Jihočeská univerzita
Přírodovědecká fakulta
Ústav aplikované informatiky

Bakalářská práce

Doporučovací systém pro e-commerce

Autor: Lukáš Kortus
Školitel: Ing. Ladislav Beránek CSc.

České Budějovice 2013

NÁZEV:

Kortus L., 2013 : Doporučovací systém pro e-commerce [Recommendation System for e-commerce] - ?.p, Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic

ABSTRAKT:

Na základě potřeb uživatele, který se rozhoduje mezi mnoha produkty či službami a nechce se zabývat procházením informací o všech těchto dostupných objektech, vzniká potřeba tzv. doporučovacích systémů, které se pokouší nabízet objekty, které by pro něho mohly být zajímavé. Cílem je tedy navrhnout vhodné způsoby zpracování informací o chování zákazníků na serveru a poté navrhnout a realizovat doporučovací systém, který bude využívat takto zpracovaných informací o chování zákazníků a bude poskytovat doporučení pro uživatele hledající informace na serveru tohoto e-commerce. V teoretické části práce se dozvídáme informace o uživatelských preferencích a doporučovacích systémech. V části praktické navrhujeme a programujeme modul pro systém e-commerce, který není omezen problémem studeného startu. Dále se zde zabýváme problémem chybějících údajů. Správnost řešení tohoto problému poté testujeme na datech reálných uživatelů.

KLÍČOVÁ SLOVA:

uživatelská preference, e-commerce, doporučovací systémy

TITLE:

Kortus L., 2013 : Doporučovací systém pro e-commerce [Recommendation System for e-commerce] - ?.p, Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic

SUMMARY:

Based on the needs of the user, who decides between many products or services and does not want to deal with the passing of information available on all of these objects, there is a need for the recommendation systems that try to offer objects that would be for him to be interesting. Objective is therefore to propose appropriate ways of processing information about behavior of customers on the server and then design and implement a recommendation system that will use such processed information about the behavior of customers and will provide recommendations for users seeking information on the server of this e-commerce. In the theoretical part, we learn information about user's preferences and recommender systems. In the practical part we propose module for e-commerce, which is not limited to the cold start problem. Furthermore, we are concerned with the problem of missing data. The accuracy of the solution to this problem then tested on the data of real users.

KEYWORDS:

user preferences, e-commerce, recommendation systems

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

České Budějovice 26. 4. 2013 …..…….

Obsah

1 Úvod

V dnešní době, ve které je mnoho domácností připojeno k celosvětové síti internetu, mají lidé možnost nakupovat a hledat zboží v systémech e-commerce z pohodlí svého domova, bez potřeby chodit nebo jezdit mnoho kilometrů do kamenných prodejen. Toto usnadňuje život zákazníkovi, ale i provozovatelům obchodů a služeb díky výpočetní technice, která zjednodušuje procesy, jež jsou spojeny s obchodováním, a umožňuje zvyšovat efektivitu procesů s ním spojených. E-commerce značí veškeré obchodní transakce realizované za pomocí internetu a dalších elektronických prostředků. Mezi e-commerce patří hlavně elektronický prodej zboží a služeb, ale také poprodejní služby, elektronická výměna nejrůznějších dat, vedení bankovních účtů, obchodní aukce a jiné.

Výhodou těchto systémů pro zákazníka je, že jsou dostupné 24 hodin denně z jakéhokoliv místa, kde je připojení na internet, dále pak možnost rychle a jednoduše srovnávat výrobky a služby, které nabízejí jiné obchody, čímž se výrazně šetří čas. Nevýhody, které z internetového nakupování pro zákazníky plynou, jsou nebezpečí zneužití osobních údajů, narušení soukromí, nedostatečné zabezpečení plateb, nemožnost kupovaný výrobek předem vidět.

Ve chvíli, kdy zákazník navštíví například elektronický obchod, jsou mu nabídnuty různé druhy zboží, které si může prohlédnout a popřípadě koupit, jako v kamenném obchodě. Výhodou elektronických obchodů ale je, že mohou jednoduše zákazníkovi rovnou doporučit zboží, které by bylo podle jeho představ a které by se mu mohlo líbit díky doporučovacím systémům. Doporučovací systém je taková aplikace, která na základě uživatelovi zpětné vazby doporučuje položky. Toto umožňuje přesně cílit zboží na jednotlivé zákazníky a zvýšit tím obrat a tedy i zisky. Tyto funkce ale nepřinášejí výhody jen majitelům obchodů, ale i zákazníkům, kteří získají lepší přehled o nabízených výrobcích a nemusejí složitě hledat výrobek, který by je mohl zajímat.

2 Cíle práce

Jak již bylo uvedeno, doporučovací systémy jsou jedním z klíčových prvků online obchodů a e-shopů a dalších online systémů založených na infrastruktuře Internetu. S ohledem na některé problémy související s problematikou doporučovacích systémů, jsou cíle této práce následující:

1. Zmapovat typy doporučovacích systémů a jejich funkce.
    a) Uvést příklady použití doporučovacích systémů u konkrétních příkladů, provést jejich stručný rozbor.
    b) Popsat mechanismy, které používají doporučovací systémy.

2. Navrhnout modul pro systém e-commerce, který bude zpracovávat informace, které zákazník poskytl a zároveň mu bude, podle těchto informací, doporučovat a poskytovat informace o položkách v tomto daném systému e-commerce.
    a) Provést analýzu a popsat ji pomocí UML diagramu.
    b) Navrhnout způsob řešení případu, kdy uživatel zadá neúplná data.
    c) Ověřit navržený způsob řešení problému neúplných dat.

3 Uživatelská preference

Uživatelská preference je obvykle definována jako funkce $PU(o): OxU \rightarrow [0,1]$, která pro konkrétního uživatele U a objekt o z množiny objektů O vrací míru „oblíbenosti“ objektu u uživatele [1]. Uživatel tím vyjadřuje, co se mu líbí a co by si popřípadě rád koupil. Vzhledem k tomu, že většina lidí se od sebe liší, budou se lišit i jejich preference, a proto se musí na každého uživatele pohlížet zvlášť. Existují dva druhy preferencí a to krátkodobá preference a dlouhodobá preference.

V úvahu se musí brát i to, že uživatel se postupem času vyvíjí. Mění se jeho názory i požadavky. Nejdříve chce co nejvyšší rozlišení, pak spíše menší hmotnost

a nakonec největší rozsah zoomu. Záleží i na aktuálním rozpoložení uživatele. Při stresu nebo při časové tísni se člověk chová jinak [2].

Uživatel při interakci se systémem e-commerce zanechává v systému zpětnou vazbu, ať už vědomě (hodnocením položek, vyplněním dotazníku nebo uživatelského profilu informacemi, jako jsou věk, bydliště, vzdělání nebo pohlaví) nebo nevědomě (pohyb na stránce, doba pobytu na stránce, …). Na základě těchto informací je pak možné činit závěry či předpoklady o uživatelově preferenci vůči některému objektu.

3.1 Krátkodobá preference

Krátkodobá preference uživatele představuje preferenci k objektům na základě aktuálního cíle. Například, je-li aktuální cíl uživatele koupit levný netbook pro manželku, pak preference stolního počítače bude blízká 0 i přesto, že uživatel je v jiné situaci (v jiném čase) vyhledává [1].

3.2 Dlouhodobá preference

Dlouhodobá preference vyjadřuje obecnější pravidla, kterými se uživatel většinou řídí. Například preference nižší ceny, preference zrcadlovek před kompaktními fotoaparáty, preference značky Škoda před ostatními automobily atd. [1].

3.3 Předmět preference

Předmět preference je vlastnost toho, o čem se uživatel rozhoduje. Každý uživatel má jinou představu o atributech. Atributy objektu se nemění (kromě ceny) a každý je nějakým způsobem důležitý. Když si uživatel chce koupit výkonný notebook a na ceně mu nezáleží, v tomto případě je pro systém důležitější parametr výkonu, než parametr ceny. Atributy mohou být [2]:

1. Nominální – Barva, výrobce, typ obrazovky, …

2. Numerické – Hmotnost, rozlišení, váha, výdrž baterie, rozměry, …

3. Speciální – Množina hodnot některého z atomických typů (herci ve filmu)
   – Těžko zachytitelné atributy (tvar, oblost, zvuk, …)

Tyto atributy je potřeba před procesem doporučení upravit (Diskretizace, Normalizace, …), aby lépe odpovídali vnímání uživatele.

4 Doporučovací systémy

Cílem doporučovacích systémů (Recommender Systems, dále jen RS) je tedy vytvářet smysluplné doporučení položek nebo produktů pro uživatele, které by je mohli zajímat a mohou být nějakým způsobem užitečné.

Příklady provozu RS v reálném světě jsou například doporučení knih na stránkách internetového obchodu Amazon, filmu na Netflix (internetová televize), přátel na sociální síti Facebook, nebo videí na YouTube a každý, kdo využívá tyto služby, se s tímto již setkal. Bohužel Netflix není pro Českou republiku dostupný, to znamená, že český uživatel se s touto službou setkat nemůže [3].

Návrhy doporučení jsou obvykle individualizované, tvořené pro každého uživatele jedinečně, takže každý uživatel dostane jiný návrh položek. Existují, ale i neindividualizované návrhy, které jsou mnohem jednodušší na vytvoření. Jsou nejčastěji k nalezení v novinách, či časopisech. Typickým příkladem takovýchto návrhů je nejlepší desítka knih, nebo DVD, podle prodeje, nebo hodnocení zákazníků. Tyto návrhy mají určitě své využití, ale většinou se jimi nezabývají žádné RS výzkumy [4].

Konstrukce modulu pro RS závisí na oblasti a konkrétní charakteristice dostupných dat. Například filmoví diváci Netflix často poskytují hodnocení na stupnici od 1 (nelíbí) do 5 (líbí). Takové datové zdroje zaznamenávají interakci kvality mezi uživateli a položkami. Navíc systém může mít přístup ke specifickým atributům, jako je demografie nebo popis produktu. Doporučovací systémy se liší ve způsobu, jakým analyzují tyto zdroje k vytvoření představy o příbuznosti mezi uživatelem a položkou, která může být použita k identifikaci párů [3].

4.1 Příklady doporučovacích systémů ve světě

4.1.1 Internetový obchod - Amazon.com

Společnost Amazon je známá jako největší a nejpopulárnější online obchod na světě. Počínaje online knihami, hudbou a filmy. Poté společnost rozšířila svojí činnost do různých segmentů, včetně prodeje hraček, elektronických strojů, oblečení, léků, dokonce i potravin. Posledním krokem bylo zavedení vlastního, velice populárního zařízení pro čtení online knih se jménem Kindle. Společnost má sídlo v Seatlu, USA [5].

Obrázek 1: Doporučení internetového obchodu Amazon.com

Uživatelům Amazon nabízí hodnocení všech nabízených produktů, díky čemuž se stal populární zejména při prodeji knih a hudebních nosičů. Ke všem produktům je také možné psát krátké recenze a podělit se tak s ostatními uživateli o vlastní zkušenosti. Výsledky doporučení se na tomto webu nachází pod podrobnostmi o produktech. Tento systém využívá metody kolaborativního filtrování [6].

4.1.2 Stránky pro sdílení videí – YouTube.com

YouTube jsou stránky, na které mohou uživatelé nahrát, sdílet a zobrazovat videa. Stránky byly vytvořené třemi bývalými zaměstnanci PayPal v únoru 2005 [7].

Společnost sídlí v San Bruno, Kalifornie, a využívá technologie Adobe Flash Video k zobrazení široké škály video obsahu vytvořeného uživateli, včetně filmových klipů, televizních klipů a hudebních videí stejně jako amatérského videoobsahu, jako jsou blogová a krátká originální videa. Většina obsahu na YouTube je nahrávána jednotlivci, i když mediální korporace, včetně CBS, BBC, VEVO a další organizace nabízejí některé ze svých materiálů prostřednictvím tohoto webu, v rámci programu partnerství s YouTube [7].

Neregistrovaní uživatelé mohou jen videa sledovat, zatímco registrovaní uživatelé mohou videa i nahrát a to neomezený počet. Videa, která obsahují potenciálně urážlivý obsah, jsou dostupná pouze registrovaným uživatelům, kteří mají věk 18 a více. V listopadu 2006, LLC YouTube koupila společnost Google Inc. a nyní působí jako její dceřiná společnost [7].

YouTube zahrnuje doporučovací systém, který uživateli doporučí pro něj zajímavá videa. Na svých stránkách uvádí, co doporučená videa mohou obsahovat. Doporučená videa mohou obsahovat především videa od několika tisíc partnerů YouTube. Mohou ale také zahrnovat vybraná videa uživatelů, která jsou aktuálně populární nebo která byla předtím uvedena v modulu Vybraná videa. Tato videa

se v průběhu dne automaticky střídají. Doporučená videa nejsou reklamy ani placená propagace produktů [8].

Toto doporučení se nachází na stránce v pravém sloupci.

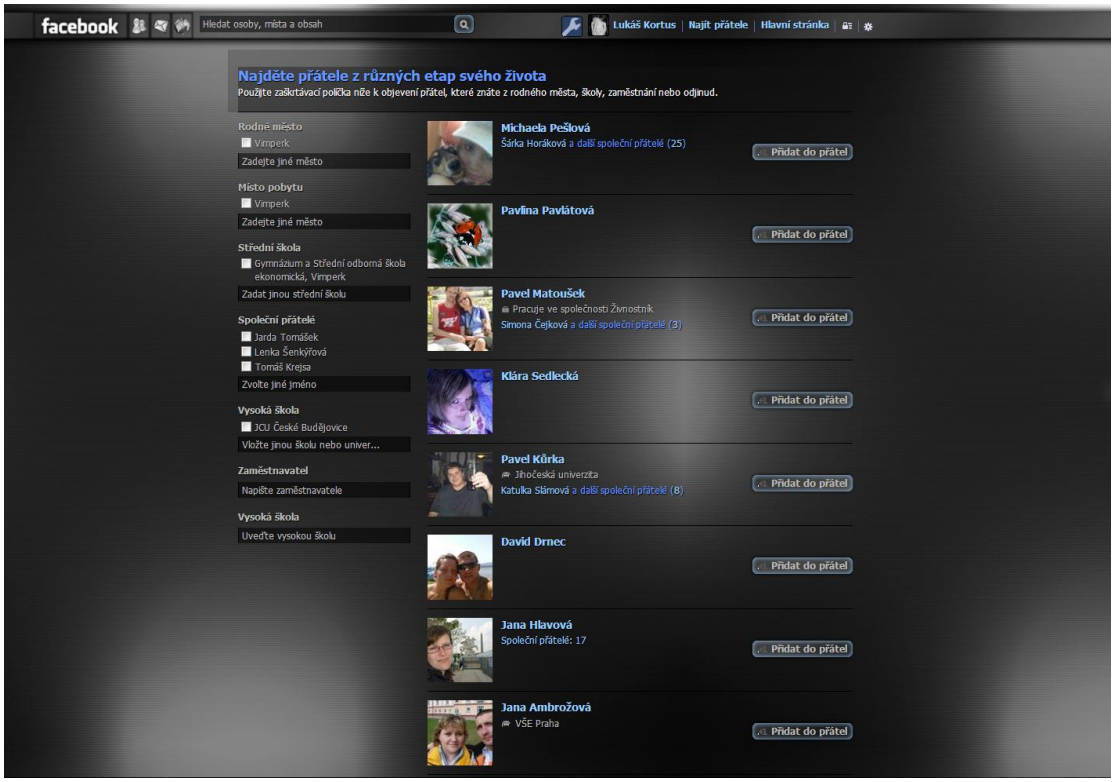
Obrázek 2: Doporučená videa na stránkách YouTube.com [27]

4.1.3 Sociální síť – Facebook.com

Facebook je dnes jedna z největších a nejrozšířenějších sociálních sítí na světě. Byl založen bývalým studentem Harvardovy univerzity Markem Zuckerbergem. Původně byl tento systém omezen jen pro studenty Harvardovy univerzity pod doménou *thefacebook.com*. Během dvou měsíců byl rozšířen na některé další

univerzity, které patří do sportovního sdružení osmi elitních soukromých univerzit na severovýchodě USA do tzv. Ivy League a již do konce roku byly připojeny další univerzity. Nakonec byl přístup otevřen pro všechny uživatele s univerzitní e-mailovou adresou (.edu, ac.uk, ...) nebo pro některé zahraniční schválené univerzity. V Česku k prvním otevřeným vysokým školám patřila Masarykova univerzita. Od 27. února 2006 se začaly do systému připojovat některé nadnárodní obchodní společnosti. Od 11. srpna 2006 se může, dle licence používání, připojit kdokoli starší 13 let. Uživatelé se v systému mohou přidat k různým skupinám uživatelů, kteří působí například v rámci jedné školy, firmy nebo geografické lokace [9].

Pro všechny registrované uživatele Facebook nabízí funkci s názvem Návrh přátelství. Je to funkce, která uživatelům navrhuje ostatní uživatele k přátelství. Doporučování přátel funguje tak, že systém nejprve doporučí lidi, které mají v přátelích vaši přátelé a zobrazí i počet společných přátel. Dále je schopen doporučit lidi podle etap života, kde uživatel použije zaškrtávací políčka k nalezení přátel, které zná například z rodného města, školy, zaměstnání nebo odjinud.



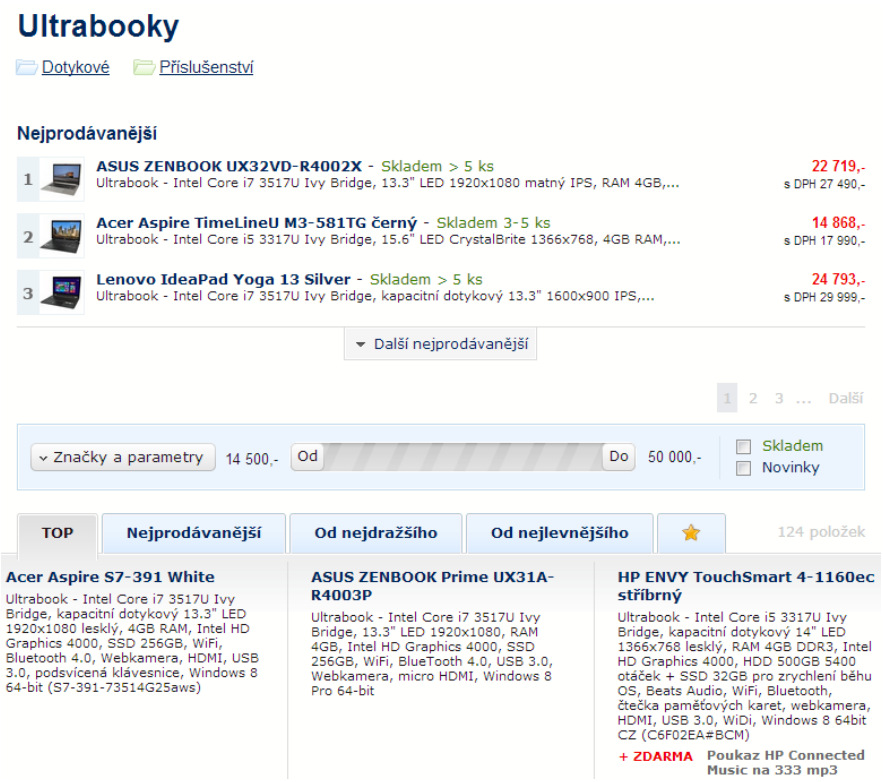
Obrázek 3: Doporučení přátel na Facebook.com

4.1.4 Internetový obchod - Alza.cz

Alza.cz a.s. je český obchod s počítači a spotřební elektronikou pro celou řadu zákazníků. Oficiální začátek společnosti se datuje k 29. listopadu 1994, kdy ji zahájil Aleš Zavoral jako fyzická osoba pod značkou Alzasoft. Společnost působí v České republice a na Slovensku s centrálou umístěnou v Praze. Na jaře 1998 vznikly první webové stránky společnosti, které ještě nesloužily k nákupu. V létě 1998 Alzasoft otevřel prodejnu v Dělnické ulici. V roce 2000 se přestěhoval do větších prostor v Jateční ulici. Vznikl elektronický obchod propojený s webovými stránkami. V roce 2006 firma prošla rebrandingem (změna obchodní značky) a přejmenovala na Alza.cz [10].

Obchod zahrnuje jednoduchý doporučovací systém, který vyhledá nejprodávanější položky v kategorii, ve které se uživatel právě nachází. Pod tímto

doporučením je uživateli umožněn výběr mezi zobrazeními nejlépe hodnocených, nejprodávanějších, nejlevnějších nebo nejdražších položek, které si může filtrovat podle rozsahu cen.



Obrázek 4: Doporučení položek na Alza.cz

Další doporučení je v tomto obchodu užito u položek samotných, kde systém doporučí příslušenství k prohlížené položce.

Obrázek 5: Doporučení příslušenství na Alza.cz

4.2 Typy doporučovacích systémů

4.2.1 Doporučení založené na vyhledávání

Hlavní výhodou tohoto typu je jednoduchost na implementaci. Zákazník zadává vyhledávací dotaz a systém vyhledá všechny položky, které odpovídají tomuto dotazu. Například uživatel zadá dotaz o zobrazení 6. nejoblíbenější knihy. Systém doporučí některou z těchto knih na základě všeobecného, neosobního hodnocení (podle prodejní pozice, popularity, atd.).

4.2.2 Doporučení založené na kategoriích

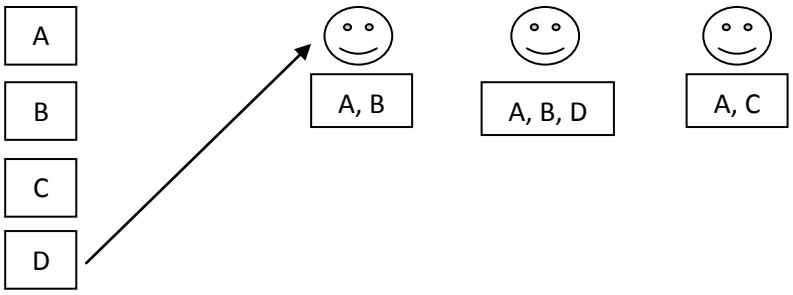
Základem je, že každá položka patří do jedné nebo více kategorií. Zákazník si vyberte kategorii zájmu (zpřesní vyhledávání). Systém vybere kategorie zájmů pro zákazníka na základě prohlížené aktuální položky, předchozích nákupů atd.

4.2.3 Kolaborativní doporučení (Collaborative filtering)

Základní myšlenkou tohoto systému je, že když uživatelé sdíleli zájem v minulosti – když si prohlíželi nebo koupili stejnou knihu – budou mít podobný vkus i v budoucnosti. Například když uživatel A a uživatel B mají historie nákupů, které jsou si silně podobné a uživatel A si koupí knihu, kterou B ještě neviděl, logickým výstupem bude nabídnout tuto položku i uživateli B. Kolaborativní přístup nevyžaduje žádnou znalost o položkách jako takových. Například o čem kniha je nebo kdo je autorem. Jasnou výhodou tohoto systému je, že tato data o položkách nemusí být vkládána do systému nebo být v něm udržována, proto není potřeba žádná údržba [11].

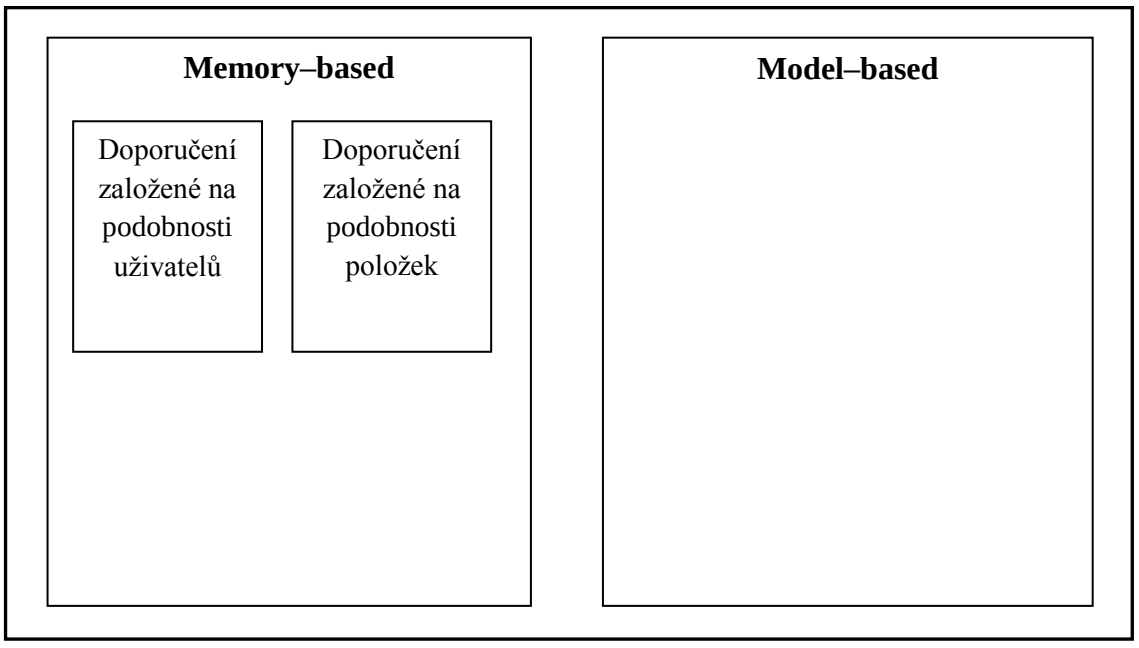
Konkrétní příklad je možné vidět na následujícím obrázku, kde aktivní uživatel má historii nákupů tvořenou položkami A, B. Dalším krokem je, že systém najde uživatele s podobnou historií. V tomto případě je blíže uživatel, který v minulosti koupil

položky A, B, D, protože oba koupili položky A, B. Jediná rozdílná položka je D, kterou si aktivní uživatel ještě nekoupil, tímto systém předpokládá, že by se mu tato položka mohla líbit, tak mu ji doporučí.



Obrázek 6: Příklad Kolaborativního doporučení

Metody kolaborativniho filtrování se mohou rozdělit do několika skupin. Přehled skupin lze vidět na následujícím obrázku.



Obrázek 7: Rozdělení metod Kolaborativního doporučení, upraveno a přeloženo z [12]

4.2.3.1 Memory-based Collaborative Filtering

Memory-based metody jsou historicky první metody kolaborativního filtrování. Tyto metody předvídají možné vztahy, které jsou počítány na základě známých vztahů mezi objekty. Agregační funkcí může být prostý průměr nebo některé další sofistikované opatření využívající rozdíly průměrných hodnocení nebo individuální podobnosti [12]. Jinými slovy využívají statistických metod k nalezení skupiny uživatelů známých jako sousedé, kteří mají podobnou historii cílového uživatele (tj., že buď je cena různých položek podobná, nebo mají tendenci koupit podobný soubor položek). Jakmile je sousedství uživatele vytvořeno, tyto systémy používají různé algoritmy ke kombinování preferencí sousedů k produkci doporučení pro aktivní uživatele [13].

Memory-based metody se rozdělují do dvou skupin, podle směru, který má být použit pro doporučení: Doporučení založené na podobnosti uživatelů (User-based) nebo Doporučení založené na podobnosti položek (Item-based).

4.2.3.1.1 Doporučení založené na podobnosti uživatelů (User-based)

Technika doporučení založené na podobnosti uživatelů je založena na vybrání podmnožiny uživatelů na základě podobnosti s aktivním uživatelem. Poté se podle hodnocení této podmnožiny vypočítají doporučení pro aktivního uživatele. Postup může být shrnut do následujících kroků [3]:

- Přiřazení váhy všem uživatelům pokud se podobají aktivnímu uživateli.
- Vybrání *k* uživatelů s nejvyšší podobností – vybrání nejbližších sousedů.
- Vypočítání předpovědi.

Prvním krokem je přiřazení váhy podobnosti ostatním uživatelům podle aktivního uživatele. Podobnost mezi dvěma uživateli se počítá pomocí Pearsonova korelačního koeficientu

$$sim(a,u) = \frac{\sum_{p \in P}\left(r_{a,p} - \overline{r_a}\right)\left(r_{u,p} - \overline{r_u}\right)}{\sqrt{\sum_{p \in P}\left(r_{a,p} - \overline{r_a}\right)^2}\sqrt{\sum_{p \in P}\left(r_{u,p} - \overline{r_u}\right)^2}},$$

kde *sim(a, u)* je váha podobnosti, $U = \{u_1, \ldots, u_n\}$ je množina uživatelů, kde *u* značí uživatele a *a* značí aktivního uživatele. $P = \{p_1, \ldots, p_n\}$ je množina položek ohodnocena oběma uživateli, $r_{u,p}$ je hodnocení položky *p* uživatelem *u* a r_u je průměrné hodnocení uživatele *u*.

Druhým krokem je výběr uživatelů, kteří mají největší podobnost s aktivním uživatelem.

Ve třetím kroku se provádí výpočet předpovědi z kombinace vybraných uživatelských hodnocení. Tato předpověď se obvykle počítá pomocí vzorce

$$pred(a,p) = \overline{r_a} + \frac{\sum_{u \in N} sim(a,u) * (r_{u,p} - \overline{r_u})}{\sum_{u \in N} sim(a,u)},$$

kde *pred(a, p)* je předpověď hodnocení aktivního uživatele *a* pro položku *p* a kde *N* je množina nejpodobnějších uživatelů – sousedství [11].

Ačkoli tento přístup byl úspěšně použit v různých doménách, některé problémy přetrvávají. Problém nastane, když se tato metoda aplikuje na velké komerční webové stránky, kde musíme zpracovat miliony uživatelů a miliony položek z katalogu. Díky nutnosti kontroly velkého počtu potencionálních sousedů není možné vypočítat předpovědi hodnocení v reálném čase. Tento problém řeší metoda založená na podobnosti položek [3].

4.2.3.1.2 Doporučení založené na podobnosti položek (Item-based)

Hlavní rozdíl algoritmů doporučení založené na podobnosti položek, oproti algoritmům doporučení založené na podobnosti uživatelů, je vypočítání předpovědi použitím podobnosti mezi položkami a ne podobnosti mezi uživateli [3]. Hlavní myšlenkou je, že pokud si uživatel koupil nějakou položku, předpokládáme, že by si mohl v budoucnu koupit položku podobnou. Analýzou historie nákupů uživatele můžeme tedy předpovědět, co si koupí v budoucnu [14].

Item-based algoritmy jsou dvoukrokové algoritmy, které se dají použít offline. V praxi to znamená rychlejší online systémy a často také kvalitnější doporučení.

Prvním krokem je zjištění podobnosti položek *a*, *b* pomocí Pearsonovi korelace vzorcem

$$sim(a,b) = \frac{\sum_{u \in U}\left(r_{u,a} - \overline{r_a}\right)\left(r_{u,b} - \overline{r_b}\right)}{\sqrt{\sum_{u \in U}(r_{u,a} - \overline{r_a})^2}\sqrt{\sum_{u \in U}(r_{u,b} - \overline{r_b})^2}},$$

kde *U* je množina všech uživatelů, kteří hodnotili obě položky *a* a *b*, $r_{u,a}$ je hodnocení uživatele *u* položky *a* a $r_{u,b}$ je hodnocení uživatele *u* položky *b* a kde r_a a r_b je průměrné hodnocení položek *a* a *b*.

Druhým krokem, poté, co jsou vypočítány podobnosti položek, je předpověď hodnocení položky *a* pro uživatele *u* pomocí váženého průměru

$$pred(u,a) = \frac{\sum_{b \in K} r_{u,b} * sim(a,b)}{\sum_{b \in K} |sim(a,b)|},$$

kde *K* je množina sousedních položek, hodnocených uživatelem *u*, které jsou nejvíce podobné položce *a* [14] [11].

Doporučení založené na podobnosti položek je používáno v internetovém obchodě Amazon, kde by se pro tak velké množství uživatelů (v současnosti přes 29 miliónů) doporučení na základě podobnosti uživatelů použít nedalo. Algoritmus v tomto obchodě nejdříve přiřadí uživatelovým nákupům a hodnocením podobné položky, a poté zkombinuje tyto podobné položky do listu doporučení. Aby autoři vylepšili škálovatelnost a výkonnost, rozdělili doporučování do dvou komponent. V režimu offline se vytvoří matice podobností položek, která je velmi náročná na zpracování. Potom se už v režimu online prohledává tato matice a produkují se doporučení [4].

4.2.3.2 Model-based Collaborative Filtering

Model-based metody poskytují doporučení položek tím, že nejprve vytvoří model uživatelských hodnocení [15]. Používají strojové techniky učení k naučení tohoto modelu k předpovídání neznámých vztahů. Známé vztahy se zde používají jako cvičná data. Mezi strojové techniky patří například Bayesovské sítě nebo neuronové sítě [12].

4.2.3.3 Nevýhody kolaborativnoho filtrovani

Metody kolaborativniho filtrování jsou nejpoužívanějšími a nejúspěšnějšími metodami, které se používají pro doporučení, ale mají i nevýhody.

Problém studeného startu (Cold start problem)

Při vytvoření nového uživatele nastává problém, jak tomuto uživateli doporučit nějakou položku, protože k doporučení je potřeba znát uživatelova hodnocení položek. Tento problém může být vyřešen požádáním uživatele k vložení dat. Tento postup se například využívá v systému internetové televize Netflix. Novému uživateli je zde podán dotazník k ohodnocení vybraných filmů. Další řešení tohoto problému je využití hybridních doporučovacích technik, které používají jednodušší doporučení, jako je nejoblíbenější položka atd. [12] [16].

Řídkost dat (Data sparsity)

V praxi je mnoho komerčních doporučovacích systémů používáno k vyhodnocení velmi velkých souborů produktů. Zákazníci většinou provádí hodnocení nebo si kupují výrobky jen ze zlomku položek v katalogu. Zde nastává problém, že metody k doporučení mohou mít extrémně řídká data, aby provedly nezpochybnitelná doporučení. Jednou možností odstranění tohoto problému, je využít dalších informací, které známe o uživateli (pohlaví, věk, vzdělání, zájmy a další dostupné informace). Soubor sousedů tohoto uživatele pak nebude založen jen na hodnocení položek, ale i na těchto informacích [17] [11].

Problém šedých ovcí (Grey sheep problem)

Šedé ovce jsou takoví uživatelé, kteří svými potřebami nezapadají ani do jedné skupiny ve srovnání se zbytkem komunity. Jinými slovy tito uživatelé mají nízké korelační koeficienty v porovnání s ostatními uživateli, protože s nimi částečně souhlasí nebo nesouhlasí. Přítomnost těchto uživatelů v malých a středních komunitách uživatelů představuje pro doporučení dvě hrozby. První hrozbou je, že tento uživatel neobdrží přesné doporučení. Druhá pak se týká toho, že mohou ovlivnit doporučení pro zbytek komunity [18]. Odstranění tohoto problému se provádí využití kombinací metod kolaborativního filtrování a metod doporučení založeného na obsahu [17].

Problém černých ovcí (Black sheep problem)

Černé ovce jsou takové skupiny uživatelů, pro které, z důvodu jejich potřeb, je doporučení téměř nemožné. Ačkoliv se jedná o selhání doporučovacího systému, jsou černé ovce přijatelným selháním [17].

Synonymie (Synonymy)

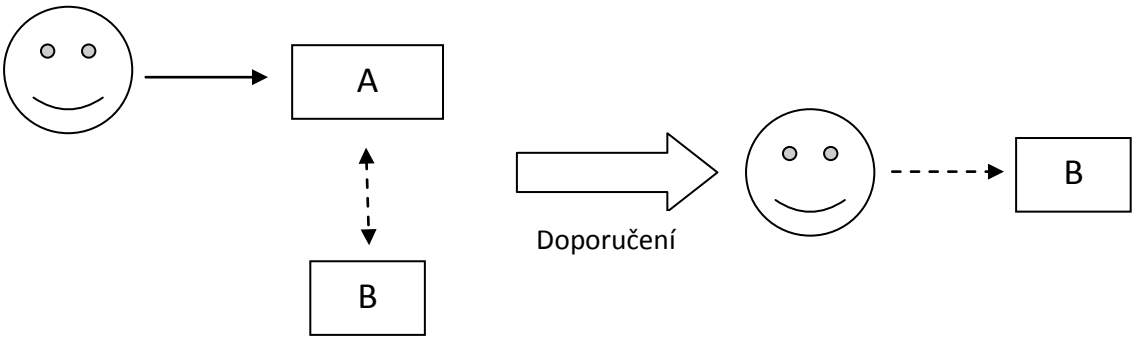
Synonymie je tendence řady stejných nebo velmi podobných položek mít různé názvy. Většina doporučovacích systémů je schopna odhalit toto latentní sdružení,

a tudíž zacházet s těmito produkty jinak. Výskyt těchto synonym snižuje výkon odporučovacích systémů [17].

4.2.4 Doporučení založené na obsahu (Content-based)

Metody doporučení založené na obsahu se snaží doporučit položky, které jsou podobné nebo souvisí s položkou, kterou si uživatel koupil nebo oblíbil v minulosti [11]. Hlavní výhodou tohoto přístupu je, že zde není potřeba provádění nákladných úkolů systémem, aby zajistil detailní a aktualizované popisy a hodnocení položek. Systému postačí základní informace o každé položce. Položky vybrané pro doporučení jsou pak položky, jejichž obsah nejvíce koreluje s preferencemi uživatele. Například, když uživatel ohodnotí položku, vytvoří se dotaz, který vyhledá další položky jako doporučení podle autora, interpreta nebo režiséra, nebo podle podobných klíčových slov [14].

Na obrázku 8 je zobrazen příklad, kde si uživatel v minulosti koupil položku A. K této položce je znám vztah s položkou B (zakreslený jako přerušovaná obousměrná šipka). Systém tento vztah vyhodnotí a doporučí uživateli položku B (zakresleno přerušovanou šipkou).



Obrázek 8: Příklad Doporučení založeném na obsahu, upraveno a přejato z [12]

I když tento přístup závisí na dalších informacích o položkách a uživatelích, nepotřebuje velkou uživatelskou základnu k tomu, aby vygeneroval doporučení. Seznam doporučení může být vytvořen, i když existuje jen jeden uživatel [4].

Systémy doporučení založené na obsahu typicky fungují pomocí hodnocení, jak silně je ještě nezobrazená položka podobná s položkami aktuálního uživatele, které si oblíbil v minulosti. Podobnost může být měřená různými způsoby. Například u nespatřené knihy B systém jednoduše zkontroluje, zda žánr této knihy je v seznamu preferovaných žánrů. Podobnost je v tomto případě 0 nebo 1. Další možností je vypočítat podobnost nebo překrytí pomocí klíčových slov. Tato možnost se nejčastěji používá pro více hodnotové charakteristiky. Zde se můžeme spolehnout na Diceův koeficient. Podobnost mezi knihami vypočítáme podle

$$\frac{2\left|K(b_i)\cap K\left(b_j\right)\right|}{|K(b_i)|+\left|K\left(b_j\right)\right|},$$

kde b_i a b_j jsou knihy a K označuje množinu klíčových slov [11].

Nevýhodou je, že automatické i manuální metody přiřazení charakteristik položkám nemusí být dostatečné k vygenerování vhodného doporučení pro uživatele. Dalším problémem je přílišná specializace doporučení. Nikdy tedy uživatel nedostane nějaké neočekávané doporučení, vždy budou doporučené položky velmi podobné těm, které hodnotil. Dalším problémem je, že nový uživatel musí ohodnotit dostatečné množství položek, aby mu mohlo být vygenerováno vhodné doporučení [12].

4.2.5 Doporučení založené na znalostech (Knowledge-based)

Tento přístup využívá znalostí o uživatelích a položkách k vygenerování doporučení. Většina obchodních odporučovacích systémů je v praxi založena

na kolaborativním doporučení nebo na doporučení založeném na obsahu. Oba přístupy mají své výhody, ale také existuje mnoho situací, pro které nejsou tou nejlepší volbou. Typickým příkladem je nákup věcí, které se nekupují často. Důvodem je, že například u bytů je nemožné získat dostatek hodnocení, protože není dostatek stejných exemplářů, nebo že uživatel by určitě nebyl spokojený s doporučením počítače podle rok starých hodnocení. Doporučení založené na znalostech pomáhá vypořádat se s těmito problémy [11] [12].

Výhodou tohoto přístupu je, že se nepotýká s žádným z problémů předchozích přístupů, protože nepotřebuje žádná hodnocení k vypočtení doporučení. Doporučení je vypočítáno individuálně pro každého uživatele a nezávisle na ostatních [11].

Naopak problém tohoto přístupu je, že doporučení nejsou osobní. Neexistují v systému žádné uživatelské profily a doporučení je založeno pouze na datech, které poskytl uživatel při vyhledávání doporučení [12].

Existují dva základní typy: doporučení na základě omezení (constrait-based) nebo na základě požadavků (case-based). Oba typy jsou podobné z hlediska procesu doporučení: uživatel musí specifikovat požadavky a poté se systém snaží určit řešení. Když nemůže být řešení nalezeno, uživatel musí změnit požadavky. Rozdílné jsou v tom, že doporučení na základě požadavků se soustřeďuje na nalezení nejvíce podobných položek na základě různých druhů podobnostních mír, zatímco v doporučení na základě omezení je sada doporučených položek například určena vyhledáváním v sadě položek, která splňují doporučovací pravidla [11].

4.2.5.1 Doporučení na základě omezení (Constraint-based)

Tento doporučovací systém je definován dvěma množinami proměnných (V_C, V_{PROD}), kde první popisuje zákazníkovy požadavky a druhá vlastnosti produktu, a třemi různými množinami omezení (C_R, C_F, C_{PROD}), které definují, jaké položky by měly být

zákazníkovi doporučeny v jaké situaci. Pro představu je zde použit příklad z prodeje digitálních kamer [11].

Zákazníkovy požadavky (V_C) – Popisuje možné požadavky zákazníka, například max-price je maximální cena kamery.

Vlastnosti produktu (V_{PROD}) – Popisuje vlastnosti produktu ve výběru, například Mpix označuje možné rozlišení digitální kamery.

Omezení (C_R) – Definuje povolené výběry požadavků uživatele, například kamera s rozlišením větším než 10Mpix nemůže mít nižší cenu než 3000 korun.

Filtrovací podmínky (C_F) – Definuje, za jakých podmínek by měl být výrobek vybrán – jinými slovy, filtrovací podmínky definují vztahy mezi zákazníkovými požadavky a vlastnostmi produktu. Například kamera s rozlišením alespoň 15Mpix může být vybrána, jen když je maximální cena větší než 10000 korun.

Omezení produktů (C_{PROD}) – Definuje, které produkty jsou v současnosti dostupné.

Když jsou všechny tyto proměnné vyplněny, doporučení je pak už jednoduché. Zákazník si například zadá, že chce digitální kameru s rozlišením minimálně 12Mpix a cena nesmí být větší než 6000 korun. Systém buď takový produkt najde, nebo napíše, z jakého důvodu nebyl žádný nalezen [11].

4.2.5.2 Doporučení na základě požadavků (Case-based)

V tomto přístupu jsou doporučené položky nalezeny použitím podobnostních mír, které popisují, jak velký rozsah vlastností produktu odpovídá požadavkům, které zákazník zadal. Položky jsou tedy doporučovány podle toho, jak moc se podobají požadavkům [11].

Takzvaná podobnostní vzdálenost položky *p* od požadavků *r* z množiny požadavku *REQ* je často definována jako

$$similarity(p, REQ) = \frac{\sum_{r \in REQ} w_r * sim(p, r)}{\sum_{r \in REQ} w_r},$$

kde *sim(p, r)* vyjadřuje, pro každou hodnotu atributu položky $\Phi_r(p)$, vzdálenost od zákazníkova požadavku *r* z množiny *REQ* a kde w_r je významová váha pro požadavek *r* [11].

V reálném světě jsou proměnné, které by zákazník chtěl maximalizovat. Například rozlišení digitální kamery. Jsou zde také proměnné, které chce zákazník minimalizovat – například cenu digitální kamery. V prvním případě mluvíme o tzv. „more-is-better“ (MIB) proměnných, ve druhém případě jsou pak odpovídající proměnné označovány jako „less-is-better“ (LIB) [11].

Tyto požadavky jsou pak vypočteny podle následujících vzorců:

V případě MIB proměnných je podobnost mezi atributem položky *p* a požadavky uživatele *r* vypočítána takto:

$$sim(p, r) = \frac{\Phi_{\mathrm{r}}(\mathrm{p}) - \min(\mathrm{r})}{\max(r) - \min(r)}$$

Podobnost mezi atributem položky *p* a požadavky uživatele *r* pro případ, že uživatel chce danou položku minimalizovat, je vypočítána takto:

$$sim(p, r) = \frac{\max(r) - \Phi_{\mathrm{r}}(\mathrm{p})}{\max(r) - \min(r)}$$

Jsou zde také situace, ve kterých je podobnost založena výhradně na vzdálenosti původně definovaných požadavků. Například, když chce uživatel určitou velikost monitoru. Pro tyto případy musí být zavedeno třetí typ podobnostní funkce:

$$sim(p,r) = 1 - \frac{|\Phi_r(\mathrm{p}) - \mathrm{r}|}{\max(r) - \min(r)}$$

Pro vytvoření doporučení uživatel zadá požadavky a systém doporučí produkty, které se nejblíže přiblížily uživatelovým požadavkům. Když není uživatel spokojen, musí změnit požadavky a doporučení se vyhledává znovu [11].

4.2.6 Hybridní doporučovací techniky

Předchozí doporučovací techniky mají své silné ale i slabé stránky. Samostatné techniky nedokážou plně využít potenciál doporučovacích vstupů (uživatelské modely, data produktu nebo znalostní modely). Hybridní doporučovací systémy kombinují dvě, nebo více doporučovacích technik. Tím se kombinuje síla jednotlivých technik a snižují se jejich slabiny [11] [19].

Většinou se jedná o zkombinování kolaborativního filtrování s technikou doporučení založeného na obsahu. Různé druhy jak kombinovat tyto techniky do hybridního doporučovacího systému mohou být klasifikovány následovně [16]:

1. Provádění kolaborativních metod a metod založených na obsahu samostatně a kombinování předpovědí.

2. Včleňování některých charakteristik založených na obsahu do kolaborativního přístupu.

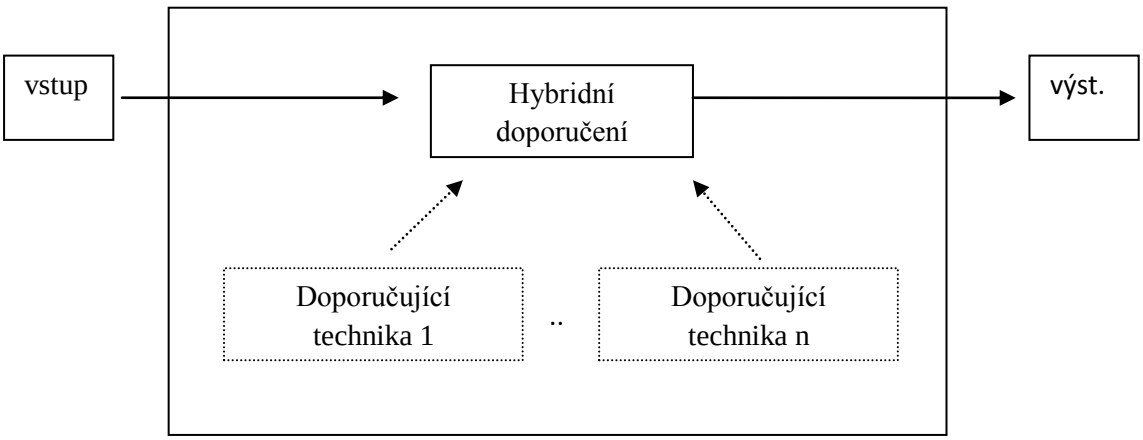
3. Včleňování některých kolaborativnich charakteristik do přístupu založeného na obsahu.

4. Konstruování obecného jednotného modelu, který zahrnuje jak charakteristiky kolaborativnich přístupů, tak charakteristiky přístupů založených na obsahu.

Návrhy hybridizace:

4.2.6.1 Monolitický návrh hybridizace

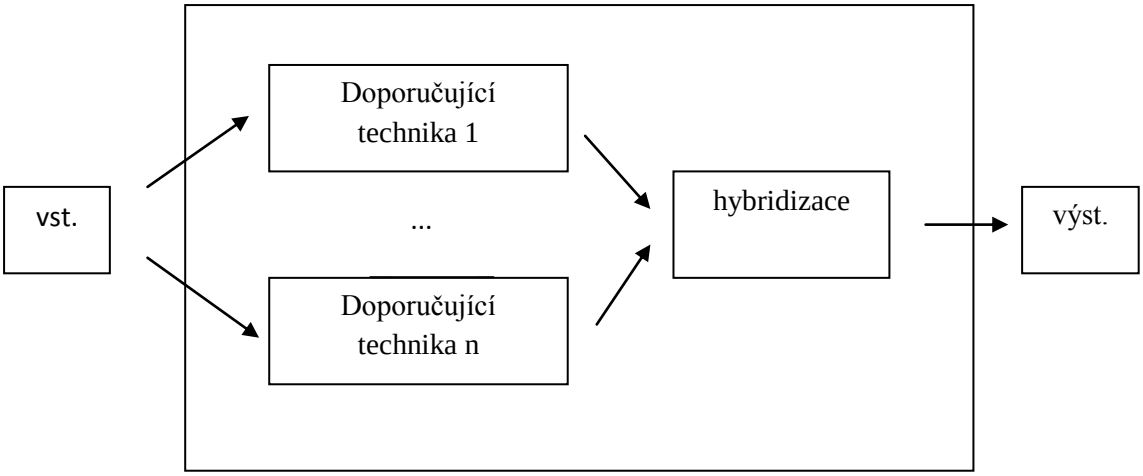
Následující dva návrhy hybridních doporučovacích systémů se skládají ze dvou nebo více komponent, jejichž výsledky jsou zkombinovány. Monolitický návrh hybridizace se skládá z jediné doporučovací komponenty, která spojuje více přístupů předzpracováním a kombinováním několika znalostních zdrojů. Hybridizace je tak dosaženo pomocí zabudovaných změn v chování algoritmu využít různých typů vstupních dat. Typicky kroky pro předzpracování specifických dat jsou použity k transformaci vstupních dat do zastoupení, které může být využito ke konkrétnímu paradigmatu algoritmu [11].



Obrázek 9: Monolitycký návrh hybridizace, přeloženo a upraveno z [11]

4.2.6.2 Paralelní návrh hybridizace

Tento návrh využívá několik technik doporučení pracujících bok po boku se specifickými hybridizačními mechanismy, které shrnují jejich výstup. Existují tři základní strategie a to Mixed hybrids (smíšené hybridy), Weighted hybrids (vážené hybridy) a Switched hybrids (přepínací hybridy) [11].



Obrázek 10: Paralelní návrh hybridizace, přeloženo a upraveno z [11]

Mixed hybrids (Smíšené hybridy)

Tato strategie kombinuje výsledky rozdílných doporučovacích systémů na úrovni uživatelského rozhraní, ve kterém jsou pak výsledky doporučení z různých technik prezentovány společně. Proto výsledek doporučení pro uživatele *u* a položku *i* je sada n-tic ⟨score,k⟩, kde score je doporučovací funkce rec_k [11]:

$$rec_{mixed}(u,i) = \bigcup_{k=1}^{n} \langle rec_k(u,i), k \rangle$$

Když použijeme najednou kolaborativní doporučení a doporučení podle obsahu, vyhneme se problému nové položky, protože se s ním vypořádá druhá technika.

Ale s problémem nového uživatele se ani toto spojení nevypořádá, protože oba přístupy potřebují nějaké informace o uživateli [4].

Weighted hybrids (Vážené hybridy)

Strategie weighted hybrids kombinuje doporučení dvou nebo více odporučovacích systémů vypočtením váženého součtu jejich hodnocení. Vezměme si tedy *n* rozdílných odporučovacích funkcí rec_k s relativními váhami β_k [11]:

$$rec_{weighted}(u,i) = \sum_{k=1}^{n} \beta_k * rec_k(u,i)$$

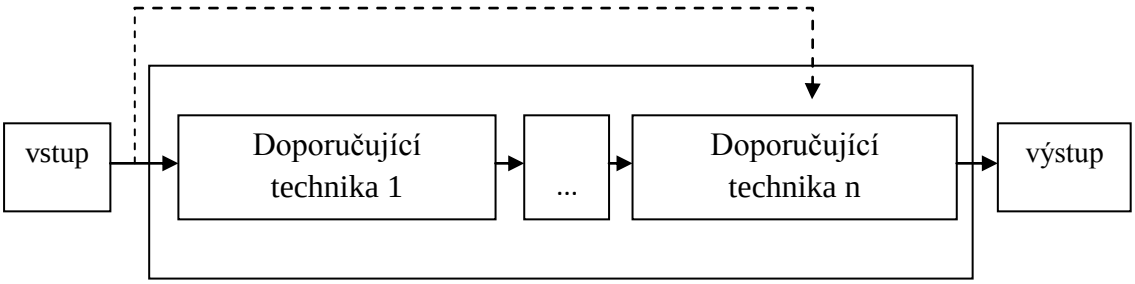
Výhodou je, že všechny systémové kapacity mohou být využity pro proces doporučení přímočaře. Nicméně, předpoklad této strategie je to, že relativní hodnota různých technik je víceméně jednotná napříč možnými položkami. Není tomu bohužel tak vždy. Například spolupráce bude slabší u těch položek, které mají malý počet hodnotitelů [19].

Switched hybrids (Přepínací hybridy)

Switched hybrids zavádějí další stupeň detailu doporučovacího procesu, protože musí být stanovena přepínací kriteria, čímž se zavádí další úroveň parametrizace. Přepínací kritéria jsou použita pro přepínání doporučovacích technik. Například, když systém používá pro doporučení hybridní techniku založenou na kolaborativním doporučením a doporučením založeném na obsahu, postupuje tak, že je aplikována první technika doporučení založeném na obsahu. V případě, že tato technika nemůže podat doporučení s dostatečnou jistotou, tak je aplikována druhá technika. Výhodou je, že systém může být citlivý na silné i slabé stránky svých doporučovacích technik [19].

4.2.6.3 Trubicový návrh hybridizace

Podstatou návrhu je fázový proces, ve kterém je prováděno několik technik postupně za sebou, a jsou ukončeny před finální technikou, která vytváří doporučení pro uživatele. Varianty hybridizace odlišují sami sebe hlavně podle výstupu, který produkují pro další fázi. Jinými slovy předchozí komponenty mohou předběžně zpracovat vstupní data k vytvoření modelu, který je využíván v další fázi [11].



Obrázek 11: Trubicový návrh hybridizace, přeloženo a upraveno z [11]

Cascade hybrids (Kaskadové hybridy)

Cascade hybryds jsou založeny na pořadí, kde jsou jednotlivé techniky řazeny za sebou, a kde každé následné doporučení zpřesňuje doporučení jejich předchůdce. Seznam položek k doporučení pro následující techniku je omezený položkami, které byly doporučeny předchozí technikou. Jinými slovy na začátku první technika nejprve vyprodukuje hrubé kandidáty na doporučení. Následující techniky pak jen upřesňují pořadí a počet v této sadě kandidátů [11] [19].

Předpokládejme sled technik *n*, kde rec_1 představuje doporučovací funkci první techniky a rec_n poslední. V důsledku toho je konečné doporučovací skóre položky počítáno *n*-tou technikou. Nicméně položka je navržena *k*-tou technikou pouze v případě, že *(k - 1)* technika také přiřadila položce nenulové skóre. To platí pro všechna $k \geq 2$ jak je definováno následovně [11]:

$$rec_{cascade}(u, i) = rec_n(u, i),$$

kde $\forall k \geq 2$ musí platit:

$$rec_k(u,i) = \begin{cases} rec_k(u,i) & : \quad rec_{k-1}(u,i) \neq 0 \\ 0 & : \quad else \end{cases}.$$

V Cascade hybrids tedy všechny techniky, krom první, mohou pouze změnit pořadí v seznamu doporučených položek od jejich předchůdce nebo vyřadit položku změnou jejího skóre na 0 [11].

Meta-level hybrids (Meta-úrovňové hybridy)

Ve strategii meta-úrovňové hybridizace jedna doporučující technika staví model, který je využíván hlavní doporučující technikou k vytvoření doporučení. První meta-úrovňový hybrid byl systém pro filtrování webu nazvaný Fab. Fab využíval kolaborativní přístup, který staví na uživatelských modelech, které využívaly doporučení založené na obsahu [11] [19].

Vzorec formalizuje toto chování, kde n-tá doporučovací technika využívá modelu Δ, který byl vytvořen předchůdcem.

$$rec_{meta-level}(u,i) = rec_n(u,i,\Delta_{rec_{n-1}})$$

Výhodou této strategie, zejména pro hybridy kombinující doporučení založené na obsahu a kolaborativní doporučení, je, že naučený model je komprimovaná reprezentace uživatelských zájmů a pro kolaborativní mechanismus, který poté následuje, může pracovat v informačně husté reprezentaci mnohem snadněji než v syrových datech [19].

5 Algoritmy využívané při procesu doporučení pro výpočet podobnosti

5.1 Algoritmus K-NN (K nearest neighbours)

Klasifikace podle nejbližších sousedů spadá mezi neparametrické metody klasifikace. Tyto metody jsou založeny na podstatně slabších předpokladech než metody parametrické. Nepředpokládáme zde znalost tvaru pravděpodobnostních charakteristik tříd. Nejčastější je klasifikace podle jednoho souseda (1-NN), ale existují i klasifikace pro obecně *k* sousedů [20].

Postup algoritmu je takový, že pro uživatele u_0 je vybráno K nejbližších sousedů $u_1, \ldots, u_K$, přičemž vzdálenost uživatelů se pro tento algoritmus určuje podle vzorce

$$d(u_a, u_b) = \sqrt{\sum_{i=1}^{n} [p_a(o_i) - p_b(o_i)]^2},$$

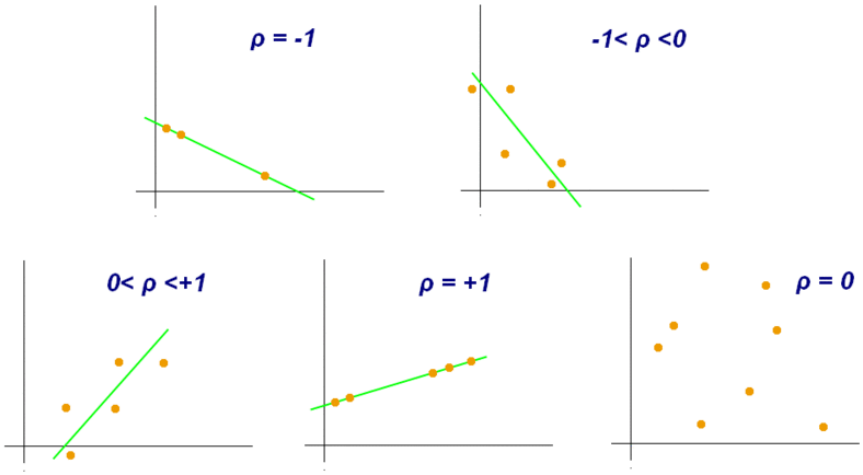
kde d je vzdálenost (distance) uživatelů u_a a u_b, n je počet porovnávaných objektů o_i a $P(o)$ je hodnocení daného objektu. Je-li tedy zvoleno K nejbližších sousedů, lze hodnocení porovnávaných objektů pro uživatele u_0 vypočítat jako aritmetický průměr hodnocení vybraných uživatelů [6]:

$$P_0(o) = \frac{\sum_{i=1}^{n} P_i(o)}{K}.$$

5.2 Pearsonův korelační koeficient

Pearsonův korelační koeficient je míra korelace (lineární závislosti) mezi dvěma proměnnými X a Y. Je široce používán ve vědách jako měřítko síly lineární závislosti

mezi dvěma proměnnými. Vyvinul jej Karl Pearson ze související myšlenky, kterou představil Francis Galton v roce 1880. Korelační koeficient se pohybuje mezi hodnotami 1 a -1 včetně. Hodnota 1 znamená, že lineární rovnice popisuje vztah mezi X a Y dokonale, všechny body leží na přímce a přímka je stoupající. Hodnota -1 znamená, že všechny datové body leží na přímce, pro kterou platí, že je klesající. Hodnota 0 znamená, že neexistuje žádný lineární vztah mezi proměnnými [21].



Obrázek 12: Příklady diagramů s různými hodnotami korelačního koeficientu [21]

Těchto mezních hodnot ovšem Pearsonův korelační koeficient nabývá velmi zřídka. Hodnotu tohoto koeficientu lze určit jako podíl kovariance sledovaných proměnných a jejich směrodatných odchylek. Kovariance je definována jako střední hodnota součinu rozdílu sledovaných proměnných od jejich středních hodnot [6] [21].

Vzorec pro výpočet je tedy následující

$$r = \frac{cov(X,\ Y)}{s_x s_y} = \frac{E\left[\left(X_i - E(X)\right)\left(Y_i - E(Y)\right)\right]}{s_x s_y} = \frac{1}{n-1}\sum_{i-1}^{n}\left(\frac{X_i - \bar{X}}{s_x}\right)\left(\frac{Y_i - \bar{Y}}{s_y}\right),$$

kde n je tzv. stupeň volnosti, tedy počet hodnot použitých k výpočtu (pro výpočet odchylky se ve jmenovateli vždy počítá s hodnotou $n - 1$), X_i a Y_i jsou konkrétní hodnoty, $\bar{X}$ a $\bar{Y}$ jsou průměrné hodnoty a s_x a s_y jsou směrodatné odchylky [6].

5.3 Spearmanův korelační koeficient

Jde o neparametrickou metodu, která při výpočtu využívá pořadí hodnot sledovaných veličin, nevyžaduje tedy normalitu dat. Výhodou je, že lze tuto metodu použít pro popis jakékoliv závislosti - lineární i nelineární. Spearmanův korelační koeficient používáme nejčastěji pro měření síly vztahu u takových veličin, u kterých nemůžeme předpokládat linearitu očekávaného vztahu nebo normální rozdělení sledovaných proměnných X a Y. Pro malé rozsahy n je výpočet Spearmanova korelačního koeficientu méně pracný než výpočet Pearsonova parametrického korelačního koeficientu. Proto je možno ho použít i k hodnocení lineárních závislostí, kde je jeho použití spíše orientační (využívá méně informací z dat) a na rozdíl od parametrického koeficientu je méně účinný [22].

Výpočet Spearmanova korelačního koeficientu vychází z pořadových čísel proměnných x_i a y_i (korelačních dvojic) naměřených u n jedinců výběrového souboru. Jsou-li hodnoty proměnných x_i a y_i seřazeny vzestupně do dvou řad a každé hodnotě je přiděleno pořadí, pak koeficient pořadové korelace je dán vztahem

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)},$$

kde d_i je rozdíl mezi pořadím hodnot x_i a y_i příslušných korelačních dvojic a n je počet korelačních dvojic [6] [22].

6 Návrh modulu pro doporučovací systém

V dnešní době existuje velké množství internetových obchodů, které nabízejí svým zákazníkům velké množství výrobků. Zákazník si výrobek vybere, objedná a potvrdí objednávku. Osoba na straně internetového obchodu objednávku zpracuje, zajistí požadované množství položek a nakonec odešle zákazníkovi. Z každého tohoto

procesu systém obchodu získává informace o chování zákazníka, které jsou použity a zpracovány k vytvoření doporučení. Ale problém nastává při registraci nového uživatele, kterému systém nemůže doporučit a nedoporučí žádný výrobek. Tento problém se dá řešit různými několika možnostmi. Jako přiklad řešení tohoto problému bylo zvoleno modifikované doporučení založené na obsahu. Výhodou je, že systém nemusí vědět o aktivitách zákazníka. Stačí mu pouze vlastnosti a atributy zákazníka a vlastnosti položek.

6.1 Analýza aplikace

Aplikace ukazuje implementaci jednoduchého doporučovacího systému. Je to tedy webová aplikace, která simuluje e-shop a skládá se ze šesti tříd a to Item.java, User.java, Db.java, Core.java, ShopBean.java a RegBean.java.

Item.java – zastupuje zákazníka a jeho vlastnosti

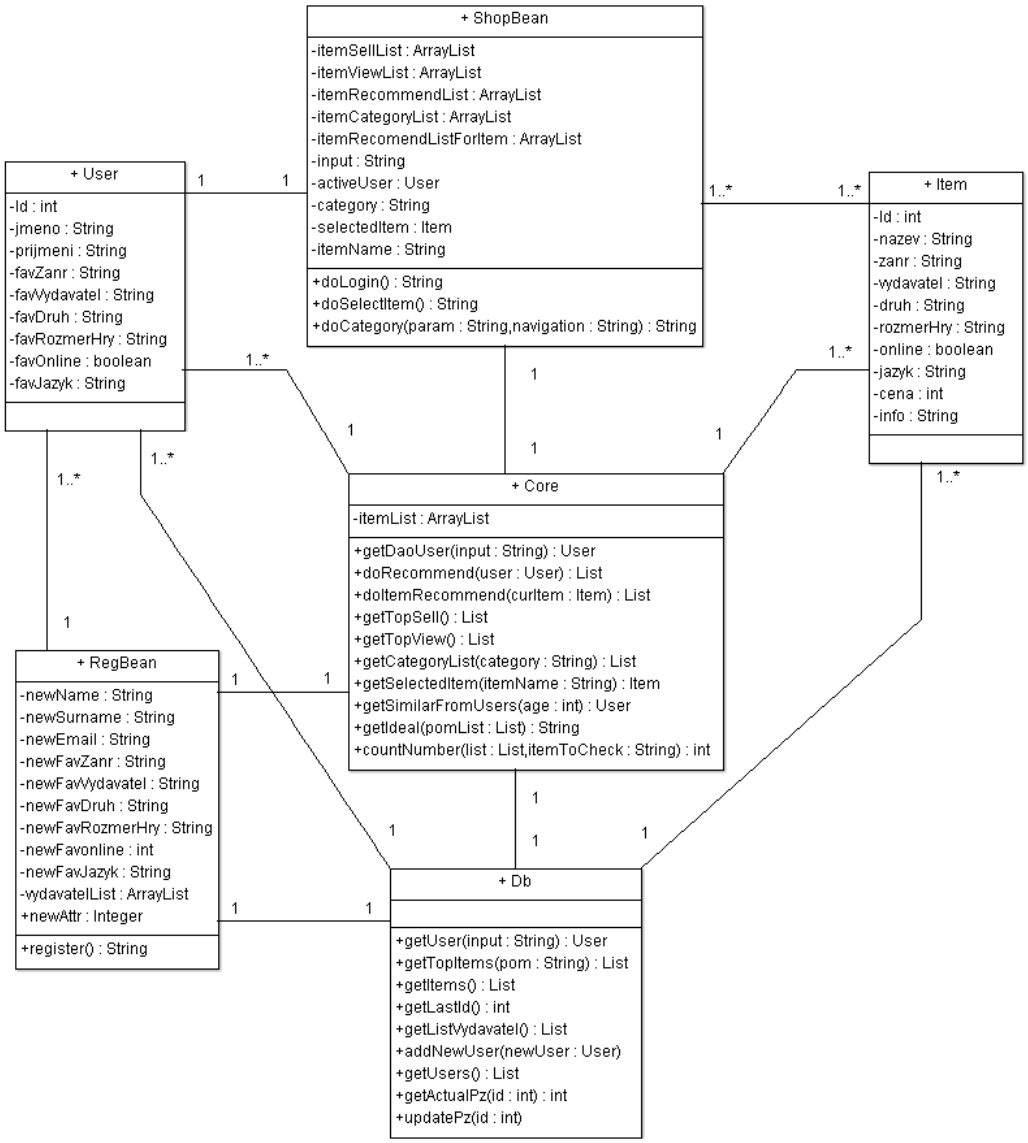
User.java – zastupuje položku a její vlastnosti

Db.java – třída, která se stará o komunikaci mezi aplikací a databází

Core.java – třída, která zpracovává data a vypočítává doporučení a tvoří virtuálního uživatele pro případné doplnění chybějících hodnot

ShopBean.java – java bean, který slouží k načítání a ukládání dat v obchodě

RegBean.java – java bean, který slouží k načítání a ukládání dat v registraci nového uživatele, dále se zde doplňují části, které uživatel nezadal



Obrázek 13: Diagram tříd

Uživateli se doporučí nejprodávanější výrobky a nejnavštěvovanější výrobky všemi zákazníky, dále dva výrobky, které by uživatele mohli zajímat, podle jeho zájmů a dva nejbližší k aktuálně prohlíženému výrobku.

6.2 Technologie použité při vývoji

6.2.1 Platforma Java EE

Java Platform, Enterprise Edition (Java EE) 6 je průmyslový standard pro podnikové nasazení jazyka Java. Tento standart se v současné době používá pro vývoj podnikových řešení pomocí webových služeb. Softwarový systém Java Platform je vývojové a hostující prostředí založené na programovacím jazyku Java. Jedná se o standardizovanou platformu, která je podporována ze strany mnoha dodavatelů, jež nabízejí vývojové nástroje, prvky běžící na straně serveru a middleware pro tvorbu a nasazení řešení na bázi jazyku Java [23].

Platforma Java Platform je rozdělena na tři hlavní vývojové a běhové platformy, z nichž každá je zaměřena na odlišný typ řešení [23].

- Platforma Java SE je navržena pro podporu tvorby desktopových aplikací.

- Java ME se soustředí na aplikaci běžící na mobilních zařízeních.

- Java EE je vybudovaná pro podporu rozsáhlých, distribuovaných řešení. Je využívána pro tvorbu tradičních vícevrstevných aplikací, ať už s nebo bez webových aplikací.

Dnes jsou různými dodavateli nabízené rozdílné vývojové produkty, jež poskytují určité prostředí, v rámci něhož lze pro budování webových služeb používat standardní jazyk Java. Mezi nejznámější patří například NetBeans IDE, JDeveloper nebo Eclipse [23].

Pro tuto aplikaci byla použita technologie Java Server Faces.

Netbeans IDE 7.1.2

NetBeans je Open Source projekt s velmi rozsáhlou uživatelskou základnou, rostoucí komunitou vývojářů a téměř 100 partnery po celém světě. Firma Sun Microsystems založila Open Source projekt NetBeans v červnu 2000 a je zároveň hlavním sponzorem celého projektu [24].

Dnes existují dva produkty: vývojové prostředí NetBeans (NetBeans IDE) a vývojová platforma NetBeans (The NetBeans Platform) [24].

Vývojové prostředí NetBeans IDE je nástroj, pomocí kterého programátoři mohou psát, překládat, ladit a distribuovat aplikace. Samotné vývojové prostředí je vytvářeno v jazyce Java - ovšem podporuje prakticky jakýkoliv programovací jazyk. Existuje rovněž velké množství modulů, které toto vývojové prostředí rozšiřují. Vývojové prostředí NetBeans je bezplatně šířený produkt a jeho užívání není nijak omezeno [24].

Kromě vývojového prostředí je také dostupná vývojová platforma NetBeans Platform, což je modulární a rozšiřitelný základ pro vytváření rozsáhlých desktopových aplikací. Nezávislí dodavatelé softwaru nabízejí dodatečné moduly, které lze snadno integrovat a které mohou být použity k vývoji jejich vlastních nástrojů a řešení [24].

Oba produkty jsou vyvíjeny pod licencí Open Source a je možné je bezplatné používat v komerčním i nekomerčním prostředí. Zdrojový kód je dostupný pod licencí Common Development and Distribution License (CDDL) v1.0 a GNU General Public License (GPL) v2 [24].

6.2.2 Databázový systém MySQL

MySQL byla vytvořena v roce 1995 jako jednoúčelová databáze pro snadné ukládání a především čtení textových dat v internetových aplikacích. Za MySQL stojí

švédská společnost MySQL AB. Její zakladatelé jsou dva Švédové a jeden Fin: David Axman, Allan Larsen a Michael „Monty“ Widenius. MySQL AB vlastní práva na zdrojový kód, obchodní značku a doménu mysql.com. Společnost má kolem 30 zaměstnanců roztroušených po celém světě – především z řad testerů a systémových integrátorů [25].

Hlavní myšlenkou této firmy bylo vytvoření takového databázového systému, který by byl dostupný všem. Jelikož cena MySQL při dodržení GNU General Public Licence (GPL) je nulová, tak nezbývá než konstatovat, že se jim to také povedlo. MySQL se stal nejpopulárnějším databázovým systémem a mezi spokojené zákazníky patří například Yahoo! Finance, MP3.com, Motorola, NASA, Silicon Graphics nebo Texas Instruments [25].

Pro administraci tohoto serveru byl použit software napsaný v PHP PhpMyAdmin. MySQL databáze i PhpMyAdmin s Apache jsou obsazeny v instalačním balíku XAMPP.

6.2.3 Aplikační server

Aplikační server tvoří vrstvu mezi operačním systémem a aplikacemi. Podobně, jako operační systém poskytuje základní funkce programům, poskytuje aplikační server často používané funkce enterprise aplikacím. Vytváří další vrstvu abstrakce, aby bylo psaní aplikací jednodušší. Příkladem takových funkcí mohou být podpora transakčního zpracování požadavků, persistence objektů do databáze, výměna zpráv mezi aplikacemi a další [23].

Pro tuto práci byl použit aplikační server GlassFish, který je možno stáhnout již s instalací NetBeans. GlassFish je aplikační server vyvinutý společností Sun Microsystems pro platformu Java EE. GlassFish se řadí mezi open source podléhající licencím GPL a CDDL. GlassFish je referenční implementace, to znamená, že není primárně určen pro provoz aplikací, ale slouží především jako ukázka implementace

nových rysů v poslední specifikaci platformy JAVA EE. Současná verze serveru GlassFish je 3.1.2 a slouží jako referenční implementace pro Javu EE6.

6.3 Funkce navrženého modulu

Obrázek 14: Doporučení na hlavní stránce obchodu

6.3.1 Nejprodávanější výrobky

Tato možnost doporučení je vidět ve většině internetových obchodů. Aplikace zobrazuje 5 nejvíce prodávaných výrobků. Tento seznam výrobků je výsledkem SQL dotazu na databázi obchodu, konkrétně na tabulku Item, která obsahuje všechny výrobky a jejich vlastnosti, které jsou prodávány v obchodě. Seznam je načítán ve třídě Db.java pomocí metody getTopItems("pk"), kde parametr pk označuje název sloupce v databázi, na který se chceme dotazovat. Metoda SQL dotazem "SELECT * from item ORDER BY pk DESC LIMIT 5" vygeneruje setříděný seznam, a protože seznam již nepotřebuje žádné úpravy, třída Core.java jej jen předá dál třídě ShopBean.java. Další možností je tento seznam vytvořit z databáze prodaných výrobků.

Id	Nazev	...	pk	...
0	Torchlight 2	…	38	…
1	Borderlands	…	58	…
2	Far Cry 3	…	39	…
3	Settlers 3	…	20	…
…	…	…	…	…

Tabulka 1: Tabulka item s počtem koupení

6.3.2 Nejnavštěvovanější výrobky

Nejnavštěvovanější výrobky je další doporučení, které může uživatele zlákat k navštívení a zakoupení daného populárního produktu. Aplikace funguje jako v předchozím případě, kde seznam nejnavštěvovanějších výrobků je výsledkem SQL dotazu na databázovou tabulku Item. Načítá se ve třídě Db.java pomocí metody getTopItems("pz"), kde parametr pz označuje název sloupce v databázi, kde je zaznamenán počet zobrazení výrobku. Metoda, jako v případě nejprodávanějších výrobků, vygeneruje setříděný seznam pomocí SQL dotazu "SELECT * from item ORDER BY pz DESC LIMIT 5". Seznam je také setříděný, proto je poslán přes třídu Core.java rovnou do třídy ShopBean.java.

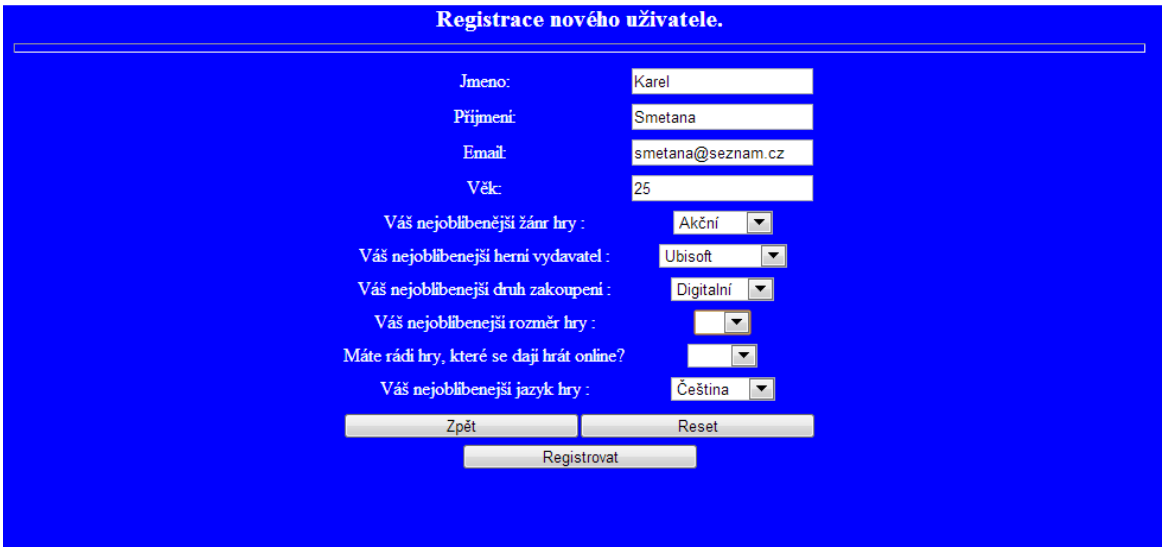
Id	Nazev	...	pz
0	Torchlight 2	…	55
1	Borderlands	…	65
2	Far Cry 3	…	70
3	Settlers 3	…	30
…	…	…	…

Tabulka 2: Tabulka item s počtem zobrazení

6.3.3 Doporučení položek, které jsou pro zákazníka zajímavé

Nezbytným bodem pro toto doporučení je znát informace o uživateli. Konkrétně jeho zájmy v oboru daného obchodu. Na tyto informace se uživatele systém zeptá při

registraci. O registraci se stará třída RegBean.java, která zachytává informace zadané uživatelem, poté vytvoří tohoto nového uživatele a pošle jej třídě Db.java pomocí metody register(), kde je tento nový uživatel zapsán do databáze pomocí metody addNewUser(User newUser).



Obrázek 15: Registrace nového uživatele

Samotné doporučení položek probíhá po přihlášení uživatele do obchodu zavoláním metody doRecommend(User user) z třídy Core.java, kde parametr user je aktuálně přihlášený uživatel. Metoda nejprve zjistí, zda je načtený seznam všech položek obchodu, pokud ne, tak zavolá metodu getItems() z třídy Db.java, která vrací tento seznam. Poté jsou všechny uživatelovi ideální vlastnosti položky porovnány s atributy položek ze seznamu. Když se atribut položky shoduje s ideálem, k výslednému skóre je přičtena určitá hodnota. Když je skóre právě porovnané položky ze seznamu větší, než doposud maximální dosažené skoré jiné položky, tak je doposud maximální položka přesunuta na druhé místo a aktuální položka na první. Při situaci, kdy aktuální položka bude mít menší skóre než maximální, ale větší než má položka na druhém místě, je aktuální zapsána na druhé místo.

```
public List doRecommend(User user)
    {
        List<Item> recommendList = new ArrayList<Item>();
        if(itemlist.isEmpty())
```

```
        {
            try {
                itemlist = dao.getItems();
            } catch (Exception ex) {

Logger.getLogger(Core.class.getName()).log(Level.SEVERE, null, ex);
            }
        }


        max = 0;
        max2 = 0;

        for (Item item : itemlist)
        {
            int count = 0;
            if(user.getFavZanr().equals(item.getZanr()))
            {
                count = count+1;
            }

            if(user.getFavVydavatel().equals(item.getVydavatel()))
            {
                count = count +1;
            }

            if(user.getFavDruh().equals(item.getDruh()))
            {
                count = count +1;
            }

            if(user.getFavRozmerHry().equals(item.getRozmerHry()))
            {
                count = count +1;
            }

            if(user.isFavOnline() && item.isOnline())
            {
                count = count +1;
            }

            if (user.getFavJazyk().equals(item.getJazyk()))
            {
                count = count +1;
            }

            if(count > max)
            {
                max2 = max;
                max = count;
                recommend2 = recommend1;
                recommend1 = item;
            }
            else if(count > max2)
            {
                max2 = count;
                recommend2 = item;
            }
```

```
        }
        recommendList.add(recommend1);
        recommendList.add(recommend2);

        return recommendList;

    }
```

Algoritmus 1: Doporučení podle uživatelova ideálu

Výsledek doporučení jsou dvě položky, které jsou svými vlastnostmi nejvíce přiblížené ideálu uživatele. Připočtená hodnota ke skóre, pokud je vlastnost ideálu a vlastnost položky identická, se může změnit. Čím hodnota připočtené hodnoty bude větší, tím větší váhu má tato vlastnost při doporučení.

Příklad: Máme uživatele Karla, který má rád hry od společnosti Ubisoft, akční, 3D, krabicové verze, v českém jazyce a hry, které se dají hrát online. Dále máme 4 položky a jejich vlastnosti.

Obrázek 16: Uživatel a položky

U všech vlastností zvolíme váhu 1. Poté každou položku porovnáme s Karlovým ideálem. Při shodě vlastnosti ideálu a vlastnosti položky přičteme k výslednému skóre hodnotu 1 (váhu). Hodnotu skóre položky po porovnání uložíme.

	Settlers 3	Far Cry 3	NBA 2013	Borderlands 2
Skóre:	4	6	2	3

Tabulka 3: Výsledek doporučení

Ze skóre vidíme, že dvou nejvyšších hodnot dosahují položky Far Cry 3 a Settlers 3. Tyto dvě položky se nejvíce blíží Karlovu ideálu, tak mu je doporučíme. Ale co když si provozovatel obchodu řekne, že nejdůležitější pro rozhodnutí, kterou hru si Karel koupí, je žánr hry. Upravíme tedy váhu u žánru na 3 a ostatní váhy vlastností necháme 1. Poté dostaneme následující výsledky:

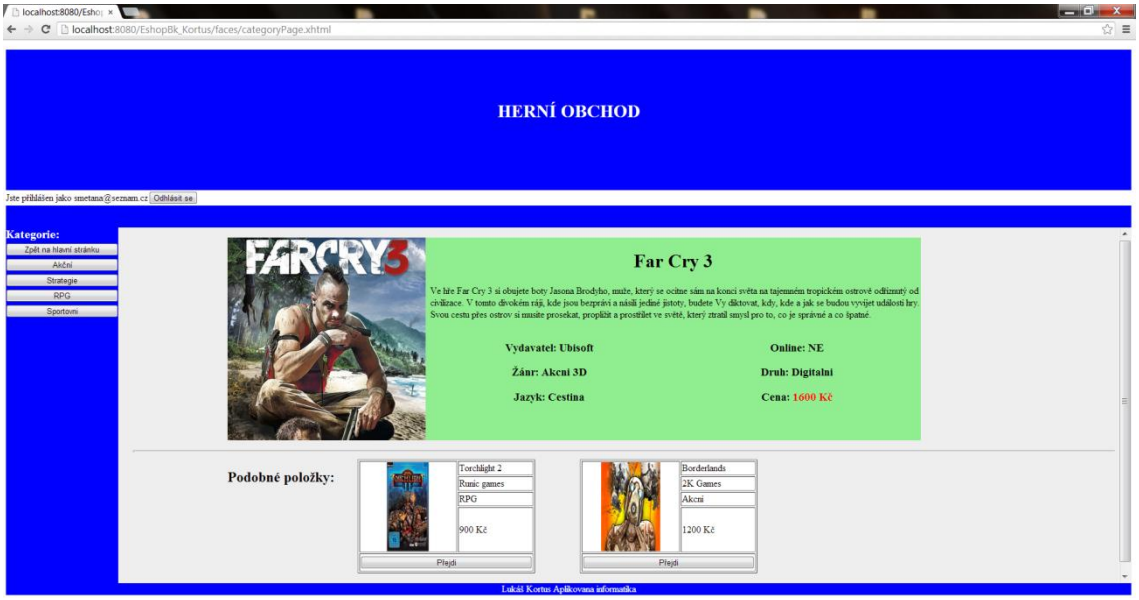
	Settlers 3	Far Cry 3	NBA 2013	Borderlands 2
Skóre:	4	8	2	6

Tabulka 4: Výsledek doporučení s váhou

Ze skóre nyní vidíme, že položka Borderlands 2 poskočila na druhé místo. V tomto případě doporučíme Karlovi položky Far Cry 3 a Borderlands 2.

6.3.4 Doporučení položek k aktuálně prohlížené položce

Pro toto doporučení není potřeba žádných informací o uživateli. Systém při tomto doporučení pracuje jen s položkami a s jejich atributy. Tím odpadá problém nového uživatele. V programu je toto doporučení řešeno podobně jako předchozí doporučení položek, které jsou pro zákazníka zajímavé, s tím rozdílem, že se nehledají dvě položky, které jsou nejvíce podobné ideálu zákazníka, ale hledají se ty dvě položky, které jsou nejvíce podobné aktuálně prohlížené položce zákazníkem.



Obrázek 17: Doporučení k prohlížené položce

V aplikaci to funguje tak, že po vybrání položky je ve třídě shopBean.java zavolána metoda třídy Core.java doItemRecommend(Item curItem), kde parametr curItem představuje zvolenou položku. Tato metoda nejprve zjistí, zda je načtený seznam všech položek obchodu, pokud ne tak zavolá metodu getItems() z třídy Db.java, která tento seznam načte. Dalším krokem je, že všechny vlastnosti aktuálně prohlížené položky jsou porovnány s atributy položek ze seznamu, kromě položky samé, která je zde také obsažena. Když je atribut položky shodný s atributem aktuální položky, k výslednému skóre je přičtena určitá hodnota. Když je skóre právě porovnané položky ze seznamu větší, než doposud maximální dosažené skoré jiné položky, tak je doposud maximální položka přesunuta na druhé místo a aktuální položka na první. Při situaci, kdy aktuální položka bude mít menší skóre než maximální, ale větší než má položka na druhém místě, je aktuální zapsána na druhé místo, stejně jako v předchozím případě.

```
public List doItemRecommend(Item curItem) {
        List<Item> itemRecommendList = new ArrayList<Item>();
        if (itemlist.isEmpty()) {
            try {
                itemlist = dao.getItems();
            } catch (Exception ex) {

Logger.getLogger(Core.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
```

```
        max = 0;
        max2 = 0;

        for (Item item : itemlist) {
            int count = 0;
            if (curItem.getId() != item.getId()) {
                if (curItem.getZanr().equals(item.getZanr())) {
                    count = count + 1;
                }

                if(curItem.getVydavatel().equals(item.getVydavatel()))
                {
                    count = count + 1;
                }

                if (curItem.getDruh().equals(item.getDruh())) {
                    count = count + 1;
                }

                if(curItem.getRozmerHry().equals(item.getRozmerHry()))
                {
                    count = count + 1;
                }

                if (curItem.isOnline() && item.isOnline()) {
                    count = count + 1;
                }

                if (curItem.getJazyk().equals(item.getJazyk())) {
                    count = count + 1;
                }

                if (count > max) {
                    max2 = max;
                    max = count;
                    itemRecommend2 = itemRecommend1;
                    itemRecommend1 = item;
                } else if (count > max2) {
                    max2 = count;
                    itemRecommend2 = item;
                }
            }
        }
        itemRecommendList.add(itemRecommend1);
        itemRecommendList.add(itemRecommend2);

        return itemRecommendList;
    }
```

Algoritmus 2: Doporučení k prohlížené položce

Metoda je tedy podobná předcházející metodě. Metoda vrací seznam, ve kterém jsou dvě položky, které jsou nejvíce podobné aktuálně prohlížené položce. Připočtená

hodnota ke skóre se může také změnit. Čím hodnota připočtené hodnoty bude větší, tím větší váhu má tato vlastnost při doporučení.

6.4 Problém chybějících údajů

Tento problém vznikne v případě, že při registraci uživatel nevyplní nebo odmítá vyplnit všechny jeho zájmy při registraci v oboru daného obchodu. Systém pak nemá podle jakých dat určit vhodnou podobnou položku jeho ideálu, která by pro něj mohla být zajímavá. Řešením je tyto údaje předpovědět na základě tužeb ostatních blízkých uživatelů.

6.4.1 Předpověď na základě věku uživatele

Tento druhý přístup je založen na myšlence, že uživatelé, kteří mají stejný nebo podobný věk, budou mít pravděpodobně podobné preference ke stejným položkám [26]. Například, že dvacetiletý uživatel bude mít rozdílné požadavky než padesátiletý, ale zároveň bude mít podobné jako dvaadvacetiletý uživatel. Algoritmus tohoto přístupu je následující: Pro každý chybějící údaj se vyhledají uživatelé, podobní věkem k aktuálnímu uživateli. Poté se vypočítá průměrná položka těchto uživatelů a tato položka se dosadí za chybějící hodnotu.

V aplikaci je algoritmus tohoto přístupu pozměněný. Při potvrzení registrace, ještě před zapsáním do databáze, je podle vyplněného věku zavoláním metody getSimilarFromUsers(int age) ze třídy Core.java vytvořen virtuální uživatel, jehož jednotlivé preference jsou nejvíce zastoupeny v rozsahu daného věku (± 5 let). Podle tohoto virtuálního uživatele jsou pak doplněny chybějící zájmy, které registrovaný uživatel nezadal.

```
public User getSimilarFromUsers(int age) throws Exception {
        List<User> userList = new ArrayList<User>();
        List<User> userRangeList = new ArrayList<User>();
        userList = dao.getUsers();
        User idealUser = new User();
```

```
        int idealRange = 5;
        int pomRange;
        for (User user : userList) {
             if (((user.getVek() - age) < idealRange) &&
            ((user.getVek() - age) > -idealRange)) {
                userRangeList.add(user);
                System.out.println(user.getJmeno());
            }
        }

        List<String> pomList = new ArrayList<String>();
        for (User userRange : userRangeList) {
            pomList.add(userRange.getFavZanr());

        }
        idealUser.setFavZanr(getIdeal(pomList));

         pomList = new ArrayList<String>();
        for (User userRange : userRangeList) {
            pomList.add(userRange.getFavVydavatel());

        }
        idealUser.setFavVydavatel(getIdeal(pomList));

        pomList = new ArrayList<String>();
        for (User userRange : userRangeList) {
            pomList.add(userRange.getFavRozmerHry());

        }
        idealUser.setFavRozmerHry(getIdeal(pomList));

        pomList = new ArrayList<String>();
        for (User userRange : userRangeList) {
            pomList.add(userRange.getFavJazyk());

        }
        idealUser.setFavJazyk(getIdeal(pomList));

        pomList = new ArrayList<String>();
        for (User userRange : userRangeList) {
            pomList.add(userRange.getFavDruh());

        }
        idealUser.setFavDruh(getIdeal(pomList));

        pomList = new ArrayList<String>();
        for (User userRange : userRangeList) {
            pomList.add(String.valueOf(userRange.isFavOnline()));

        }
        idealUser.setFavOnline(Boolean.valueOf(getIdeal(pomList)));

        return idealUser;
    }

    public String getIdeal(List<String> pomList)
    {
        String ideal = "";
```

```
        int count = 0;
        for (String s : pomList) {
            if (count < countNumber(pomList, s)) {
                ideal = s;
                count = countNumber(pomList, s);
            }
        }
         return ideal;
    }

    public int countNumber(List<String> list, String itemToCheck) {
        int count = 0;
        for (String s : list) {
            if (s.equals(itemToCheck)) {
                count++;
            }
        }
        return count;
    }
```

Algoritmus 3: Vytvoření virtuálního uživatele

6.5 Testování spolehlivosti navrženého řešení

Cílem testování bylo zjistit, jestli navržené řešení problému chybějících údajů je použitelné a spolehlivé i pro data reálných uživatelů.

Nejprve byl vytvořen dotazník, který měl simulovat otázky, které jsou uživateli položeny při registraci, respondent se tedy měl vžít do role budoucího uživatele elektronického obchodu. Tento dotazník byl šířen pomocí sociálních sítí a e-mailu. Vyplnilo jej 47 respondentů v rozmezí věku 13-27. Respondent, jehož věk byl 13, byl vyřazen z důvodu, že měl nejbližšího svému věku uživatele, kterému bylo 18 let. Testování tedy proběhlo na 46 respondentech v rozmezí věku 18-27. Tito respondenti poté byli rozděleni do dalších dvou skupin (18-22 let a 22-27let) a jejich atributy byly převedeny na číselné hodnoty.

Dále pak proběhlo první testování, kde hlavním úkolem bylo, zda způsob nejvíce zastoupené vlastnosti, který je použit pro vyhledání vlastností virtuálního uživatele, podle kterého se dosazují chybějící hodnoty, jenž uživatel nezadal, je efektivnější, než ten, který pro virtuálního uživatele vyhledává vlastnosti na základě průměru.

Pro každou skupinu byli vytvoření dva rituální uživatelé, kde virtuální uživatel 1 je složen z průměrných vlastností a virtuální uživatel 2 z nejvíce zastoupených vlastností.

Virtuální uživatelé pro všechna data					
	Žánr	Vydavatel	Druh	Online	Jazyk
V. uživatel 1	2,391304	2,95652174	1,652174	1,434783	1,52173
V. uživatel 2	2	3	2	1	2

Tabulka 5: Atributy virtuálních uživatelů pro všechna data

Virtuální uživatelé pro věk 18-22					
	Žánr	Vydavatel	Druh	Online	Jazyk
V. uživatel 1	2,578947	2,84210526	1,631579	1,315789	1,52631
V. uživatel 2	3	3	2	1	2

Tabulka 6: Atributy virtuálních uživatelů pro věk 18-22

Virtuální uživatelé pro věk 23-27					
	Žánr	Vydavatel	Druh	Online	Jazyk
V. uživatel 1	2,259259	3,03703704	1,666667	1,518519	1,51851
V. uživatel 2	2	3	2	2	2

Tabulka 7: Atributy virtuálních uživatelů pro věk 23-27

Z těchto uživatelů lze vyvodit závěr, že v případě těchto testovaných dat se v jednotlivých skupinách, po zaokrouhlení hodnot, virtuální uživatelé a jejich vlastnosti rovnají. U virtuálního uživatele se u některých vlastností hodnota pohybuje na hraně. Stačilo by jen malý počet respondentů a hodnota vlastnosti průměru by se změnila, hodnota vlastnosti nejvíce zastoupených nikoliv. Například, kdyby byl dotazník vyplněn dalšími dva respondenty, ve věku 18-22 let, kteří oba mají rádi akční hry, tak by se průměrná hodnota po zaokrouhlení změnila na hodnotu 2, ale nejvíce zastoupená vlastnost by stále byla 3. Tímto bylo dokázáno, že řešení, které bylo navrženo, je přesnější.

Druhé testování bylo zaměřeno na ověření toho, zda skupiny mají jiné zájmy a tím dokázat, že při určování předpovědi pro chybějící údaje záleží na věku. Tedy že uživatelé podobného věku budou mít podobné zájmy. Pro testování byly použity již z minulého příkladu vytvořené tabulky 6 a 7. Z těchto tabulek je možné vyčíst

rozdílnost virtuálních uživatelů pro jednotlivé skupiny. Zatímco pro skupinu 18-22 let by virtuální uživatel vypadal podle tabulky 8, pro skupinu uživatelů 23-27 let by se virtuální uživatel, který je zobrazený také tabulce 8, lišil ve vlastnostech žánru a hraní online. Bylo tedy dokázáno, že tyto dvě skupiny mají jiné zájmy a tedy byla potvrzena myšlenka předpovědi na základě věku uživatele.

Název vlastnosti	Žánr	Vydavatel	Druh	Online	Jazyk
V.uživatel 18-22	RPG	Ubisoft	Digitální	Ano	Angličtina
V.uživatel 23-27	Strategie	Ubisoft	Digitální	Ne	Angličtina

Tabulka 8: Porovnání virtuálních uživatelů skupin

Závěr

Cílem této práce bylo zmapovat typy doporučovacích systémů a jejich funkce, konkrétně uvést příklady použití doporučovacích systémů u konkrétních příkladů a popsat mechanismy, které používají doporučovací systémy. Dále bylo cílem navrhnout modul pro systém e-commerce, který bude zpracovávat informace, které zákazník poskytl a zároveň mu bude, podle těchto informací, doporučovat a poskytovat informace o položkách v tomto daném systému e-commerce a dále pak provést analýzu a popsat ji pomocí UML diagramu, navrhnout způsob řešení případu, kdy uživatel zadá neúplná data a ověřit tento navržený způsob řešení problému neúplných dat.

V první části byla přiblížena problematika e-commerce a byly uvedeny výhody oproti kamenným obchodům z pohledu zákazníka i z pohledu majitele obchodu. V následující části byly vytyčeny cíle této práce.

Třetí část se zabývala uživatelskými preferencemi, co ovlivňuje uživatele a z čeho je možné činit závěry či předpoklady o uživatelově preferenci. Dále se tato popisovala dva druhy preferencí a to dlouhodobou a krátkodobou. Nakonec zde byl vysvětlen předmět preference.

V další části byla práce zaměřena na doporučovací systémy. Nejprve byly vybrány čtyři doporučovací systémy z reálného světa, které byly následně popsány. Dále byly zmapovány typy doporučovacích systému a uvedeno na jakém principu fungují a jaké mají výhody a nevýhody, případně problémy.

V páté části byly popsány algoritmy využívané při procesu doporučení pro výpočet podobnosti.

Šestá část byla zaměřena převážně prakticky. Byl navrhnut modul pro e-commerce, ve kterém byl kladen důraz na problematiku studeného startu a popsány použité technologie při vývoji. Modul doporučuje položky několika způsoby. Prvním je zobrazení deseti nejprodávanějších výrobků, dále pak zobrazení deseti nejnavštěvovanějších výrobků. Mezi hlavní doporučení ale patří doporučení k aktuální prohlížené položce, kde se doporučí dva výrobky, které se jí nejvíce podobají vlastnostmi. Poslední a nejdůležitější doporučení je založeno na vyhledání dvou položek, které se nejvíce podobají ideálu uživatele. Tento ideál se vytvoří při registraci zodpovězením několika otázek. Dalším krokem byla analýza tohoto modulu a navrhnutí řešení případu, kdy uživatel zadá neúplná data a ověřit tento navržený způsob řešení problému neúplných dat.

Výsledkem této práce je tedy poskytnutí informací o doporučovacích systémech pro e-commerce a přehled jejich mechanismů, dále naprogramování a implementace modulu do webové aplikace simulující e-shop. Modul zpracovává informace, které zákazník poskytl a zároveň mu podle těchto informací doporučuje položky v tomto systému e-commerce, a který řeší problém studeného startu a problém, kdy uživatel zadá neúplná data.

Seznam obrázků

Seznam tabulek

Seznam algoritmů

Literatura

[1] PEŠKA, Ladislav. *Uživatelské preference v prostředí prodejních webů* [online]. Praha, 2010 [cit. 2013-03-18]. Dostupné z: <up-comp.googlecode.com/files/diplomka-text_final.pdf>. Diplomová práce. Univerzita Karlova v Praze, Matematicko-fyzikální fakulta, Katedra softwarového inženýrství. Vedoucí práce prof. RNDr. Peter Vojtáš, DrSC.

[2] VOJTÁŠ, Peter. *Modely uživatelských preferencí* [online]. [cit. 2013-03-18]. Dostupné z: <http://www.ksi.mff.cuni.cz/~vojtas/vyuka/NDBI021PrincipyUzivatelskychPreferenci/1112_NSWI021_DotazovaniSPreferencemi/DBI021modelyUzivatele.ppt>

[3] MELVILLE P., SINDHWANI, V., Recommender Systems [online]. In *Encyclopedia of Machine Learning*, 2010. [cit. 2013-03-18]. Dostupné z WWW: <http://www.prem-melville.com/publications/recommender-systems-eml2010.pdf>

[4] VALA, Martin. *E-learning – doporučovací systémy* [online]. Brno, 2012 [cit. 2013-03-18]. Dostupné z: <http://is.muni.cz/th/359917/fi_b/bp_final_vala.pdf>. Bakalářská práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Mgr. Jan Géryk.

[5] CAPITAL PARTNERS, a.s. *SHRNUTÍ INVESTIČNÍHO DOPORUČENÍ Amazon.com, Inc.* [online]. 2009 [cit. 2013-03-18]. Dostupné z: <http://www.globalmarkets.bg/images/other/39_AMZN_.pdf>

[6] STRUŽSKÝ, Martin. *Kolaborativní filtrování pro adaptivní web* [online]. Praha, 2009 [cit. 2013-03-18]. Dostupné z: <https://dip.felk.cvut.cz/browse/pdfcache/struzm1_2009bach.pdf>. Bakalářská práce. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra počítačů. Vedoucí práce Ing. Martin Balík.

[7] Historie YouTube. In: *Portál věnovaný Youtube* [online]. [cit. 2013-03-18]. Dostupné z: <http://youtube.vseznamu.cz/historie-youtube>

[8] Doporučená videa. *Podpora Google* [online]. [cit. 2012-12-26]. Dostupné z: <http://support.google.com/youtube/bin/answer.py?hl=cs&answer=143421>

[9] Facebook historie. *Facebook přihlášení* [online]. [cit. 2013-03-18]. Dostupné z: <http://www.facebook-prihlaseni.cz/com-navody/historie.php>

[10] Historie a současnost. In: *Alza.cz* [online]. [cit. 2013-03-18]. Dostupné z: <http://www.alza.cz/article/141.htm>

[11] JANNACH, D., ZANKER, M., FELFERNIG, A., FRIEDRICH, G., *Recommender Systems: An Introduc-tion.* Cambridge University Press, 2010. ISBN: 978-0521493369.

[12]CVENGROŠ, Petr. *Universal Recommender System* [online]. Prague, 2011 [cit. 2013-03-18]. Dostupné z:< https://unresyst.googlecode.com/files/dp_final.pdf>. Master thesis. Charles University in Prague, Faculty of Mathematics and Physics, Department of Software Engineering. Vedoucí práce prof. RNDr. Peter Vojtáš, DrSc.

[13] SARWAR, Badrul, George KARYPIS, Joseph KONSTAN a John RIEDL. *Item-Based Collaborative Filtering Recommendation Algorithms* [online]. New York, 2001 [cit. 2013-03-18]. Dostupné z: <http://www.ra.ethz.ch/cdstore/www10/papers/pdf/p519.pdf>. Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN.

[14] ALMAZRO, Dhoha, Ghadeer SHAHATAH, Lamia ALBDULKARIM, Mona KHEREES, Romy MARTINEZ a William NZOUKOU. *A Survey Paper on Recommender Systems* [online]. 2010 [cit. 2013-03-18]. Dostupné z: <http://arxiv.org/pdf/1006.5278v4.pdf>

[15] SARWAR, Badrul. *Model-based Collaborative Filtering Algorithms*. 2001 [cit. 2013-03-18]. Dostupné z: <http://www10.org/cdrom/papers/519/node8.html>

[16] ADOMAVICIUS, Gediminas a Alexander TUZHILIN. *Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions*. 2005[cit. 2013-03-18]. Dostupné z: <http://citeseer.ist.psu.edu/viewdoc/download?doi=10.1.1.107.2790&rep=rep1&type=pdf>

[17] SU, Xiaoyuan a Taghi KHOSHGOFTAAR. *A Survey of Collaborative Filtering Techniques* [online]. 2009 [cit. 2013-03-18]. Dostupné z: <http://www.hindawi.com/journals/aai/2009/421425/>

[18] GHAZANFAR a PRUGEL-BENNETT. *Fulfilling the Needs of Gray-Sheep Users in Recommender Systems, A Clustering Solution* [online]. Southampton

[cit. 2013-03-18]. Dostupné z: <http://eprints.soton.ac.uk/271770/1/PaperID_201.pdf>. School of Electronics and Computer Science University of Southampton.

[19] BURKE, R., *Hybrid Recommender Systems: Survey and Experiments* [online]. In User Modeling and User-Adapted Interaction Volume 12 Issue 4, 2002. s. 331–370. [cit. 2013-03-18]. Dostupné z WWW: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.8200&rep=rep1&type=pdf >.

[20] HOUDEK, Michal, Tomáš SVOBODA a Tomáš PROCHÁZKA. *Klasifikace podle nejbližších sousedů Nearest Neighbour Classification [k-NN]* [online]. 2001 [cit. 2013-03-18]. Dostupné z: <http://cmp.felk.cvut.cz/cmp/courses/recognition/zapis_prednasky/zapis_01/4/rpz4.pdf >

[21] *Pearson product-moment correlation coefficient* [online]. [cit. 2013-03-18]. Dostupné z: <http://www.answers.com/topic/pearson-s-correlation>

[22]*Nelineární korelační závislost* [online]. [cit. 2013-03-18]. Dostupné z: <http://cit.vfu.cz/statpotr/POTR/Teorie/Predn5/nelinear.htm>

[23] KOTLÁROVÁ, Jaroslava. *Doporučovací systém pro internetové obchody* [online]. Brno, 2010 [cit. 2013-03-18]. Dostupné z: <http://is.muni.cz/th/151384/fi_m/diplomovaprace.pdf>. Diplomová práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce RNDr. Radek Ošlejšek, Ph.D.

[24] *Vítejte u NetBeans a na stránkách www.netbeans.org* [online]. 2013 [cit. 2013-03-18]. Dostupné z: <http://netbeans.org/index_cs.html>

[25] KAPITÁN, Lukáš. *Technologie SQL a PHP ve výuce* [online]. Praha, 2010 [cit. 2013-03-18]. Dostupné z: http://info.sks.cz/www/zavprace/soubory/71930.pdf. Bakalářská práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky.

[26] XIA, Weiwei, Liang HE, GU a Keqin HE. *Effective Collaborative Filtering Approaches Based on Missing Data Imputation* [online]. 2009 [cit. 2013-03-18]. Dostupné z:

<http://www.ica.stc.sh.cn/picture/article/176/a9/c8/c0b56fa54cc9a334ea065e3375ff/15e3790d-d108-4131-9a75-04724d789103.pdf>

[27] VÁCLAVÍK, Lukáš. Nový vzhled YouTube se dále rýsuje. [online]. 2011 [cit. 2013-03-26].
<Dostupné z: http://www.cnews.cz/novy-vzhled-youtube-se-dale-rysuje>

Přílohy

Příloha č. 1

Obsah přiloženého CD

- text práce ve formátu PDF,
- exportovaná databáze ve formátu sql,
- zdrojový kód webové aplikace v projektu netbeans.

Příloha č. 2

Dotazník k bakalářské práci

Sběr dat pro doporučovací systém

Dobry den. Tento dotazník slouží pro sběr reálných dat reálných uživatelů pro bakalářskou práci pro doporučovací systém e-shop, který je zaměřen na prodej her. E-shop nabízí jen určitý sortiment zboží a otázky budou směřovány na něj. Data jsou sbírána proto, aby se ověřila funkčnost postupů, které jsou v aplikaci použity. Nyní se vžíte do kůže zákazníka a vyplňte prosím následující otázky. Dotazník je anonymní.

1) **Jsem?**
   - Muž
   - Žena

2) **Můj nejoblíbenější žánr hry?**
   - Akční
   - Strategie
   - RPG
   - Sportovní

3) **Můj nejoblíbenější herní vydavatel z nabízených možností?**
   - Runic Games
   - 2K Games
   - Ubisoft
   - Codemasters

4) **Preferuji zakoupení v podobě:**
   - Krabicové
   - Digitální

5) **Preferuji online hry?**
   - Ano
   - Ne

6) **Jako jazyk hry preferuji:**
   - Angličtina
   - Čeština

7) **A nakonec prosím o vyplnění Vašeho věku.**

Děkuji za pomoc a Váš čas při vyplnění dotazníku.