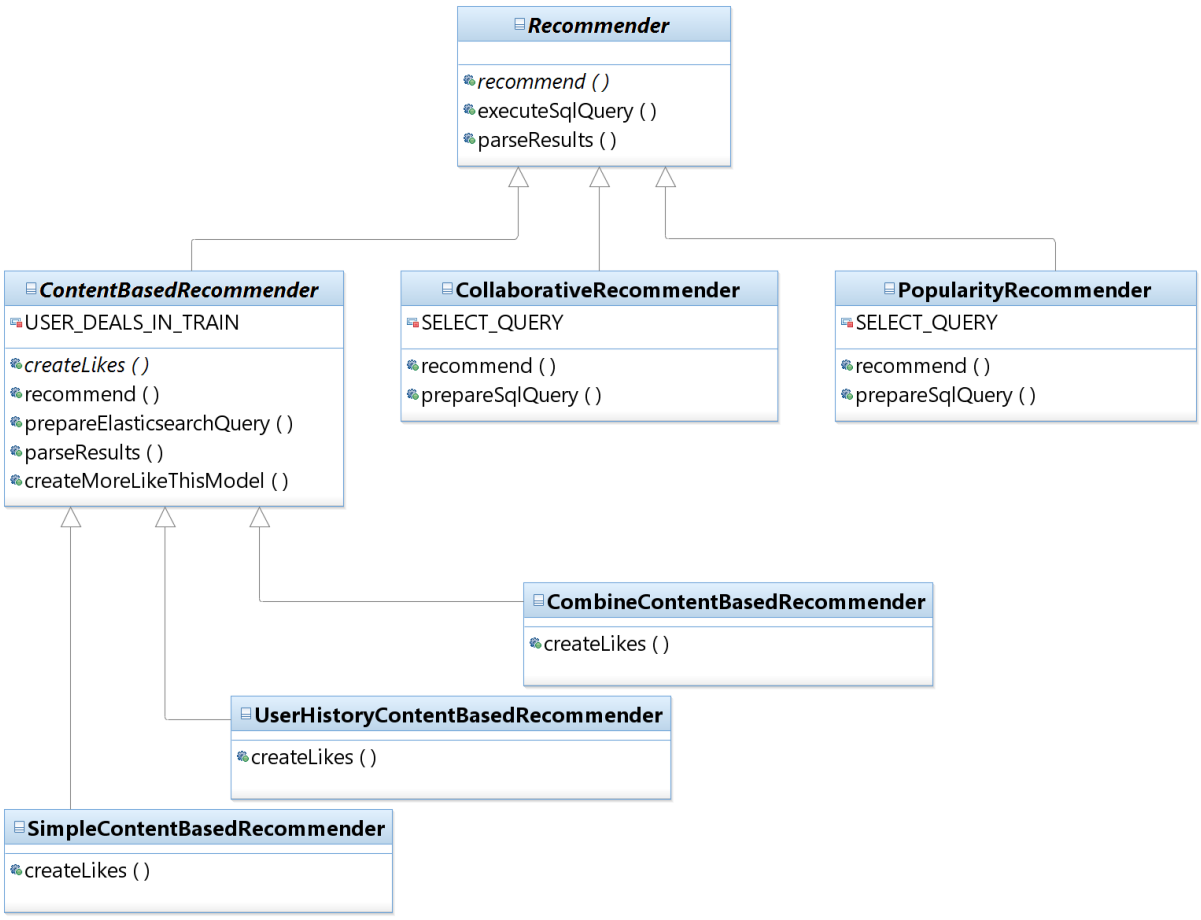
```
        },
        "mappings": {
                "deal": {
                        "properties": {
                                "title": {
                                        "type": "string",
                                        "analyzer": "slovak_synonym_analyzer"
                                },
                                "description": {
                                        "type": "string",
                                        "analyzer": "slovak_synonym_analyzer"
                                },
                                "in_test": {
                                        "type": "boolean"
                                },
                                "in_train": {
                                        "type": "boolean"
                                },
                                "begin_date": {
                                        "type": "date",
                                        "format": "yyyy-MM-dd"
                                },
                                "end_date": {
                                        "type": "date",
                                        "format": "yyyy-MM-dd"
                                }
                        }
                }
        }
}
```

Dáta do boli indexované cez aplikáciu priamo z databázy (tabuľka deals) **pomocou REST volania** na Elasticsearch API.

4 Odporúčanie dealov

Pre potreby odporúčaní sme vytvorili štruktúru tried, pričom jednotlivé konkrétne triedy predstavujú jeden zo spôsobov odporúčania (balík `sk.zatko.vi.recommender.recommend`). Vďaka tejto hiererchickej štruktúre je možné jednoducho pridávať nová typy odporúčania.

Základom je metóda `public ArrayList<Integer> recommend(int currentuserId, int currentDealId, Date currentDate, int countOfResults)`, ktorá vráti zoznam id odporúčaných dealov (ich počet je určený parametrom funkcie). Toto je realizované pre aktuálneho používateľa prezerejúceho si konkrétny deal v nejakom čase.



Obrázok 2 - Diagram tried, ktoré predstavujú jednotlivé typy odporúčania

5 Odporúčanie na základe obsahu dealov

Implementovali sme 3 typy odporúčania na základe obsahu:

1. podobné zľavy k aktuálne prezeranej zľave na základe podobnosti textu
   (trieda `SimpleContentBasedRecommender`)
2. podobné zľavy ku všetkým zľavám, ktoré si daný používateľ kúpil (tiež na základe textu)
   (trieda `UserHistoryContentBasedRecommender`)
3. kombinácia predchádzajúcich dvoch prípadov
   (trieda `CombineContentBasedRecommender`)

Tento spôsob odporúčania sa **realizuje na základe textovej podobnosti jednotlivých zliav**. Na základe štruktúry poskytnutých dát sme sa rozhodli hľadať podobnosť na základe atribútov **title** a **description**. Pre tieto polia sme nastavili analyzer (viď kapitolu 3.2).

Základom implementácie je more_like_this dopyt do Elastisearch-u. Ten sa vytvorí serializovaním tried balíka `sk.zatko.vi.recommender.elasticsearch.morelikethis` do JSON-u pomocou knižnice google.gson. Príprava more_like_this dopytu realizuje v metóde `prepareElasticSearchQuery()` v spoločnej nadtriede `ContentBasedRecommender.`

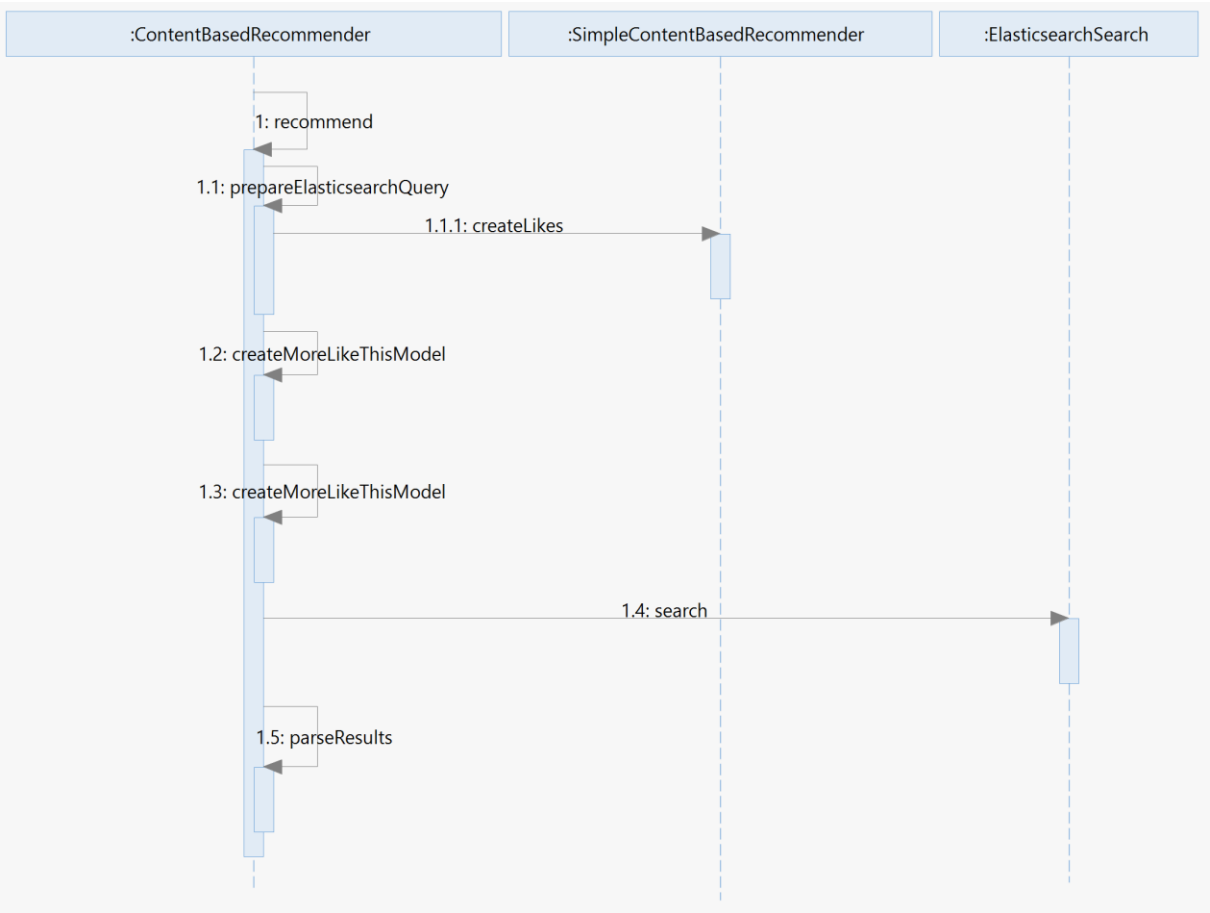
Aby sme zlepšili relevantnosť výsledkov nastavili sme atribút **boost atribútu title na 3**. To znamená, že zhoda v tomto atribúte bude mať výraznejší vplyv na výsledky dopytu. Tiež sme nastavili atribút **min_term_freq na hodnotu 1 pre atribút title a na hodnotu 2 pre atribút description**.

Všetky 3 implementované odporúčania na základe textu využívajú **more_like_this dopyt**, pričom a líšia v obsahu atribútu **„like“**, ktorého model sa vytvára metódov `createLikes()` konkrétnych tried.

Príklad dopytu:

```
{
	"query": {
		"filtered": {
			"query": {
				"dis_max": {
					"queries": [
						{
							"more_like_this": {
								"fields": [
									"title"
								],
								"like": [
									{
										"_index": "deals",
										"_type": "deal",
										"_id": "35749"
									}
								],
								"min_term_freq": 1,
								"boost": 3
							}
						},
						{
							"more_like_this": {
								"fields": [
									"description"
								],
								"like": [
									{
										"_index": "deals",
										"_type": "deal",
										"_id": "35749"
									}
								],
								"min_term_freq": 2,
								"boost": 1
							}
						}
					]
				}
			},
			"filter": {
				"range": {
					"end_date": {
						"lt": "2014-08-01",
						"format": "yyyy-MM-dd"
					}
				}
			},
			"filter": {
				"match" : {
					"inTest": true
				}
			}
		}
	}
}
```

Po vytvorení dopytu sa cez metódu `search()` triedy `ElasticsearchConnector` zavolá REST rozhranie inštancie ElasticSearch-u, pričom z výsledku dopytu sa v metóde `parseResults()` vytvorí zoznam id-čiek podobných dealov.



Obrázok 3 – Sekvenčný diagram vytvorenia odporúčania na základe obsahu

5.1 Podobné zľavy k aktuálne prezeranej zľave

Toto odporúčanie nájde obsahovo najpodobnejšie zľavové ponuky k aktuálnej zľave pomocou more_like_this dopytu. Obsah atribútu like tvorí len aktuálna zľava.

5.2 Podobné zľavy ku všetkým zľavám, ktoré si daný používateľ kúpil

Toto odporúčanie nájde obsahovo najpodobnejšie zľavové ponuky ku všetkým zľavám, ktoré si používateľ v tréningovej množine kúpil. Obsah atribútu like tvoria všetky tieto zľavy. Všetky používateľove zľavy získame nasledovným SQL dopytom:

```
SELECT DISTINCT deal_id, created_at FROM activities
WHERE created_at <= '2014-08-01 00:00:00
AND user_id = :user_id
```

5.3 Kombinácia prípadov 5.1 a 5.2

Tento spôsob odporúčania zahrnie do more_like_this dopytu aktuálny deal a aj všetky používateľom kúpené dealy.

6 Odporúčanie založené na podobnosti používateľov

Tento prístup je implementovaný triedou `CollaborativeRecommender` pričom na rozdiel od predchádzajúceho prípadu využíva dáta z relačnej databázy.

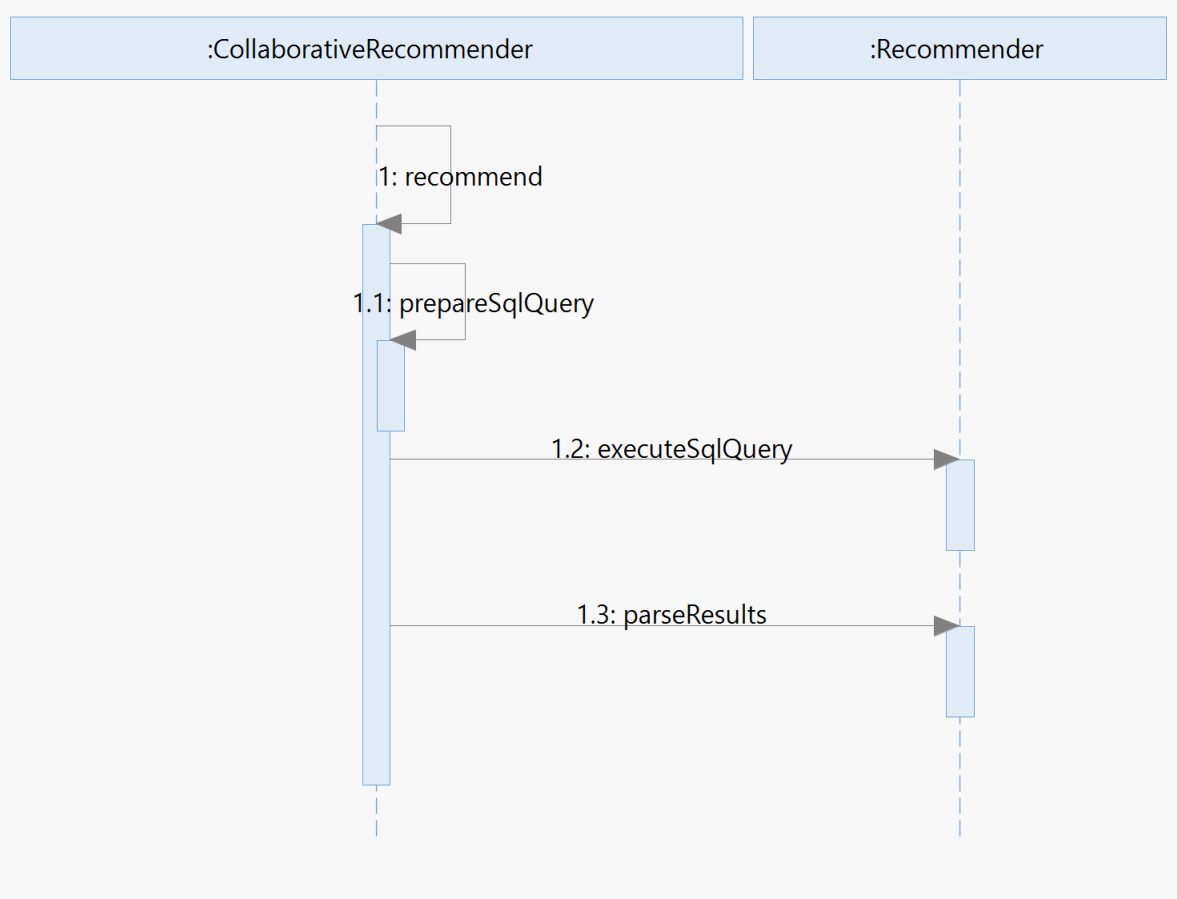
Základom je **SQL dopyt**, ktorý sa vytvára v metóde prepareSqlQuery(). Tento dopyt nájde v tréningovej množine pre aktuálnej používateľa **20 najpodobnejších používateľov na základe rovnakých dealov, ktoré si kúpili**. Dôležité je to, že ponuku, ktorý si nejaký používateľ kúpil viackrát, započíta ako jednu (DISCTINC).

Potom u týchto 20 používateľov nájde najkupovanejšie ešte platné zľavy, ktoré si aktuálny používateľ nekúpil.

```
WITH
user_activities AS (
      SELECT DISTINCT deal_id, user_id FROM activities a
      WHERE created_at <= '2014-08-01 00:00:01'
      AND user_id = :user_id
),
all_activities AS (
      SELECT DISTINCT deal_id, user_id FROM activities
      WHERE created_at <= '2014-08-01 00:00:00'
),
similiar_users AS(
      SELECT aa.user_id, count(aa.deal_id) FROM user_activities ua
      JOIN all_activities aa
      ON ua.deal_id = aa.deal_id AND ua.user_id <> aa.user_id
      GROUP BY aa.user_id
      ORDER BY count DESC
      LIMIT 20
)

SELECT a.deal_id, count(a.id) FROM activities a
JOIN deals d ON d.id = a.deal_id
JOIN similiar_users su ON su.user_id = a.user_id
WHERE NOT EXISTS (
      SELECT * FROM activities
      WHERE user_id = :user_id
      AND d.id = deal_id
)
AND a.created_at >= '2014-08-01 00:00:01'
AND d.in_test = true
AND d.end_date >= :date
GROUP BY a.deal_id
ORDER BY count DESC
LIMIT 10
```

Z výsledku dopytu sa v metóde parseResults() vytvorí zoznam id-čiek podobných dealov.



Obrázok 4 – Sekvenčný diagram vytvorenia odporúčania na základe podobnosti používateľov

7 Odporúčanie najpredávanejších zliav v lokalite aktuálne prezeranej zľavy

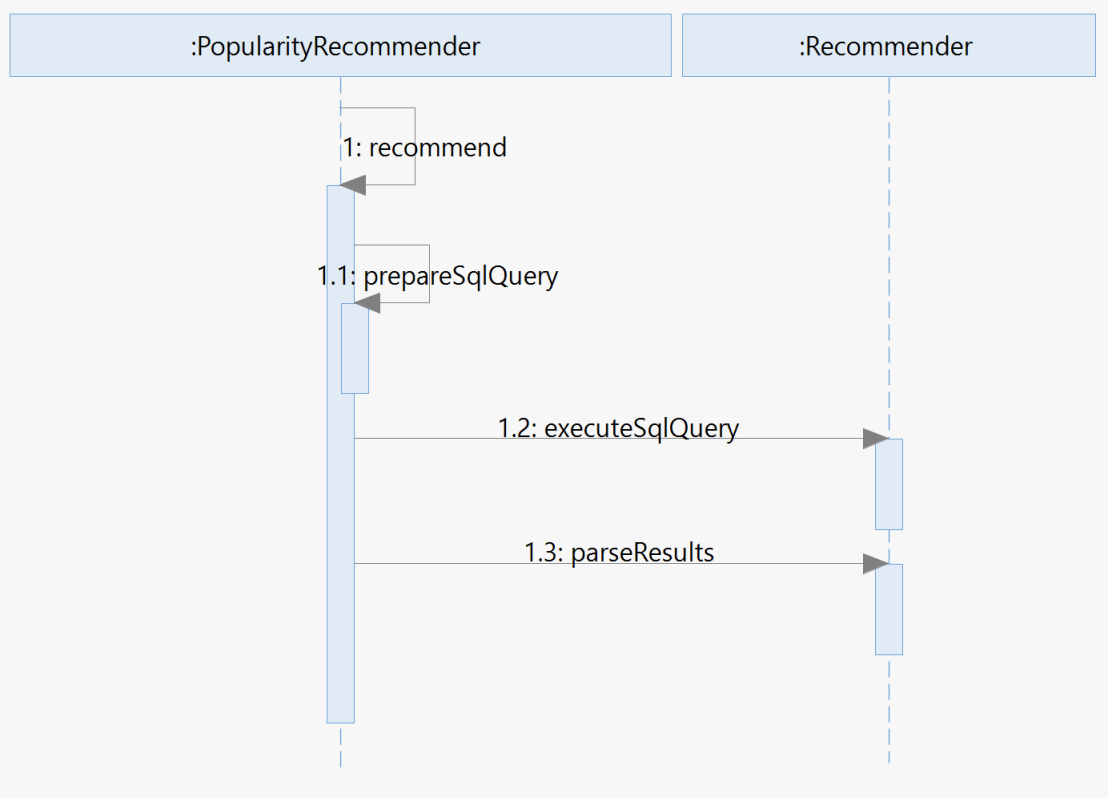
Posledným implementovaným spôsobom odporúčania bolo **odporúčania najpredávanejších zliav**, avšak len takých, ktoré sa nachádzajú **v geografickej blízkosti** aktuálne prezeranej zľavy (cca 50km).

Základom je SQL dopyt, ktorý sa vytvára v metóde `prepareSqlQuery()`. Tento dopyt nájde v testovacej množine také dealy, ktoré s kúpilo čo najviac unikátnych používateľov (DISCTINCT) a zároveň nie sú od aktuálne prezeranej zľavy ďalej ako 50km.

```
WITH
actual_deal AS (
	SELECT id, location FROM deals
	WHERE id = :deal_id
)
close_deals AS (
	SELECT d.id, d.end_date FROM deals d
	JOIN actual_deal ad ON ST_Distance(d.location, ad.location) < 0.4
	WHERE d.id <> ad.id
	AND d.in_test = true
)
dist_activities AS (
	SELECT DISTINCT deal_id, user_id FROM activities
)
```

```
SELECT a.deal_id, count(d.id) from close_deals d
JOIN dist_activities a ON d.id = a.deal_id
WHERE d.end_date > :date
GROUP BY a.deal_id
ORDER BY count DESC
LIMIT :limit";
```

Z výsledku dopytu sa v metóde parseResults() vytvorí zoznam id-čiek podobných dealov.



Obrázok 5 - Sekvenčný diagram vytvorenia odporúčania najkupovanejších zliav

8 Precision a Recall testovanie

Testovanie jednotlivých metód odporúčania sa realizuje v balíku `sk.zatko.vi.recommender.test`.

Najskôr získame množinu používateľom, pre ktorých použijeme na testovanie. Na to slúži trieda `RandomUserGenerator`, ktorá dokáže získať zoznam používateľov z testovacej množiny. Pre potreby testovania a jeho väčšiu relevantnosť vyberáme používateľom podľa počtu unikátnych dealov, ktoré si v testovacom období kúpili.

Zvolili sme testovanie pre 1000 používateľov, pričom ich vyberáme nasledovne:

Počet vybraných používateľov	Počet zľiav kúpených v testovacom období
300	2
300	3
250	4
100	5
30	6
20	7

Obrázok 6 – Rozdelenie používateľov pri testovaní

Testovanie jednotlivých metód odporúčania prebieha tak, že sa pre každého používateľa nájdu všetky testovacie dealy, ktoré si kúpil. Keďže potrebujeme pre potreby odporúčania **id aktuálneho dealu a aktuálny dátum**, tak si tieto údaje **získame z prvého kúpeného dealu používateľa**.

Potom prebehne vygenerovanie zoznamu odporúčaných zliav. Ich zoznam sa následne porovná so zoznamom používateľom kúpených dealov – okrem prvého. Pre vyhodnotenie konkrétnej odporúčacej metódy sa počítatjú 2 metriky:

- **Precision** P = tp/(tp + fp)
- **Recall** R = tp/(tp + fn)

tp – počet položiek zo zoznamu odporúčaní, ktoré si používateľ naozaj kúpil
fn – počet používateľom kúpených zliav - tp - 1 (prvý neporovnávame)
fp – počet odporúčaných položiek - tp

Tieto testy sme aplikovali na všetkých 5 implementovaných spôsobov odporúčania, pričom sme vykonali merania pre 5 unikátnych vzoriek po 1000 používateľov. Výsledky zaznamenali do tabuliek.

		5 odporúčaní	10 odporúčaní	25 odporúčaní
SimpleContentBasedRecommender	Precision	1,299%	1,358%	2,990%
	Recall	0,800%	0,440%	0,367%
UserHistoryContentBasedRecommender	Precision	1,524%	2,144%	3,819%
	Recall	0,960%	0,623%	0,504%
CombineContentBasedRecommender	Precision	1,850%	2,076%	4,177%
	Recall	1,120%	0,580%	0,525%
CollaborativeRecommender	Precision	0,000%	0,000%	0,000%
	Recall	0,000%	0,000%	0,000%
PopularityRecommender	Precision	7,835%	8,518%	15,163%
	Recall	4,260%	2,450%	1,821%

Obrázok 7 – Testovanie odporúčaní pomocou metrík precision a recall (1. meranie)

		5 odporúčaní	10 odporúčaní	25 odporúčaní
SimpleContentBasedRecommender	Precision	0,783%	1,783%	3,865%
	Recall	0,545%	0,460%	0,446%
UserHistoryContentBasedRecommender	Precision	1,726%	2,087%	3,715%
	Recall	1,027%	0,623%	0,513%
CombineContentBasedRecommender	Precision	1,592%	2,178%	5,449%
	Recall	0,965%	0,550%	0,628%
CollaborativeRecommender	Precision	0,000%	0,000%	0,000%
	Recall	0,000%	0,000%	0,000%
PopularityRecommender	Precision	6,714%	9,712%	15,181%
	Recall	3,860%	2,850%	1,775%

Obrázok 8 – Testovanie odporúčaní pomocou metrík precision a recall (2. meranie)

		5 odporúčaní	10 odporúčaní	25 odporúčaní
SimpleContentBasedRecommender	Precision	1,258%	1,865%	3,195%
	Recall	0,705%	0,535%	0,351%
UserHistoryContentBasedRecommender	Precision	1,893%	2,431%	4,321%
	Recall	0,973%	0,750%	0,536%
CombineContentBasedRecommender	Precision	2,128%	2,635%	5,442%
	Recall	0,965%	0,765%	0,592%
CollaborativeRecommender	Precision	0,000%	0,000%	0,000%
	Recall	0,000%	0,000%	0,000%
PopularityRecommender	Precision	6,862%	9,487%	14,973%
	Recall	3,820%	2,690%	1,833%

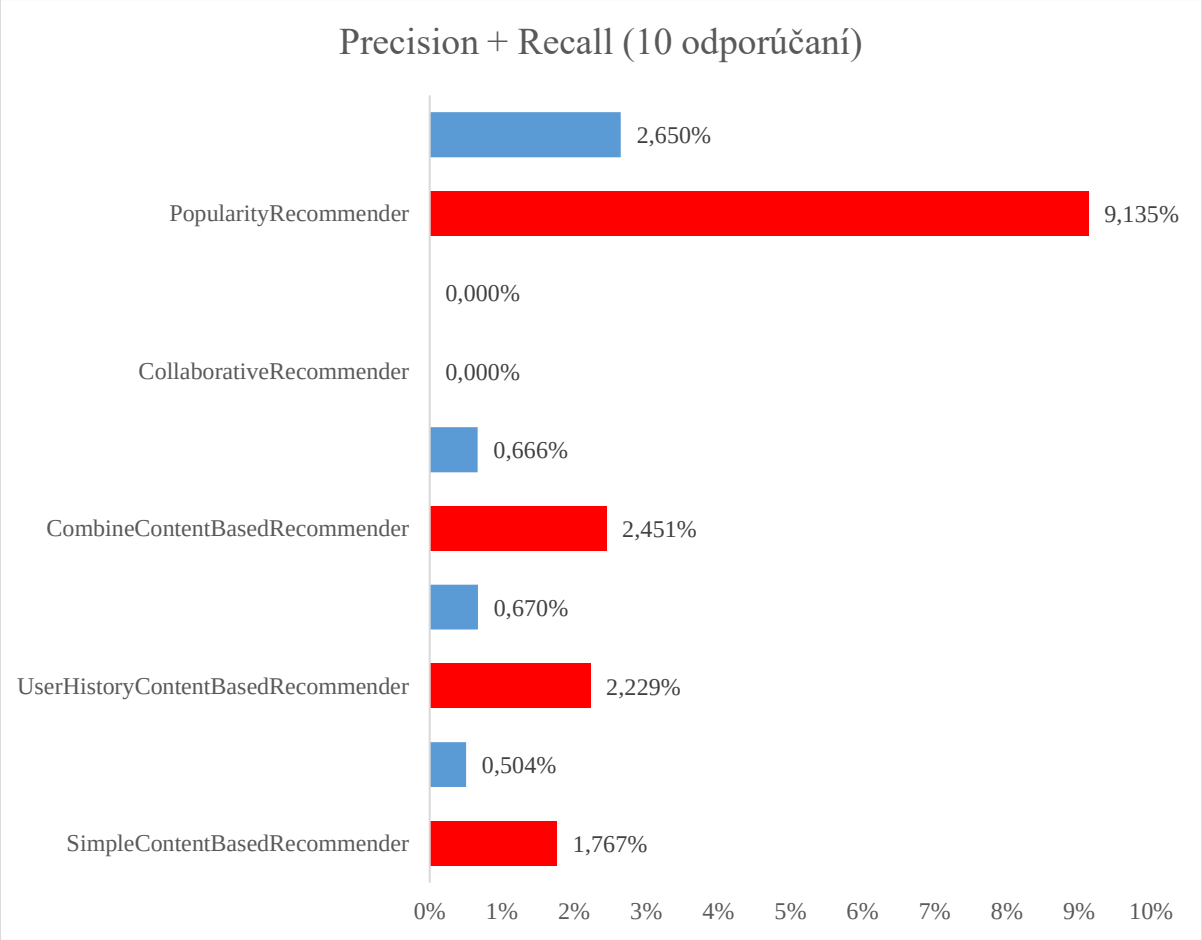
Obrázok 9 – Testovanie odporúčaní pomocou metrík precision a recall (3. meranie)

		5 odporúčaní	10 odporúčaní	25 odporúčaní
SimpleContentBasedRecommender	Precision	1,473%	1,868%	3,775%
	Recall	0,825%	0,546%	0,454%
UserHistoryContentBasedRecommender	Precision	1,596%	1,855%	3,884%
	Recall	0,993%	0,583%	0,528%
CombineContentBasedRecommender	Precision	1,973%	2,259%	5,318%
	Recall	1,105%	0,645%	0,641%
CollaborativeRecommender	Precision	0,000%	0,000%	0,000%
	Recall	0,000%	0,000%	0,000%
PopularityRecommender	Precision	7,247%	8,729%	17,431%
	Recall	4,120%	2,600%	1,970%

Obrázok 10 – Testovanie odporúčaní pomocou metrík precision a recall (4. meranie)

		5 odporúčaní	10 odporúčaní	25 odporúčaní
SimpleContentBasedRecommender	Precision	1,181%	1,958%	3,233%
	Recall	0,625%	0,540%	0,386%
UserHistoryContentBasedRecommender	Precision	1,872%	2,626%	4,478%
	Recall	0,980%	0,770%	0,522%
CombineContentBasedRecommender	Precision	1,324%	3,110%	5,143%
	Recall	0,745%	0,790%	0,600%
CollaborativeRecommender	Precision	0,000%	0,000%	0,000%
	Recall	0,000%	0,000%	0,000%
PopularityRecommender	Precision	6,769%	9,227%	16,348%
	Recall	3,900%	2,660%	1,837%

Obrázok 11 – Testovanie odporúčaní pomocou metrík precision a recall (5. meranie)



Obrázok 12 – Priemerné hodnoty Precision (červená) a Recall (modrá)

Na výsledkoch vidno, že najlepším spôsobom odporúčania je odporúčanie najpredávanejších zliav na základe geografickej vzdialenosti. Naopak, odporúčanie na základe podobnosti používateľom dosahuje prakticky nulové výsledky.

9 Testovanie počtu výsledkov odporúčania

Keďže jednotlivé výsledky odporúčania sú závislé od dát, ktoré máme k dispozícii, tak je možné, že niektoré odporúčania vracajú menej výsledkov ako požadujeme. Toto bude pravdepodobne problém odporúčania na základe podobnosti používateľom. Preto sme sa rozhodli odmerať, koľko výsledkov odporúčanie vracia. Výsledky sme rozdelili do 3 kategórii:

- Odporúčanie vracia 0 výsledkov
- Odporúčanie vracia aspoň 1 výsledok, ale nie požadovaný počet
- Odporúčanie vracia požadovaný počet výsledkov

Tieto testy sme podobne ako v predchadzajúcom prípade aplikovali na všetkých 5 implementovaných spôsobov odporúčania, pričom sme vykonali merania pre 5 unikátnych vzoriek po 1000 používateľov. Výsledky zaznamenali do tabuliek.

	Počet výsledkov	5 odporúčaní	10 odporúčaní	25 odporúčaní
SimpleContentBasedRecommender	0	16	19	18
	1 až n-1	14	40	110
	n	970	941	872
UserHistoryContentBasedRecommender	0	291	289	288
	1 až n-1	0	3	20
	n	709	708	692
CombineContentBasedRecommender	0	4	4	6
	1 až n-1	5	7	27
	n	991	989	967
CollaborativeRecommender	0	334	344	355
	1 až n-1	306	519	665
	n	630	137	0
PopularityRecommender	0	3	5	6
	1 až n-1	20	32	60
	n	977	963	934

Obrázok 13 – Testovanie počtu výsledkov odporúčaní (1. meranie)

	Počet výsledkov	5 odporúčaní	10 odporúčaní	25 odporúčaní
SimpleContentBasedRecommender	0	21	25	16
	1 až n-1	17	46	114
	n	962	929	870
UserHistoryContentBasedRecommender	0	292	286	298
	1 až n-1	5	5	14
	n	703	709	688
CombineContentBasedRecommender	0	6	9	6
	1 až n-1	10	3	32
	n	984	988	962
CollaborativeRecommender	0	334	343	343
	1 až n-1	308	521	656
	n	358	136	2
PopularityRecommender	0	7	5	8
	1 až n-1	23	21	69
	n	970	974	923

Obrázok 14 – Testovanie počtu výsledkov odporúčaní (2. meranie)

	Počet výsledkov	5 odporúčaní	10 odporúčaní	25 odporúčaní
SimpleContentBasedRecommender	0	17	27	20
	1 až n-1	15	44	118
	n	968	929	862
UserHistoryContentBasedRecommender	0	281	283	278
	1 až n-1	5	4	18
	n	714	713	704
CombineContentBasedRecommender	0	4	6	8
	1 až n-1	3	9	24
	n	993	985	968
CollaborativeRecommender	0	328	326	317
	1 až n-1	303	526	680
	n	369	148	3
PopularityRecommender	0	9	7	4
	1 až n-1	19	27	63
	n	972	966	933

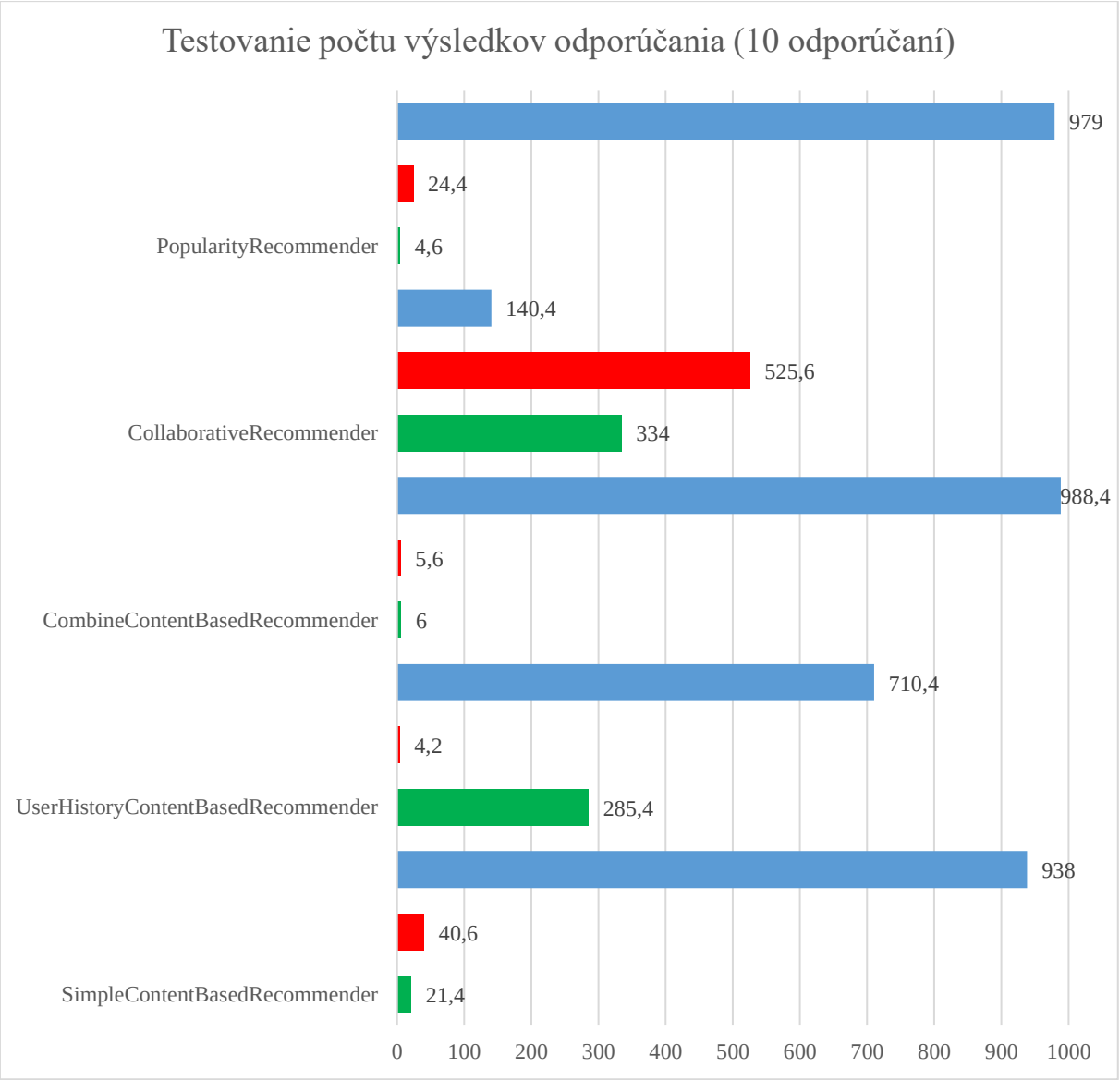
Obrázok 15 – Testovanie počtu výsledkov odporúčaní (3. meranie)

	Počet výsledkov	5 odporúčaní	10 odporúčaní	25 odporúčaní
SimpleContentBasedRecommender	0	22	17	16
	1 až n-1	18	42	104
	n	960	941	880
UserHistoryContentBasedRecommender	0	277	285	286
	1 až n-1	3	5	26
	n	720	710	688
CombineContentBasedRecommender	0	6	5	6
	1 až n-1	8	7	26
	n	986	988	968
CollaborativeRecommender	0	337	333	336
	1 až n-1	297	523	662
	n	366	144	2
PopularityRecommender	0	4	5	3
	1 až n-1	16	22	72
	n	980	973	925

Obrázok 16 – Testovanie počtu výsledkov odporúčaní (4. meranie)

	Počet výsledkov	5 odporúčaní	10 odporúčaní	25 odporúčaní
SimpleContentBasedRecommender	0	22	19	18
	1 až n-1	17	31	120
	n	961	950	862
UserHistoryContentBasedRecommender	0	277	284	278
	1 až n-1	1	4	24
	n	722	712	698
CombineContentBasedRecommender	0	6	6	5
	1 až n-1	7	2	28
	n	987	992	967
CollaborativeRecommender	0	327	324	329
	1 až n-1	308	539	669
	n	365	137	2
PopularityRecommender	0	10	1	4
	1 až n-1	13	20	59
	n	977	979	937

Obrázok 17 – Testovanie počtu výsledkov odporúčaní (5. meranie)



Obrázok 18 – Priemerné počty výsledkov odporúčania: požadovaný počet (modrá), nejaké výsledky (červená), žiadne výsledky (zelená)

Na výsledkoch vidíme, že pri každom spôsobe odporúčania dochádza k tomu, že sa nevráti požadovaný počet výsledkov. U odporúčania na základe podobnosti používateľov získame požadovaný počet výsledkov len v tretine prípadov. Toto s časti vysvetľuje nulové výsledky pri testovaní precision a recall.

10 Performance testovanie

Aby sme mohli komplexne zhodnotiť jednotlivé prístupy, tak sme vykonali testovanie ich časovej náročnosti. Každý spôsob odporúčania sme spustili 5krát pre 1000 používateľov, pričom sme sledovali nielen celkové ich časy ale aj časy čiastkových operácií. Uvedené výsledky sú v milisekundách (ms).

SimpleContentBasedRecommender					
	Meranie 1	**Meranie 2**	**Meranie 3**	**Meranie 4**	**Meranie 5**
prepareQuery	0,198	0,161	0,176	0,194	0,358
executeSearch	5,084	4,397	4,808	5,255	6,975
parseResults	0,068	0,058	0,069	0,070	0,049
Spolu	**5,350**	**4,616**	**5,053**	**5,519**	**7,382**

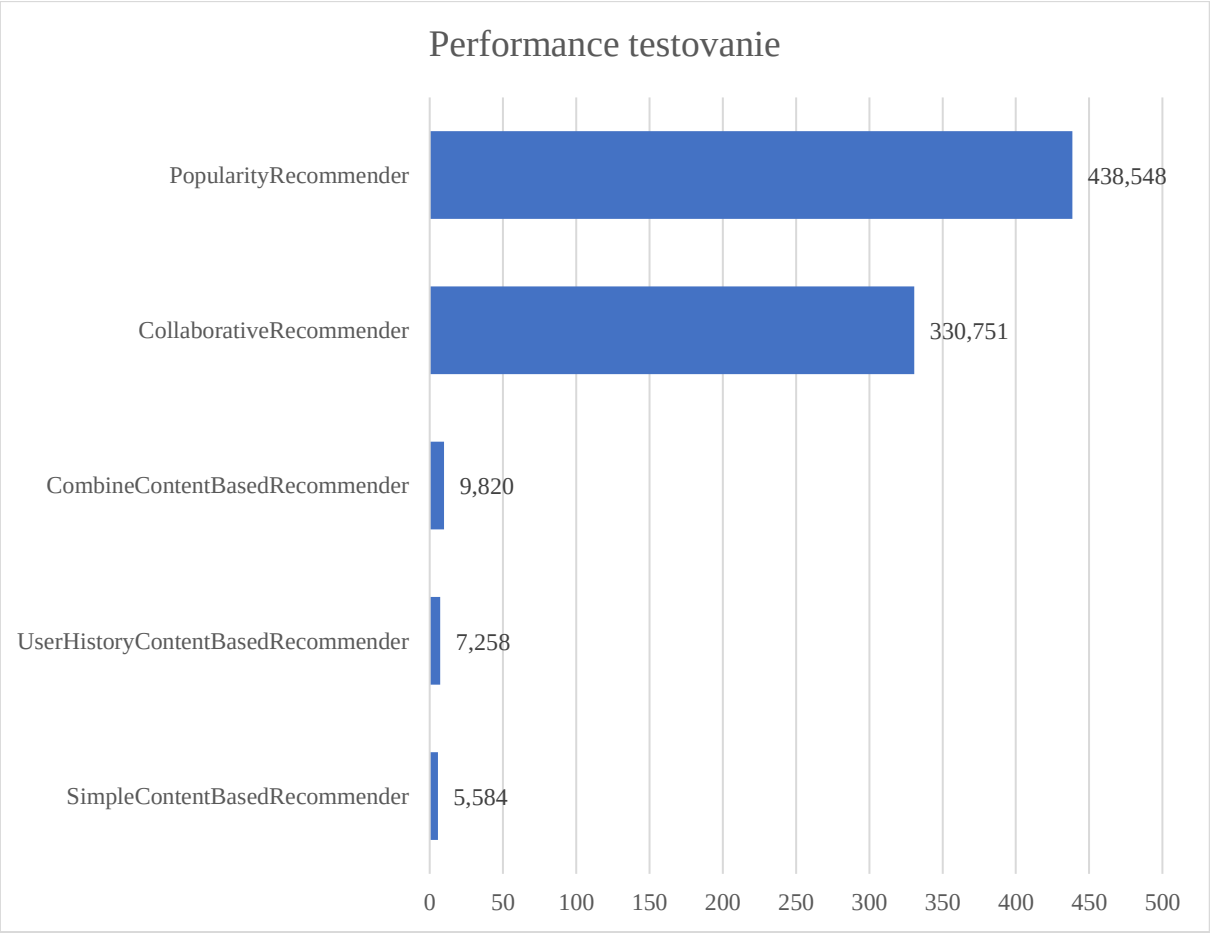
UserHistoryContentBasedRecommender					
	Meranie 1	**Meranie 2**	**Meranie 3**	**Meranie 4**	**Meranie 5**
prepareQuery	0,388	0,358	0,364	0,378	0,358
executeSearch	7,007	6,196	6,881	7,175	6,975
parseResults	0,034	0,039	0,046	0,041	0,049
Spolu	**7,429**	**6,593**	**7,291**	**7,594**	**7,382**

CombineContentBasedRecommender					
	Meranie 1	**Meranie 2**	**Meranie 3**	**Meranie 4**	**Meranie 5**
prepareQuery	0,536	0,509	0,480	0,559	0,504
executeSearch	9,355	9,016	9,088	9,583	9,147
parseResults	0,062	0,074	0,056	0,064	0,068
Spolu	**9,953**	**9,599**	**9,624**	**10,206**	**9,719**

CollaborativeRecommender					
	Meranie 1	**Meranie 2**	**Meranie 3**	**Meranie 4**	**Meranie 5**
prepareQuery	0,011	0,012	0,008	0,010	0,014
executeSearch	334,136	344,836	334,377	320,009	320,334
parseResults	0,003	0,000	0,000	0,004	0,001
Spolu	**334,150**	**344,848**	**334,385**	**320,023**	**320,349**

PopularityRecommender					
	Meranie 1	**Meranie 2**	**Meranie 3**	**Meranie 4**	**Meranie 5**
prepareQuery	0,019	0,010	0,012	0,012	0,010
executeSearch	451,722	433,021	452,689	428,975	426,258
parseResults	0,003	0,002	0,000	0,002	0,003
Spolu	**451,744**	**433,033**	**452,701**	**428,989**	**426,271**

Obrázok 19 – Testovanie časovej náročnosti jednotlivých spôsobov odporúčania. Uvedené výsledky sú v milisekundách (ms).



Obrázok 20 – Testovanie časovej náročnosti jednotlivých spôsobov odporúčania. Uvedené výsledky sú v milisekundách (ms).

Odporúčania na základe obsahu, ktoré používajú dopyty na Elasticsearch trvajú do 10 ms. Ale odporúčania využívajúce komplikované dopyty na relačnú databázu trvajú niekoľkonásobne dlhšie.

11 Zlepšovanie algoritmov odporúčania

Najlepšie výsledky dosiahlo odporúčanie najkupovanejších zliav v blízkosti aktuálne zobrazovanej zľavy. Vstupom tohto spôsobu odporúčania je vzdialenosť v rámci ktorej odporúčací systém vracia výsledky. Rozhodli sme sa otestovať rôzne vzdialenosti (5km, 10km, 25km, 50km, 100km, 150km, 200km) na vzorke 1000 používateľov, aby sme zistili, aká hodnota vracia najlepšie výsledky metriky precision. Najlepšie výsledky sme namerali pre hodnoty 100 a 150 km

	Vzdialenosť	Precision
PopularityRecommender	5 km	7,185%
	10 km	8,986%
	25 km	11,113%
	50 km	10,644%
	100 km	14,078%
	150 km	14,228%
	200 km	13,453%

Obrázok 21 – Testovanie odporúčania najpopulárnejších zliav pre rôzne vzdialenosti

12 Zhodnotenie a záver

Na základe poskytnutých dát sme implementovali **5 spôsobov odporúčania** zľavových ponúk pre používateľov portálu zlavadna.sk. Tieto spôsoby sme pomocou metrík **precision a recall** vyhodnotili.

Ako najlepší spôsob odporúčania sa ukázalo **odporúčanie najpredávanejších ponúk** v blízkosti aktuálne prezeranej ponuky. Pri tomto spôsobe sme dokonca zisťovali, pre akú hodnotu vzdialenosti vracia najlepšie výsledky. Podarilo sa nám zistiť, že pre vzdialenosť 100 km (prípadne 150 km) vracia algoritmus lepšie výsledky ako pre pôvodných 50 km. Žiaľ, tento spôsob bol zároveň najviac časovo náročný. Jedno odporúčanie trvalo 400 – 500 ms.

Druhým najlepším spôsobom bolo hybridné **odporúčanie na základe podobnosti obsahu**, ktoré spájalo aktuálne prezeranú ponuku s históriou ponúk, ktoré si používateľ kúpil. Pri tomto prístupe boli síce hodnoty len štvrtinové oproti predchádzajúcemu prístupu, ale časová náročnosť bola cca 10 ms, čo je podstatný rozdiel.

Zaujímavým faktom je, že na prvých dvoch miestach sa umiestnili algoritmy, ktoré používajú úplne odlišné dáta. Odporúčací systém na základe populárnosti zliav využíva informácie o kúpených zľavách všetkými používateľmi a geo-dáta. Naopak, hybridný systém založený na podobnosti obsahu využíva históriu konkrétneho používateľa a textové zhody. Ďalšou zaujímavosťou je, že odporúčanie najpredávanejších ponúk v blízkosti aktuálne prezeranej ponuky vracia najlepšie výsledky pre vzdialenosť 100 až 150km.

Pre každý spôsob odporúčania sme zisťovali, či vracia požadovaný počet výsledkov. Zistili sme, že pri každom spôsobe odporúčania dochádza k tomu, že sa nevráti požadovaný počet výsledkov. U odporúčania na základe podobnosti používateľov získame požadovaný počet výsledkov len v tretine prípadov. Toto je spôsobené hlavne nedostatkom dát, keďže v dátach je obrovské množstvo používateľom, ktorý v tréningovej množine nemajú žiaden záznam, prípadne len jeden.

Tento problém s nedostatkom výsledkov má negatívny vplyv na hodnoty precision a recall. Pri nedostatku výsledkov navrhujeme vrátiť používateľovi najpredávanejšie zľavy bez ohľadu na ich geografickú polohu (pretože pri využití geodát môže vzniknúť nedostatok výsledkov – ak je ponuka niekde v zahraničí).

Na základe testovania časovej náročnosti sme zistili, že najväčšiu réžiu zaberajú SQL dopyty na Postgres databázu. Tie v prípade odporúčania najpredávanejších zliav v geografickej blízkosti trvajú 400 – 500ms, čo je už značný čas. Tento problém by mohlo značne zredukovať to, keby sme si pri každej zľavovej ponuke evidovali, koľko unikátnych používateľov si ju zakúpilo.

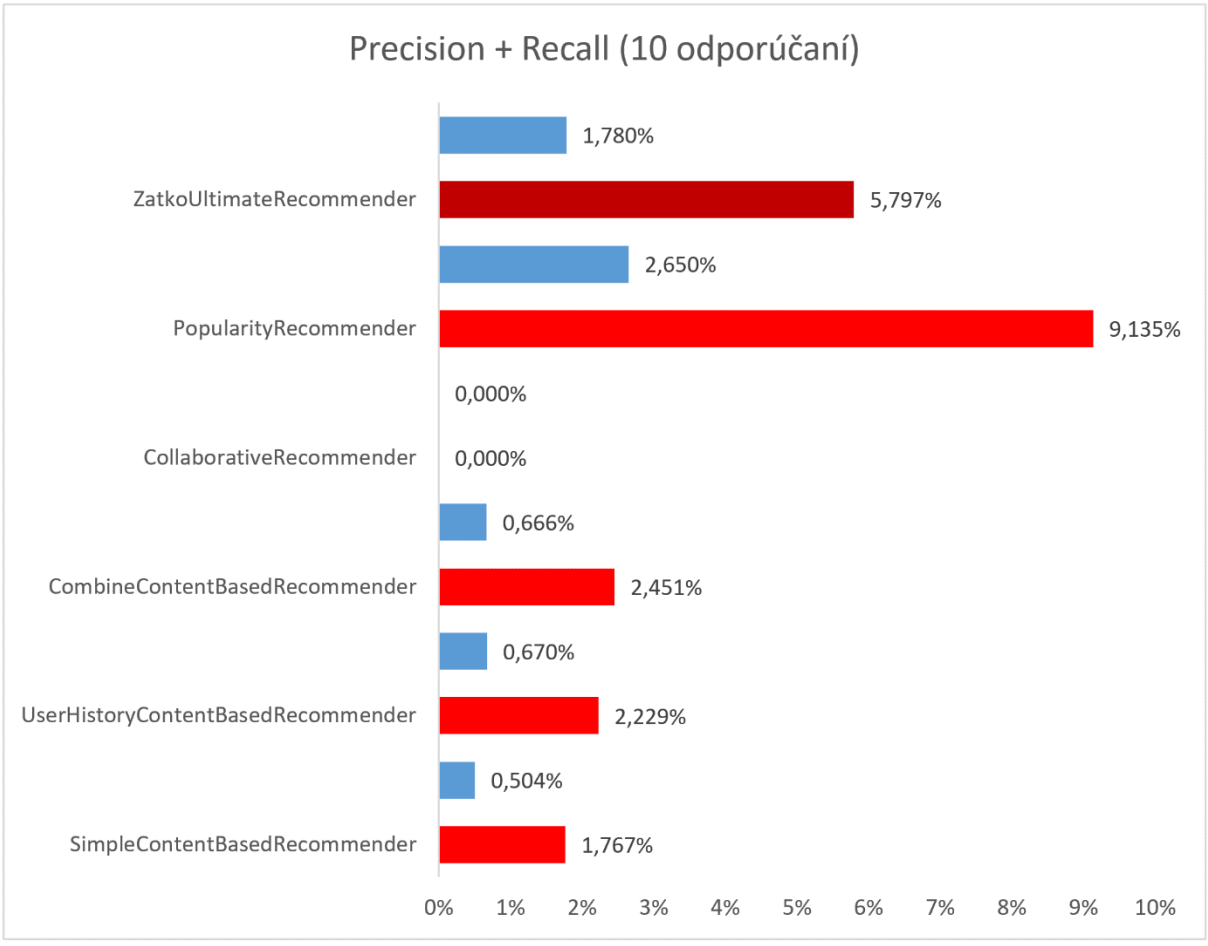
ZatkoUltimateRecommender

Na základe meraní 5 predchádzajúcich spôsobov odporúčanie sme vytvorili odporúčanie využívajúce dáta, ktoré sa ukázali ako relevantné z pohľadu úspešnosti odporúčania. Vytvorili sme tak odporúčanie s príznačným názvom ZatkoUltimateRecommender. Tento kombinoval viaceré prístupy, pričom ich výsledky prispievali do skóre jednotlivých ponúk. Po vypočítaní celkového skóre sme ponuky zoradili a vrátili tie s najvyšším skóre.

Algoritmus funguje nasledovne:

1. Všetky ponuky z tréningovej množiny s platnosťou aj v testovacom období sme zoradili podľa počtu zakúpení. Najlepšej ponuke sme priradili skóre 100, postupne skóre klesalo až ku 20. Ponuky nezakúpené v testovacom období dostali skóre 1.
2. Potom sme vygenerovali odporúčanie ponúk `UserHistoryContentBasedRecommender.` (bez použitia limitu). Tieto ponuky dostali násobiteľ 10 až po 5 – podľa poradia v odporúčaní. Skóre ponúk sme vynásobili prideleným násobiteľom.
3. Následne sme skóre ponúk, ktoré boli vytvorené firmou od ktorej mal používateľ už niečo kúpené, vynásobili číslom 8.

Takto vypočítané skóre určilo, aké ponuky sa odporučili používateľovi. Tento prístup priniesol prekvapivo dobré výsledky, čo sa odrazilo aj pri meraní. Tento prístup mal viac než dvojnásobné hodnoty precision (5,769%) ako 2. najlepší prístup (CombineContentBasedRecommender – 2,451%). (PopularityRecommender nerátame, lebo používa testovacie dáta).



Obrázok 22 - Priemerné hodnoty Precision (červená) a Recall (modrá)

Súťaž VI Challenge

Na základe implementovaných odporúčacích systémom sa konala súťaž VI Challenge na FIIT STU pre všetkých študentov predmetu VI. Cieľom bolo vygenerovať 10 odporúčaní 2200 rôznym používateľom. Ak sa ponuka zakúpená používateľom nachádzala medzi 10 odporúčaniami, súťažiaci získal 1 bod.

ZatkoUltimateRecommender obsadil 3. miesto s počtom bodov 192 (8,73%).

Výsledky:

1. 288 bodov
2. 221
3. Zatko 192
4. 155
5. 145
6. 143
7. 135
8. 131
9. 115 (p@10 = 0.0054)
10. 117 (p@10 = 0.0053)
11. 107
12. 92
13. 73